SparkseePython3

6.1.0

Generated by Doxygen 1.16.1

1 Namespace Index 1

1.1 Package List . . . 1

2 Hierarchical Index 1

2.1 Class Hierarchy . . . 1

3 Class Index 4

3.1 Class List . . . 4

4 Namespace Documentation 9

4.1 sparksee Namespace Reference . . . 9

4.1.1 Detailed Description . . . 13

4.1.2 Function Documentation . . . 13

5 Class Documentation 15

5.1 sparksee.Attribute Class Reference . . . 15

5.1.1 Detailed Description . . . 16

5.1.2 Member Function Documentation . . . 16

5.2 sparksee.AttributeKind Class Reference . . . 18

5.2.1 Detailed Description . . . 19

5.2.2 Member Data Documentation . . . 19

5.3 sparksee.AttributeList Class Reference . . . 19

5.3.1 Detailed Description . . . 20

5.3.2 Constructor & Destructor Documentation . . . 20

5.3.3 Member Function Documentation . . . 20

5.4 sparksee.AttributeListIterator Class Reference . . . 21

5.4.1 Detailed Description . . . 22

5.4.2 Member Function Documentation . . . 22

5.5 sparksee.AttributeStatistics Class Reference . . . 23

5.5.1 Detailed Description . . . 24

5.5.2 Member Function Documentation . . . 24

5.6 sparksee.BooleanList Class Reference . . . 27

5.6.1 Detailed Description . . . 28

5.6.2 Constructor & Destructor Documentation . . . 28

5.6.3 Member Function Documentation . . . 28

5.7 sparksee.BooleanListIterator Class Reference . . . 29

5.7.1 Detailed Description . . . 29

5.7.2 Member Function Documentation . . . 30

5.8 sparksee.CommunitiesSCD Class Reference . . . 31

5.8.1 Detailed Description . . . 32

5.8.2 Constructor & Destructor Documentation . . . 33

5.8.3 Member Function Documentation . . . 33

5.9 sparksee.CommunityDetection Class Reference . . . 37

5.9.1 Detailed Description . . . 38

5.9.2 Member Function Documentation . . . 38
5.10 sparksee.Condition Class Reference . . . 41
5.10.1 Detailed Description . . . 41
5.10.2 Member Data Documentation . . . 42
5.11 sparksee.ConnectedComponents Class Reference . . . 44
5.11.1 Detailed Description . . . 44
5.11.2 Constructor & Destructor Documentation . . . 45
5.11.3 Member Function Documentation . . . 45
5.12 sparksee.Connectivity Class Reference . . . 47
5.12.1 Detailed Description . . . 48
5.12.2 Member Function Documentation . . . 48
5.13 sparksee.Context Class Reference . . . 51
5.13.1 Detailed Description . . . 51
5.13.2 Constructor & Destructor Documentation . . . 52
5.13.3 Member Function Documentation . . . 52
5.14 sparksee.CSVReader Class Reference . . . 55
5.14.1 Detailed Description . . . 57
5.14.2 Member Function Documentation . . . 57
5.15 sparksee.CSVWriter Class Reference . . . 61
5.15.1 Detailed Description . . . 62
5.15.2 Member Function Documentation . . . 63
5.16 sparksee.Database Class Reference . . . 65
5.16.1 Detailed Description . . . 66
5.16.2 Member Function Documentation . . . 67
5.17 sparksee.DatabaseStatistics Class Reference . . . 69
5.17.1 Detailed Description . . . 70
5.17.2 Member Function Documentation . . . 70
5.18 sparksee.DataType Class Reference . . . 71
5.18.1 Detailed Description . . . 72
5.18.2 Member Data Documentation . . . 72
5.19 sparksee.DefaultExport Class Reference . . . 73
5.19.1 Detailed Description . . . 74
5.19.2 Member Function Documentation . . . 75
5.20 sparksee.DisjointCommunities Class Reference . . . 82
5.20.1 Detailed Description . . . 83
5.20.2 Constructor & Destructor Documentation . . . 83
5.20.3 Member Function Documentation . . . 84
5.21 sparksee.DisjointCommunityDetection Class Reference . . . 86
5.21.1 Detailed Description . . . 87
5.21.2 Member Function Documentation . . . 87
5.22 sparksee.EdgeData Class Reference . . . 91
5.22.1 Member Function Documentation . . . 92

5.23 sparksee.EdgeExport Class Reference . . . 92
5.23.1 Detailed Description . . . 93
5.23.2 Member Function Documentation . . . 94
5.24 sparksee.EdgesDirection Class Reference . . . 97
5.24.1 Detailed Description . . . 98
5.25 sparksee.EdgeTypeExporter Class Reference . . . 98
5.25.1 Detailed Description . . . 99
5.25.2 Constructor & Destructor Documentation . . . 99
5.25.3 Member Function Documentation . . . 100
5.26 sparksee.EdgeTypeLoader Class Reference . . . 104
5.26.1 Detailed Description . . . 105
5.26.2 Constructor & Destructor Documentation . . . 106
5.26.3 Member Function Documentation . . . 106
5.27 sparksee.ExportManager Class Reference . . . 113
5.27.1 Detailed Description . . . 114
5.27.2 Member Function Documentation . . . 114
5.28 sparksee.ExportType Class Reference . . . 118
5.28.1 Detailed Description . . . 118
5.28.2 Member Data Documentation . . . 119
5.29 sparksee.Graph Class Reference . . . 119
5.29.1 Detailed Description . . . 123
5.29.2 Member Function Documentation . . . 123
5.30 sparksee.GraphExport Class Reference . . . 155
5.30.1 Detailed Description . . . 156
5.30.2 Member Function Documentation . . . 156
5.31 sparksee.Int32List Class Reference . . . 157
5.31.1 Detailed Description . . . 157
5.31.2 Constructor & Destructor Documentation . . . 158
5.31.3 Member Function Documentation . . . 158
5.32 sparksee.Int32ListIterator Class Reference . . . 159
5.32.1 Detailed Description . . . 159
5.32.2 Member Function Documentation . . . 159
5.33 sparksee.KeyValue Class Reference . . . 160
5.33.1 Member Function Documentation . . . 160
5.34 sparksee.KeyValues Class Reference . . . 161
5.34.1 Detailed Description . . . 161
5.34.2 Member Function Documentation . . . 162
5.35 sparksee.KOpt Class Reference . . . 163
5.35.1 Detailed Description . . . 163
5.36 sparksee.LogLevel Class Reference . . . 163
5.36.1 Detailed Description . . . 164
5.36.2 Member Data Documentation . . . 164

5.37 sparksee.MissingEndpoint Class Reference . . . 165
5.37.1 Detailed Description . . . 166
5.38 sparksee.NodeExport Class Reference . . . 166
5.38.1 Detailed Description . . . 167
5.38.2 Member Function Documentation . . . 168
5.39 sparksee.NodeShape Class Reference . . . 173
5.39.1 Detailed Description . . . 174
5.40 sparksee.NodeTypeExporter Class Reference . . . 174
5.40.1 Detailed Description . . . 175
5.40.2 Constructor & Destructor Documentation . . . 175
5.40.3 Member Function Documentation . . . 176
5.41 sparksee.NodeTypeLoader Class Reference . . . 178
5.41.1 Detailed Description . . . 180
5.41.2 Constructor & Destructor Documentation . . . 180
5.41.3 Member Function Documentation . . . 180
5.42 sparksee.Objects Class Reference . . . 185
5.42.1 Detailed Description . . . 186
5.42.2 Member Function Documentation . . . 186
5.43 sparksee.ObjectsIterator Class Reference . . . 194
5.43.1 Detailed Description . . . 194
5.43.2 Member Function Documentation . . . 195
5.44 sparksee.ObjectType Class Reference . . . 196
5.44.1 Detailed Description . . . 196
5.45 sparksee.OIDList Class Reference . . . 196
5.45.1 Detailed Description . . . 197
5.45.2 Constructor & Destructor Documentation . . . 197
5.45.3 Member Function Documentation . . . 197
5.46 sparksee.OIDListIterator Class Reference . . . 199
5.46.1 Detailed Description . . . 199
5.46.2 Member Function Documentation . . . 199
5.47 sparksee.Order Class Reference . . . 200
5.47.1 Detailed Description . . . 200
5.48 sparksee.PageRank Class Reference . . . 200
5.48.1 Detailed Description . . . 201
5.48.2 Constructor & Destructor Documentation . . . 201
5.48.3 Member Function Documentation . . . 202
5.49 sparksee.Platform Class Reference . . . 206
5.49.1 Detailed Description . . . 206
5.49.2 Member Function Documentation . . . 206
5.50 sparksee.PlatformStatistics Class Reference . . . 206
5.50.1 Detailed Description . . . 207
5.50.2 Member Function Documentation . . . 207

5.51 sparksee.Query Class Reference . . . 208
5.51.1 Detailed Description . . . 209
5.51.2 Member Function Documentation . . . 209
5.52 sparksee.QueryContext Class Reference . . . 211
5.52.1 Detailed Description . . . 211
5.53 sparksee.QueryLanguage Class Reference . . . 211
5.53.1 Detailed Description . . . 211
5.54 sparksee.QueryStream Class Reference . . . 211
5.54.1 Detailed Description . . . 212
5.54.2 Member Function Documentation . . . 212
5.55 sparksee.QueryStreamManager Class Reference . . . 213
5.55.1 Detailed Description . . . 214
5.55.2 Member Function Documentation . . . 214
5.56 sparksee.RandomWalk Class Reference . . . 215
5.56.1 Detailed Description . . . 216
5.56.2 Constructor & Destructor Documentation . . . 217
5.56.3 Member Function Documentation . . . 217
5.57 sparksee.ResultSet Class Reference . . . 223
5.57.1 Detailed Description . . . 223
5.57.2 Member Function Documentation . . . 224
5.58 sparksee.ResultSetList Class Reference . . . 226
5.58.1 Detailed Description . . . 227
5.58.2 Constructor & Destructor Documentation . . . 227
5.58.3 Member Function Documentation . . . 227
5.59 sparksee.ResultSetListIterator Class Reference . . . 228
5.59.1 Detailed Description . . . 229
5.59.2 Member Function Documentation . . . 229
5.60 sparksee.RowReader Class Reference . . . 230
5.60.1 Detailed Description . . . 230
5.60.2 Member Function Documentation . . . 231
5.61 sparksee.RowWriter Class Reference . . . 232
5.61.1 Detailed Description . . . 233
5.61.2 Member Function Documentation . . . 233
5.62 sparksee.ScriptParser Class Reference . . . 234
5.62.1 Detailed Description . . . 234
5.62.2 Member Function Documentation . . . 235
5.63 sparksee.Session Class Reference . . . 236
5.63.1 Detailed Description . . . 237
5.63.2 Member Function Documentation . . . 237
5.64 sparksee.ShortestPath Class Reference . . . 239
5.64.1 Detailed Description . . . 240
5.64.2 Member Function Documentation . . . 240

5.65 sparksee.SinglePairShortestPath Class Reference . . . 243
5.65.1 Detailed Description . . . 245
5.65.2 Member Function Documentation . . . 245
5.66 sparksee.SinglePairShortestPathBFS Class Reference . . . 249
5.66.1 Detailed Description . . . 251
5.66.2 Constructor & Destructor Documentation . . . 251
5.66.3 Member Function Documentation . . . 251
5.67 sparksee.SinglePairShortestPathDijkstra Class Reference . . . 255
5.67.1 Detailed Description . . . 257
5.67.2 Constructor & Destructor Documentation . . . 257
5.67.3 Member Function Documentation . . . 258
5.68 sparksee.SinglePairShortestPathDijkstraDynamicCost Class Reference . . . 262
5.68.1 Detailed Description . . . 262
5.68.2 Member Function Documentation . . . 263
5.69 sparksee.Sparksee Class Reference . . . 263
5.69.1 Detailed Description . . . 264
5.69.2 Constructor & Destructor Documentation . . . 265
5.69.3 Member Function Documentation . . . 265
5.70 sparksee.SparkseeConfig Class Reference . . . 274
5.70.1 Detailed Description . . . 278
5.70.2 Constructor & Destructor Documentation . . . 279
5.70.3 Member Function Documentation . . . 280
5.71 sparksee.SparkseeProperties Class Reference . . . 301
5.71.1 Detailed Description . . . 302
5.71.2 Member Function Documentation . . . 302
5.72 sparksee.StringList Class Reference . . . 304
5.72.1 Detailed Description . . . 305
5.72.2 Constructor & Destructor Documentation . . . 305
5.72.3 Member Function Documentation . . . 305
5.73 sparksee.StringListIterator Class Reference . . . 306
5.73.1 Detailed Description . . . 307
5.73.2 Member Function Documentation . . . 307
5.74 sparksee.StrongConnectivity Class Reference . . . 308
5.74.1 Detailed Description . . . 309
5.74.2 Member Function Documentation . . . 309
5.75 sparksee.StrongConnectivityGabow Class Reference . . . 313
5.75.1 Detailed Description . . . 314
5.75.2 Constructor & Destructor Documentation . . . 315
5.75.3 Member Function Documentation . . . 315
5.76 sparksee.TextStream Class Reference . . . 318
5.76.1 Detailed Description . . . 319
5.76.2 Constructor & Destructor Documentation . . . 319

5.76.3 Member Function Documentation . . . 319
5.77 sparksee.Traversal Class Reference . . . 321
5.77.1 Detailed Description . . . 322
5.77.2 Member Function Documentation . . . 322
5.78 sparksee.TraversalBFS Class Reference . . . 326
5.78.1 Detailed Description . . . 327
5.78.2 Constructor & Destructor Documentation . . . 328
5.78.3 Member Function Documentation . . . 328
5.79 sparksee.TraversalDFS Class Reference . . . 332
5.79.1 Detailed Description . . . 333
5.79.2 Constructor & Destructor Documentation . . . 334
5.79.3 Member Function Documentation . . . 334
5.80 sparksee.Type Class Reference . . . 338
5.80.1 Detailed Description . . . 339
5.80.2 Member Function Documentation . . . 339
5.81 sparksee.TypeExporter Class Reference . . . 341
5.81.1 Detailed Description . . . 342
5.81.2 Member Function Documentation . . . 342
5.82 sparksee.TypeExporterEvent Class Reference . . . 345
5.82.1 Detailed Description . . . 345
5.82.2 Member Function Documentation . . . 345
5.83 sparksee.TypeExporterListener Class Reference . . . 346
5.83.1 Detailed Description . . . 346
5.83.2 Member Function Documentation . . . 347
5.84 sparksee.TypeList Class Reference . . . 347
5.84.1 Detailed Description . . . 347
5.84.2 Constructor & Destructor Documentation . . . 348
5.84.3 Member Function Documentation . . . 348
5.85 sparksee.TypeListIterator Class Reference . . . 349
5.85.1 Detailed Description . . . 349
5.85.2 Member Function Documentation . . . 349
5.86 sparksee.TypeLoader Class Reference . . . 350
5.86.1 Detailed Description . . . 351
5.86.2 Member Function Documentation . . . 352
5.87 sparksee.TypeLoaderEvent Class Reference . . . 356
5.87.1 Detailed Description . . . 357
5.88 sparksee.TypeLoaderListener Class Reference . . . 357
5.88.1 Member Function Documentation . . . 357
5.89 sparksee.UnrecoverableError Class Reference . . . 357
5.89.1 Detailed Description . . . 358
5.90 sparksee.Value Class Reference . . . 358
5.90.1 Detailed Description . . . 359

5.90.2 Constructor & Destructor Documentation . . . 360
5.90.3 Member Function Documentation . . . 360
5.91 sparksee.ValueArray Class Reference . . . 370
5.91.1 Detailed Description . . . 371
5.91.2 Member Function Documentation . . . 372
5.92 sparksee.ValueList Class Reference . . . 383
5.92.1 Detailed Description . . . 383
5.92.2 Constructor & Destructor Documentation . . . 383
5.92.3 Member Function Documentation . . . 384
5.93 sparksee.ValueListIterator Class Reference . . . 385
5.93.1 Member Function Documentation . . . 385
5.94 sparksee.Values Class Reference . . . 386
5.94.1 Detailed Description . . . 386
5.94.2 Member Function Documentation . . . 386
5.95 sparksee.ValuesIterator Class Reference . . . 388
5.95.1 Detailed Description . . . 388
5.95.2 Member Function Documentation . . . 388
5.96 sparksee.WeakConnectivity Class Reference . . . 390
5.96.1 Detailed Description . . . 391
5.96.2 Member Function Documentation . . . 391
5.97 sparksee.WeakConnectivityDFS Class Reference . . . 395
5.97.1 Detailed Description . . . 396
5.97.2 Constructor & Destructor Documentation . . . 397
5.97.3 Member Function Documentation . . . 397

1 Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

sparksee
Embedded Graph Database **9**

2 Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

sparksee.Attribute **15**

sparksee.AttributeKind **18**

sparksee.AttributeList 19

sparksee.AttributeListIterator 21

sparksee.AttributeStatistics 23

sparksee.BooleanList 27

sparksee.BooleanListIterator 29

sparksee.CommunityDetection 37

sparksee.DisjointCommunityDetection 86

sparksee.CommunitiesSCD 31

sparksee.Condition 41

sparksee.ConnectedComponents 44

sparksee.Connectivity 47

sparksee.StrongConnectivity 308

sparksee.StrongConnectivityGabow 313

sparksee.WeakConnectivity 390

sparksee.WeakConnectivityDFS 395

sparksee.Context 51

sparksee.Database 65

sparksee.DatabaseStatistics 69

sparksee.DataType 71

sparksee.DisjointCommunities 82

sparksee.EdgeData 91

sparksee.EdgeExport 92

sparksee.EdgesDirection 97

sparksee.ExportManager 113

sparksee.DefaultExport 73

sparksee.ExportType 118

sparksee.Graph 119

sparksee.GraphExport 155

sparksee.Int32List 157

sparksee.Int32ListIterator 159

sparksee.KeyValue 160

sparksee.KeyValues 161

sparksee.KOpt 163

sparksee.LogLevel 163

sparksee.MissingEndpoint 165

sparksee.NodeExport 166

sparksee.NodeShape 173

sparksee.Objects 185

sparksee.ObjectsIterator 194

sparksee.ObjectType 196

sparksee.OIDList 196

sparksee.OIDListIterator 199

sparksee.Order 200

sparksee.PageRank 200

sparksee.Platform 206

sparksee.PlatformStatistics 206

sparksee.Query 208

sparksee.QueryContext 211

sparksee.QueryLanguage 211

sparksee.QueryStream 211

sparksee.QueryStreamManager 213

sparksee.ResultSet 223

sparksee.ResultSetList 226

sparksee.ResultSetListIterator 228

sparksee.RowReader 230

- sparksee.CSVReader 55

sparksee.RowWriter 232

- sparksee.CSVWriter 61

sparksee.ScriptParser 234

sparksee.Session 236

sparksee.ShortestPath 239

- sparksee.SinglePairShortestPath 243
  - sparksee.SinglePairShortestPathBFS 249
  - sparksee.SinglePairShortestPathDijkstra 255

sparksee.SinglePairShortestPathDijkstraDynamicCost 262
sparksee.Sparksee 263
sparksee.SparkseeConfig 274
sparksee.SparkseeProperties 301
sparksee.StringList 304
sparksee.StringListIterator 306
sparksee.TextStream 318
sparksee.Traversal 321
sparksee.RandomWalk 215
sparksee.TraversalBFS 326
sparksee.TraversalDFS 332
sparksee.Type 338
sparksee.TypeExporter 341
sparksee.EdgeTypeExporter 98
sparksee.NodeTypeExporter 174
sparksee.TypeExporterEvent 345
sparksee.TypeExporterListener 346
sparksee.TypeList 347
sparksee.TypeListIterator 349
sparksee.TypeLoader 350
sparksee.EdgeTypeLoader 104
sparksee.NodeTypeLoader 178
sparksee.TypeLoaderEvent 356
sparksee.TypeLoaderListener 357
sparksee.UnrecoverableError 357
sparksee.Value 358
sparksee.ValueArray 370
sparksee.ValueList 383
sparksee.ValueListIterator 385
sparksee.Values 386
sparksee.ValuesIterator 388

3 Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

sparksee.Attribute
Attribute data class 15

sparksee.AttributeKind
Attribute kind enumeration 18

sparksee.AttributeList
Sparksee attribute identifier list 19

sparksee.AttributeListIterator
AttributeList iterator class 21

sparksee.AttributeStatistics
Attribute statistics class 23

sparksee.BooleanList
Boolean list 27

sparksee.BooleanListIterator
BooleanList iterator class 29

sparksee.CommunitiesSCD
CommunitiesSCD class 31

sparksee.CommunityDetection
CommunityDetection class 37

sparksee.Condition
Condition operators enumeration 41

sparksee.ConnectedComponents
ConnectedComponents class 44

sparksee.Connectivity
Connectivity class 47

sparksee.Context
Context class 51

sparksee.CSVReader
CSVReader interface 55

sparksee.CSVWriter
CSVWriter interface 61

sparksee.Database
Database class 65

sparksee.DatabaseStatistics
Database statistics 69

sparksee.DataType
Data type (domain) enumeration 71

sparksee.DefaultExport
Default implementation for ExportManager class 73

sparksee.DisjointCommunities
DisjointCommunities class 82

sparksee.DisjointCommunityDetection
DisjointCommunityDetection class 86

sparksee.EdgeData 91

sparksee.EdgeExport
Stores edge exporting values 92

sparksee.EdgesDirection
Edges direction enumeration 97

sparksee.EdgeTypeExporter
EdgeTypeExporter class 98

sparksee.EdgeTypeLoader
EdgeTypeLoader class 104

sparksee.ExportManager
Defines how to export a graph to an external format 113

sparksee.ExportType
Export type 118

sparksee.Graph
Graph class 119

sparksee.GraphExport
Stores the graph exporting values 155

sparksee.Int32List
Sparksee 32-bit signed integer list 157

sparksee.Int32ListIterator
Int32List iterator class 159

sparksee.KeyValue 160

sparksee.KeyValues
Value set class 161

sparksee.KOpt
KOpt class 163

sparksee.LogLevel
Log level enumeration 163

sparksee.MissingEndpoint
The policy to follow whenever and edge endpoint is missing during a loading 165

sparksee.NodeExport
Stores the node exporting values 166

sparksee.NodeShape
Node shape 173

sparksee.NodeTypeExporter
NodeTypeExporter class 174

sparksee.NodeTypeLoader
NodeTypeLoader class 178

sparksee.Objects
Object identifier set class 185

sparksee.ObjectsIterator
ObjectsIterator class 194

sparksee.ObjectType
Object type enumeration 196

sparksee.OIDList
Sparksee object identifier list 196

sparksee.OIDListIterator
OIDList iterator class 199

sparksee.Order
Order enumeration 200

sparksee.PageRank
PageRank class 200

sparksee.Platform
Platform class 206

sparksee.PlatformStatistics
Platform data and statistics 206

sparksee.Query
Query class 208

sparksee.QueryContext
Query context interface 211

sparksee.QueryLanguage
The supported query languages 211

sparksee.QueryStream
Query stream interface 211

sparksee.QueryStreamManager
QueryStreamManager interface 213

sparksee.RandomWalk
RandomWalk class 215

sparksee.ResultSet
ResultSet class 223

sparksee.ResultSetList
ResultSet list 226

sparksee.ResultSetListIterator
ResultSetList iterator class 228

sparksee.RowReader
RowReader interface 230

sparksee.RowWriter
RowWriter interface 232

sparksee.ScriptParser
ScriptParser 234

sparksee.Session
Session class 236

sparksee.ShortestPath
ShortestPath class 239

sparksee.SinglePairShortestPath
SinglePairShortestPath class 243

sparksee.SinglePairShortestPathBFS
SinglePairShortestPathBFS class 249

sparksee.SinglePairShortestPathDijkstra
SinglePairShortestPathDijkstra class 255

sparksee.SinglePairShortestPathDijkstraDynamicCost
Defines how to calculate an edge weight 262

sparksee.Sparksee
Sparksee class 263

sparksee.SparkseeConfig
Sparksee configuration class 274

sparksee.SparkseeProperties
Sparksee properties file 301

sparksee.StringList
String list 304

sparksee.StringListIterator
StringList iterator class 306

sparksee.StrongConnectivity
StrongConnectivity class 308

sparksee.StrongConnectivityGabow
This class can be used to solve the problem of finding strongly connected components in a directed graph 313

sparksee.TextStream
TextStream class 318

sparksee.Traversal
Traversal class 321

sparksee.TraversalBFS
Breadth-First Search implementation of Traversal 326

sparksee.TraversalDFS
Depth-First Search (DFS) implementation of Traversal 332

sparksee.Type
Type data class 338

sparksee.TypeExporter
Base TypeExporter class 341

sparksee.TypeExporterEvent
Provides information about the progress of an TypeExproter instance 345

sparksee.TypeExporterListener
Interface to be implemented to receive TypeExporterEvent events from a TypeExporter 346

sparksee.TypeList
Sparksee type identifier list 347

sparksee.TypeListIterator
TypeList iterator class 349

sparksee.TypeLoader
Base TypeLoader class 350

sparksee.TypeLoaderEvent
Provides information about the progress of a TypeLoader instance 356

sparksee.TypeLoaderListener 357

sparksee.UnrecoverableError
Enum representing unrecoverable errors 357

sparksee.Value
Value class 358

sparksee.ValueArray
ValueArray class 370

sparksee.ValueList
Value list 383

sparksee.ValueListIterator 385

sparksee.Values
Value set class 386

sparksee.ValuesIterator
Values iterator class 388

sparksee.WeakConnectivity
WeakConnectivity class 390

sparksee.WeakConnectivityDFS
WeakConnectivityDFS class 395

4 Namespace Documentation

4.1 sparksee Namespace Reference

Embedded Graph Database.

Classes

- class TypeLoaderListener
- class Platform

  Platform class.
- class Query

  Query class.
- class DatabaseStatistics

  Database statistics.
- class ResultSetListIterator

  ResultSetList iterator class.
- class SinglePairShortestPathDijkstraDynamicCost

  Defines how to calculate an edge weight.
- class AttributeList

  Sparksee attribute identifier list.
- class NodeTypeExporter

  NodeTypeExporter class.
- class Values

  Value set class.
- class Traversal

  Traversal class.
- class TypeListIterator

  TypeList iterator class.
- class Int32ListIterator

  Int32List iterator class.
- class NodeExport

  Stores the node exporting values.
- class ConnectedComponents

  ConnectedComponents class.
- class TraversalBFS

  Breadth-First Search implementation of Traversal.
- class TraversalDFS

  Depth-First Search (DFS) implementation of Traversal.
- class Database

  Database class.
- class ResultSet

  ResultSet class.
- class SinglePairShortestPathBFS

  SinglePairShortestPathBFS class.
- class ShortestPath

  ShortestPath class.
- class CommunitiesSCD

  CommunitiesSCD class.
- class DisjointCommunities

  DisjointCommunities class.
- class TextStream

  TextStream class.
- class TypeList

  Sparksee type identifier list.
- class RowReader

  RowReader interface.

- class SinglePairShortestPathDijkstra
  SinglePairShortestPathDijkstra class.
- class CSVReader
  CSVReader interface.
- class CommunityDetection
  CommunityDetection class.
- class GraphExport
  Stores the graph exporting values.
- class WeakConnectivityDFS
  WeakConnectivityDFS class.
- class Int32List
  Sparksee 32-bit signed integer list.
- class WeakConnectivity
  WeakConnectivity class.
- class Graph
  Graph class.
- class TypeLoaderEvent
  Provides information about the progress of a TypeLoader instance.
- class EdgeData
- class Objects
  Object identifier set class.
- class Value
  Value class.
- class Connectivity
  Connectivity class.
- class QueryStreamManager
  QueryStreamManager interface.
- class EdgeTypeExporter
  EdgeTypeExporter class.
- class KeyValues
  Value set class.
- class BooleanList
  Boolean list.
- class StrongConnectivity
  StrongConnectivity class.
- class ObjectsIterator
  ObjectsIterator class.
- class KeyValue
- class EdgeTypeLoader
  EdgeTypeLoader class.
- class Attribute
  Attribute data class.
- class AttributeStatistics
  Attribute statistics class.
- class ValueList
  Value list.
- class ExportManager
  Defines how to export a graph to an external format.
- class DefaultExport
  Default implementation for ExportManager class.
- class CSVWriter

CSVWriter interface.

- class TypeLoader

  Base TypeLoader class.
- class DisjointCommunityDetection

  DisjointCommunityDetection class.
- class TypeExporter

  Base TypeExporter class.
- class KOpt

  KOpt class.
- class ValueListIterator
- class RowWriter

  RowWriter interface.
- class Sparksee

  Sparksee class.
- class OIDListIterator

  OIDList iterator class.
- class StrongConnectivityGabow

  This class can be used to solve the problem of finding strongly connected components in a directed graph.
- class ScriptParser

  ScriptParser.
- class SparkseeProperties

  Sparksee properties file.
- class PlatformStatistics

  Platform data and statistics.
- class Context

  Context class.
- class StringListIterator

  StringList iterator class.
- class ResultSetList

  ResultSet list.
- class StringList

  String list.
- class NodeTypeLoader

  NodeTypeLoader class.
- class OIDList

  Sparksee object identifier list.
- class PageRank

  PageRank class.
- class QueryContext

  Query context interface.
- class TypeExporterListener

  Interface to be implemented to receive TypeExporterEvent events from a TypeExporter.
- class SinglePairShortestPath

  SinglePairShortestPath class.
- class EdgeExport

  Stores edge exporting values.
- class Type

  Type data class.
- class TypeExporterEvent

  Provides information about the progress of an TypeExproter instance.
- class BooleanListIterator

BooleanList iterator class.

- class QueryStream

  Query stream interface.
- class AttributeListIterator

  AttributeList iterator class.
- class RandomWalk

  RandomWalk class.
- class SparkseeConfig

  Sparksee configuration class.
- class ValuesIterator

  Values iterator class.
- class ValueArray

  ValueArray class.
- class Session

  Session class.
- class Condition

  Condition operators enumeration.
- class Order

  Order enumeration.
- class EdgesDirection

  Edges direction enumeration.
- class ObjectType

  Object type enumeration.
- class UnrecoverableError

  Enum representing unrecoverable errors.
- class ExportType

  Export type.
- class NodeShape

  Node shape.
- class DataType

  Data type (domain) enumeration.
- class QueryLanguage

  The supported query languages.
- class AttributeKind

  Attribute kind enumeration.
- class LogLevel

  Log level enumeration.
- class MissingEndpoint

  The policy to follow whenever and edge endpoint is missing during a loading.

Functions

- load_nodes_csv (self, graph, file_name, node_type, separator, header, columns, attr_names, data_types, attr_kinds)

  Loads nodes from a CSV file.
- load_edges_csv (self, graph, file_name, edge_type, tail_node_type, head_node_type, tail, head, separator, directed, header, on_missing_tail, on_missing_head, columns, attr_names, data_types, attr_kinds)

  Loads edges from a CSV file.

4.1.1 Detailed Description

Embedded Graph Database.

Interface to be implemented to receive TypeLoaderEvent events from a TypeLoader.

Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

4.1.2 Function Documentation

load_edges_csv()

```
sparksee.load_edges_csv (
              self,
              graph,
              file_name,
              edge_type,
              tail_node_type,
              head_node_type,
              tail,
              head,
              separator,
              directed,
              header,
              on_missing_tail,
              on_missing_head,
              columns,
              attr_names,
              data_types,
              attr_kinds)
```

Loads edges from a CSV file.

nodeType[in] The type of the edge to load. If it does not exist, it creates it

Parameters

graph	[in] The graph to load the edges into
file_name	[in] The name of the file
edge_type	null
tail_node_type	[in] The type of the tail nodes. If it does not exist and onMissingTail is set to "Create", it creates it. Otherwise, an exception is thrown
head_node_type	[in] The type of the head nodes. It it does not exist and onMissingHead is set to "Create", it creates it. Otherwise, an exception is thrown
tail	[in] The tail column index. Default: 0
head	[in] The head column index. Default: 1
separator	[in] The column separator. Default: ","

directed	[in] True if this edge is directed or not. False otherwise. Default: True
header	[in] True if the CSV contains a header. False otherwise. Default: True
on_missing_tail	[in] The policy to follow when a tail is missing. Default: "IsError"
on_missing_head	[in] The policy to follow when a head is missing. Default: "IsError"
columns	[in] The list of columns to load. tail and head columns must be in this list if this list is specified. If the list is empty, all columns are loaded. Default: empty
attr_names	[in] The attribute names. If this list is empty and hasHeader is set to true, the header values are used as attribute names. Default: empty
data_types	[in] The dataTypes of the attributes if this do not already exist. If this list is empty, the method tries to infer the data type from the first non-header row. Default: empty
attr_kinds	[in] The attributeKinds of the attributes if these do not already exist. If this list is empty, the attributeKind of the created attributes are set to Basic. Default: empty

load_nodes_csv()

```
sparksee.load_nodes_csv (
              self,
              graph,
              file_name,
              node_type,
              separator,
              header,
              columns,
              attr_names,
              data_types,
              attr_kinds)
```

Loads nodes from a CSV file.

Parameters

graph	[in] The graph to load the nodes into
file_name	[in] The name of the file
node_type	[in] The type of the node to load. If it does not exist, it creates it
separator	[in] The separator of the CSV file. Default: ","
header	[in] True if the CSV contains a header. False otherwise. Default: True
columns	[in] The list of columns to load. tail and head columns mus be in this list if this list is specified. Default: all columns
attr_names	[in] The attribute names. If this list is empty and hasHeader is set to true, the header values are used as attribute names. Default: empty
data_types	[in] The dataTypes of the attributes if this do not already exist. Default: empty If this list is empty, the method tries to infer the data type from the first non-header row
attr_kinds	[in] The attributeKinds of the attributes if these do not already exist. Default: empty If this list is empty, the attributeKind of the created attributes are set to Basic. Default: empty

5 Class Documentation

5.1 sparksee.Attribute Class Reference

Attribute data class.

Public Member Functions

- get_size (self)
  Gets the number of different values.
- get_id (self)
  Gets the Sparksee attribute identifier.
- get_type_id (self)
  Gets the Sparksee type identifier.
- is_array_attribute (self)
  Check if it's an array attribute.
- get_count (self)
  Gets the number of non-NULL values.
- is_session_attribute (self)
  Check if it's a session attribute or a persistent one.
- get_data_type (self)
  Gets the data type.
- get_kind (self)
  Gets the attribute kind.
- get_default_value (self)
  Gets the default value for this attribute.
- get_array_size (self)
  Gets the number of elements in the array.
- get_name (self)
  Gets the unique attribute name.

Static Public Attributes

- int **INVALID_ATTRIBUTE** = 0
  Invalid attribute identifier constant.

5.1.1 Detailed Description

Attribute data class.

It contains information about an attribute.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.1.2 Member Function Documentation

get_array_size()

```
sparksee.Attribute.get_array_size (
            self)
```

Gets the number of elements in the array.

Returns

The size of the array

get_count()

```
sparksee.Attribute.get_count (
              self)
```

Gets the number of non-NULL values.

Returns

The number of non-NULL values.

get_data_type()

```
sparksee.Attribute.get_data_type (
              self)
```

Gets the data type.

Returns

The DataType.

get_default_value()

```
sparksee.Attribute.get_default_value (
              self)
```

Gets the default value for this attribute.

Returns

A new Value instance with the default value, or NULL if none is set.

get_id()

```
sparksee.Attribute.get_id (
              self)
```

Gets the Sparksee attribute identifier.

Returns

The Sparksee attribute identifier.

get_kind()

```
sparksee.Attribute.get_kind (
              self)
```

Gets the attribute kind.

Returns

The AttributeKind.

get_name()

```
sparksee.Attribute.get_name (
              self)
```

Gets the unique attribute name.

Returns

The unique attribute name.

get_size()

```
sparksee.Attribute.get_size (
              self)
```

Gets the number of different values.

Returns

The number of different values.

get_type_id()

```
sparksee.Attribute.get_type_id (
              self)
```

Gets the Sparksee type identifier.

Returns

The Sparksee type identifier.

is_array_attribute()

```
sparksee.Attribute.is_array_attribute (
              self)
```

Check if it's an array attribute.

Returns

True if it's an array attribute, or false otherwise.

is_session_attribute()

```
sparksee.Attribute.is_session_attribute (
              self)
```

Check if it's a session attribute or a persistent one.

Returns

True if it's a session attribute, or false otherwise.

5.2 sparksee.AttributeKind Class Reference

Attribute kind enumeration.

Static Public Attributes

- int **BASIC** = 0

  Basic attribute (non indexed attribute).
- int **INDEXED** = 1

  Indexed attribute.
- int UNIQUE = 2

  Unique attribute (indexed + unique restriction).

5.2.1 Detailed Description

Attribute kind enumeration.

It determines the indexing-capabilities of an attribute.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.2.2 Member Data Documentation

UNIQUE

```
int sparksee.AttributeKind.UNIQUE = 2  [static]
```

Unique attribute (indexed + unique restriction).

Unique restriction sets two objects cannot have the same value for an attribute but NULL.

5.3 sparksee.AttributeList Class Reference

Sparksee attribute identifier list.

Public Member Functions

- add (self, attr)

  Adds a Sparksee attribute identifier at the end of the list.
- **clear** (self)

  Clears the list.
- __iter__ (self)

  Gets a new TypeListIterator.
- iterator (self)

  Gets a new AttributeListIterator.
- __init__ (self)

  Constructor.
- count (self)

  Number of elements in the list.

5.3.1 Detailed Description

Sparksee attribute identifier list.

It stores a Sparksee attribute identifier list. Optionally it may also store tha corresponding attribute names. It can also optionally store a single set of values for each attribute.

Use AttributeListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.3.2 Constructor & Destructor Documentation

__init__()

```
sparksee.AttributeList.__init__ (
              self)
```

Constructor.

This creates an empty list.

5.3.3 Member Function Documentation

__iter__()

```
sparksee.AttributeList.__iter__ (
              self)
```

Gets a new TypeListIterator.

Returns

TypeListIterator instance

add()

```
sparksee.AttributeList.add (
              self,
              attr)
```

Adds a Sparksee attribute identifier at the end of the list.

Parameters

attr	[in] Sparksee attribute identifier.

count()

```
sparksee.AttributeList.count (
              self)
```

Number of elements in the list.

Returns

Number of elements in the list.

iterator()

```
sparksee.AttributeList.iterator (
              self)
```

Gets a new AttributeListIterator.

Returns

AttributeListIterator instance.

5.4 sparksee.AttributeListIterator Class Reference

AttributeList iterator class.

Public Member Functions

- **get_current_name** (self)

  Get the current attribute Name if it's available or an empty string otherwise.
- next (self)

  Moves to the next element.
- has_next (self)

  Gets if there are more elements.
- get_current_value (self, value)

  Get the value for the current attribute if it's available or set it Null otherwise.
- __next__ (self)

  Return the next item from the container.

5.4.1 Detailed Description

AttributeList iterator class.

Iterator to traverse all the Sparksee attribute identifier into a AttributeList instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.4.2 Member Function Documentation

__next__()

```
sparksee.AttributeListIterator.__next__ (
              self)
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

get_current_value()

```
sparksee.AttributeListIterator.get_current_value (
              self,
              value)
```

Get the value for the current attribute if it's available or set it Null otherwise.

Parameters

value	[in|out] The Value will be set to the current attribute value or as a Null Value.

has_next()

```
sparksee.AttributeListIterator.has_next (
              self)
```

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

next()

```
sparksee.AttributeListIterator.next (
              self)
```

Moves to the next element.

Returns

The next element.

5.5 sparksee.AttributeStatistics Class Reference

Attribute statistics class.

Public Member Functions

- get_min_length_string (self)
  Gets the minimum length.
- get_mode_count (self)
  Gets the number of objects with a Value equal to the mode.
- get_variance (self)
  Gets the variance.
- get_mode (self)
  Gets the mode.
- get_min (self)
  Gets the minimum existing value (BASIC statistics).
- get_median (self)
  Gets the median.
- get_total (self)
  Gets the number of objects with a non-NULL Value (BASIC statistic).
- get_max_length_string (self)
  Gets the maximum length.
- get_mean (self)
  Gets the mean or average.
- get_null (self)
  Gets the number of objects NULL a Value (BASIC statistics).
- get_distinct (self)
  Gets the number of distinct values (BASIC statistics).
- get_avg_length_string (self)
  Gets the average length.
- get_max (self)
  Gets the maximum existing value (BASIC statistics).

5.5.1 Detailed Description

Attribute statistics class.

It contains statistic data about an attribute.

Some fields are valid just for numerical attributes and others just for string attributes. Also, some statistics are considered BASIC because computing them do not require to traverse all the different values of the attribute. For each getter method the documentation tells if the statistic is BASIC or not. See the Graph class method get↩ AttributeStatistics or check out the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.5.2 Member Function Documentation

get_avg_length_string()

```
sparksee.AttributeStatistics.get_avg_length_string (
            self)
```

Gets the average length.

If the attribute is not an string attribute, it just returns 0.

Returns

The average length.

get_distinct()

```
sparksee.AttributeStatistics.get_distinct (
              self)
```

Gets the number of distinct values (BASIC statistics).

Returns

The number of distinct values.

get_max()

```
sparksee.AttributeStatistics.get_max (
              self)
```

Gets the maximum existing value (BASIC statistics).

Returns

The maximum existing value.

get_max_length_string()

```
sparksee.AttributeStatistics.get_max_length_string (
              self)
```

Gets the maximum length.

If the attribute is not an string attribute, it just returns 0.

Returns

The maximum length.

get_mean()

```
sparksee.AttributeStatistics.get_mean (
              self)
```

Gets the mean or average.

Mean or average: Sum of all Values divided by the number of observations.

It is computed just for numerical attributes.

Returns

The mean.

get_median()

```
sparksee.AttributeStatistics.get_median (
              self)
```

Gets the median.

Median: Middle value that separates the higher half from the lower.

If a $<$ b $<$ c, then the median of the list {a, b, c} is b, and if a $<$ b $<$ c $<$ d, then the median of the list {a, b, c, d} is the mean of b and c, i.e. it is (b + c)/2

It is computed just for numerical attributes.

Returns

The median.

get_min()

```
sparksee.AttributeStatistics.get_min (
              self)
```

Gets the minimum existing value (BASIC statistics).

Returns

The minimum existing value.

get_min_length_string()

```
sparksee.AttributeStatistics.get_min_length_string (
              self)
```

Gets the minimum length.

If the attribute is not an string attribute, it just returns 0.

Returns

The minimum length.

get_mode()

```
sparksee.AttributeStatistics.get_mode (
              self)
```

Gets the mode.

Mode: Most frequent Value.

Returns

The mode.

get_mode_count()

```
sparksee.AttributeStatistics.get_mode_count (
              self)
```

Gets the number of objects with a Value equal to the mode.

Returns

The number of objects with a Value equal to the mode.

get_null()

```
sparksee.AttributeStatistics.get_null (
              self)
```

Gets the number of objects NULL a Value (BASIC statistics).

Returns

The number of objects NULL a Value.

get_total()

```
sparksee.AttributeStatistics.get_total (
              self)
```

Gets the number of objects with a non-NULL Value (BASIC statistic).

Returns

The number of objects with a non-NULL Value.

get_variance()

```
sparksee.AttributeStatistics.get_variance (
              self)
```

Gets the variance.

It is computed just for numerical attributes.

Returns

The variance.

5.6 sparksee.BooleanList Class Reference

Boolean list.

Public Member Functions

- **clear** (self)

  Clears the list.
- __init__ (self)

  Constructor.
- __iter__ (self)

  Gets a new TypeListIterator.
- iterator (self)

  Gets a new BooleanListIterator.
- add (self, value)

  Adds a Boolean at the end of the list.
- count (self)

  Number of elements in the list.

5.6.1 Detailed Description

Boolean list.

It stores a Boolean list.

Use BooleanListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.6.2 Constructor & Destructor Documentation

__init__()

```
sparksee.BooleanList.__init__ (
              self)
```

Constructor.

This creates an empty list.

5.6.3 Member Function Documentation

__iter__()

```
sparksee.BooleanList.__iter__ (
              self)
```

Gets a new TypeListIterator.

Returns

TypeListIterator instance

add()

```
sparksee.BooleanList.add (
              self,
              value)
```

Adds a Boolean at the end of the list.

Parameters

value	[in] Boolean.

count()

```
sparksee.BooleanList.count (
              self)
```

Number of elements in the list.

Returns

Number of elements in the list.

iterator()

```
sparksee.BooleanList.iterator (
              self)
```

Gets a new BooleanListIterator.

Returns

BooleanListIterator instance.

5.7 sparksee.BooleanListIterator Class Reference

BooleanList iterator class.

Public Member Functions

- next (self)
  Moves to the next element.
- has_next (self)
  Gets if there are more elements.
- __next__ (self)
  Return the next item from the container.

5.7.1 Detailed Description

BooleanList iterator class.

Iterator to traverse all the strings into a BooleanList instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.7.2 Member Function Documentation

__next__()

```
sparksee.BooleanListIterator.__next__ (
              self)
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

has_next()

```
sparksee.BooleanListIterator.has_next (
              self)
```

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

next()

```
sparksee.BooleanListIterator.next (
              self)
```

Moves to the next element.

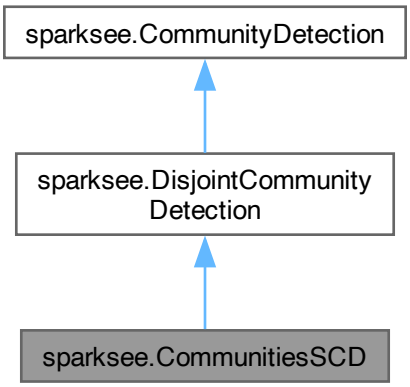
Returns

The next element.

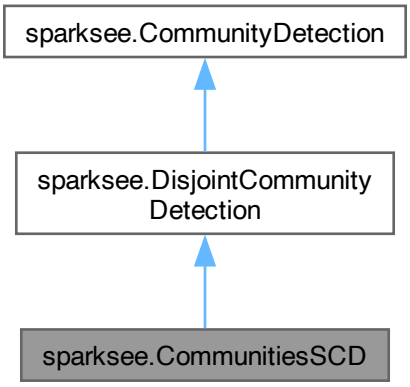
5.8 sparksee.CommunitiesSCD Class Reference

CommunitiesSCD class.

Inheritance diagram for sparksee.CommunitiesSCD:



Collaboration diagram for sparksee.CommunitiesSCD:



Public Member Functions

- __init__ (self, session)

  Creates a new instance of CommunitiesSCD.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.

- add_all_edge_types (self)

  Allows connectivity through all edge types of the graph.
- get_communities (self)

  Returns the results generated by the execution of the algorithm.
- **run** (self)

  Executes the algorithm.
- add_edge_type (self, type)

  Allows connectivity through edges of the given type.
- exclude_edges (self, edges)

  Set which edges can't be used.
- set_materialized_attribute (self, attribute_name)

  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the disjoint communities found while executing this algorithm.
- add_node_type (self, type)

  Allows connectivity through nodes of the given type.
- include_nodes (self, nodes)

  Set additional nodes that can be used.
- set_look_ahead (self, lookahead)

  Sets the size of the lookahead iterations to look (5 by default).
- **add_all_node_types** (self)

  Allows connectivity through all node types of the graph.
- include_edges (self, edges)

  Set additional edges that can be used.
- close (self)

  Closes the CommunityDetection instance.
- is_closed (self)

  Gets if CommunityDetection has been closed or not.

5.8.1 Detailed Description

CommunitiesSCD class.

Implementation of the community detection algorithm "Scalable Community Detection" based on the paper "High quality, scalable and parallel community detection for large real graphs" by Arnau Prat-Perez, David Dominguez-Sal, Josep-Lluis Larriba-Pey - WWW 2014.

The purpose of this algorithm is to find disjoint communities in an undirected graph or in a directed graph which will be considered as an undirected one.

It is possible to set some restrictions after constructing a new instance of this class and before running it in order to limit the results.

After the execution, we can retrieve the results stored in an instance of the DisjointCommunities class using the getCommunities method.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.8.2 Constructor & Destructor Documentation

__init__()

```
sparksee.CommunitiesSCD.__init__ (
              self,
              session)
```

Creates a new instance of CommunitiesSCD.

After creating this instance is required to indicate the set of edge types and the set of node types which will be navigated through while traversing the graph in order to find the communities.

Parameters

session	[in] Session to get the graph from and calculate the communities

5.8.3 Member Function Documentation

add_all_edge_types()

```
sparksee.CommunitiesSCD.add_all_edge_types (
              self)
```

Allows connectivity through all edge types of the graph.

The edges can be used in Any direction.

Reimplemented from sparksee.DisjointCommunityDetection.

add_edge_type()

```
sparksee.CommunitiesSCD.add_edge_type (
              self,
              type)
```

Allows connectivity through edges of the given type.

The edges can be used in Any direction.

Parameters

type	[in] Edge type.

Reimplemented from sparksee.DisjointCommunityDetection.

add_node_type()

```
sparksee.CommunitiesSCD.add_node_type (
              self,
              type)
```

Allows connectivity through nodes of the given type.

Parameters

type	null

Reimplemented from sparksee.DisjointCommunityDetection.

close()

```
sparksee.CommunityDetection.close (
              self)  [inherited]
```

Closes the CommunityDetection instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.CommunitiesSCD.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.DisjointCommunityDetection.

exclude_nodes()

```
sparksee.CommunitiesSCD.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.DisjointCommunityDetection.

get_communities()

```
sparksee.CommunitiesSCD.get_communities (
              self)
```

Returns the results generated by the execution of the algorithm.

These results contain information related to the disjoint communities found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class DisjointCommunities which contain information related to the disjoint communities found.

Reimplemented from sparksee.DisjointCommunityDetection.

include_edges()

```
sparksee.CommunitiesSCD.include_edges (
              self,
              edges)
```

Set additional edges that can be used.

This will replace any previously specified set of include edges. Using this optional method adds valid edges to the edges of any edge type explicitly set as a valid type. Should only be used to include specific small sets of edges because it's less efficient than just using an edge type. For any edge to be used, both nodes must be also valid.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.DisjointCommunityDetection.

include_nodes()

```
sparksee.CommunitiesSCD.include_nodes (
              self,
              nodes)
```

Set additional nodes that can be used.

This will replace any previously specified set of include nodes. Using this optional method adds valid nodes to the nodes of any node type explicitly set as a valid type. Should only be used to include specific small sets of nodes because it's less efficient than just using a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.DisjointCommunityDetection.

is_closed()

```
sparksee.CommunityDetection.is_closed (
              self)  [inherited]
```

Gets if CommunityDetection has been closed or not.

See also

close()

Returns

TRUE if the CommunityDetection instance has been closed, FALSE otherwise.

set_look_ahead()

```
sparksee.CommunitiesSCD.set_look_ahead (
              self,
              lookahead)
```

Sets the size of the lookahead iterations to look (5 by default).

Parameters

lookahead	[in] Number of iterations. It must be positive or zero.

set_materialized_attribute()

```
sparksee.CommunitiesSCD.set_materialized_attribute (
              self,
              attribute_name)
```

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the disjoint communities found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class DisjointCommunities indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the disjoint communities found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

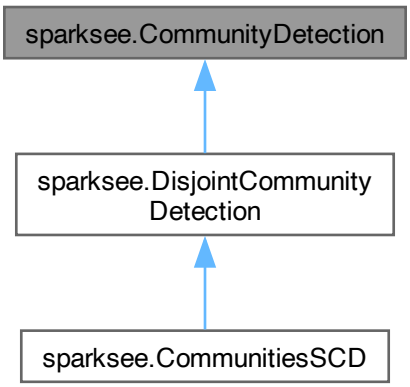
attribute_name	[in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm.

Reimplemented from sparksee.DisjointCommunityDetection.

5.9 sparksee.CommunityDetection Class Reference

CommunityDetection class.

Inheritance diagram for sparksee.CommunityDetection:



Public Member Functions

- run (self)

  Runs the algorithm in order to find the connected components.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- add_node_type (self, type)

  Allows connectivity through nodes of the given type.
- exclude_edges (self, edges)

  Set which edges can't be used.
- close (self)

  Closes the CommunityDetection instance.
- include_nodes (self, nodes)

  Set additional nodes that can be used.
- **add_all_node_types** (self)

  Allows connectivity through all node types of the graph.
- include_edges (self, edges)

  Set additional edges that can be used.
- is_closed (self)

  Gets if CommunityDetection has been closed or not.

5.9.1 Detailed Description

CommunityDetection class.

Any class implementing this abstract class can be used to solve a problem related to graph connectivity as finding the strongly connected components, finding the weakly connected components.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.9.2 Member Function Documentation

add_node_type()

```
sparksee.CommunityDetection.add_node_type (
              self,
              type)
```

Allows connectivity through nodes of the given type.

Parameters

type	null

Reimplemented in sparksee.CommunitiesSCD, and sparksee.DisjointCommunityDetection.

close()

```
sparksee.CommunityDetection.close (
              self)
```

Closes the CommunityDetection instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.CommunityDetection.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented in sparksee.CommunitiesSCD, and sparksee.DisjointCommunityDetection.

exclude_nodes()

```
sparksee.CommunityDetection.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented in sparksee.CommunitiesSCD, and sparksee.DisjointCommunityDetection.

include_edges()

```
sparksee.CommunityDetection.include_edges (
              self,
              edges)
```

Set additional edges that can be used.

This will replace any previously specified set of include edges. Using this optional method adds valid edges to the edges of any edge type explicitly set as a valid type. Should only be used to include specific small sets of edges because it's less efficient than just using an edge type. For any edge to be used, both nodes must be also valid.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented in sparksee.CommunitiesSCD, and sparksee.DisjointCommunityDetection.

include_nodes()

```
sparksee.CommunityDetection.include_nodes (
              self,
              nodes)
```

Set additional nodes that can be used.

This will replace any previously specified set of include nodes. Using this optional method adds valid nodes to the nodes of any node type explicitly set as a valid type. Should only be used to include specific small sets of nodes because it's less efficient than just using a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented in sparksee.CommunitiesSCD, and sparksee.DisjointCommunityDetection.

is_closed()

```
sparksee.CommunityDetection.is_closed (
              self)
```

Gets if CommunityDetection has been closed or not.

See also

close()

Returns

TRUE if the CommunityDetection instance has been closed, FALSE otherwise.

run()

```
sparksee.CommunityDetection.run (
              self)
```

Runs the algorithm in order to find the connected components.

This method can be called only once.

Reimplemented in sparksee.CommunitiesSCD, and sparksee.DisjointCommunityDetection.

5.10 sparksee.Condition Class Reference

Condition operators enumeration.

Static Public Attributes

- int EQUAL = 0

  Equal condition (==).
- int GREATER_EQUAL = 1

  Greater or equal condition (>=).
- int GREATER_THAN = 2

  Greater than condition (>).
- int LESS_EQUAL = 3

  Less or equal condition (<=).
- int LESS_THAN = 4

  Less than condition (<).
- int NOT_EQUAL = 5

  Not equal condition (!=).
- int LIKE = 6

  Substring condition.
- int LIKE_NO_CASE = 7

  Substring (no case sensitive) condition.
- int BETWEEN = 8

  In range operator condition ([x,y]).
- int REG_EXP = 9

  Regular expression condition.

5.10.1 Detailed Description

Condition operators enumeration.

It is mainly used in the attribute-based graph select operations.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.10.2 Member Data Documentation

BETWEEN

```
int sparksee.Condition.BETWEEN = 8  [static]
```

In range operator condition ([x,y]).

Null values cannot be used together with this condition.

EQUAL

```
int sparksee.Condition.EQUAL = 0  [static]
```

Equal condition (==).

Null values can be used together with this condition to retrieve all objects having a null value for an attribute.

GREATER_EQUAL

```
int sparksee.Condition.GREATER_EQUAL = 1  [static]
```

Greater or equal condition (>=).

Null values cannot be used together with this condition.

GREATER_THAN

```
int sparksee.Condition.GREATER_THAN = 2  [static]
```

Greater than condition (>).

Null values cannot be used together with this condition.

LESS_EQUAL

```
int sparksee.Condition.LESS_EQUAL = 3  [static]
```

Less or equal condition (<=).

Null values cannot be used together with this condition.

LESS_THAN

```
int sparksee.Condition.LESS_THAN = 4  [static]
```

Less than condition (<).

Null values cannot be used together with this condition.

LIKE

```
int sparksee.Condition.LIKE = 6  [static]
```

Substring condition.

Null values cannot be used together with this condition.

This condition can just be used together with String values. It allows for searching substrings (case sensitive). Ex:

'AAABBBCCCD' Like 'BBB' returns TRUE

'AAABBBCCCD' Like 'bbb' returns FALSE

'AAABBBCCCD' Like 'E' returns FALSE

LIKE_NO_CASE

```
int sparksee.Condition.LIKE_NO_CASE = 7  [static]
```

Substring (no case sensitive) condition.

Null values cannot be used together with this condition.

This condition can just be used together with String values. It allows for searching substrings (no case sensitive). Ex:

'AAABBBCCCD' LikeNoCase 'BBB' returns TRUE

'AAABBBCCCD' LikeNoCase 'bbb' returns TRUE

'AAABBBCCCD' LikeNoCase 'E' returns FALSE

NOT_EQUAL

```
int sparksee.Condition.NOT_EQUAL = 5  [static]
```

Not equal condition (!=).

Null values can be used together with this condition to retrieve all objects having a non-null value for an attribute.

REG_EXP

```
int sparksee.Condition.REG_EXP = 9  [static]
```

Regular expression condition.

Null values cannot be used together with this condition.

This condition can just be used together with String values.

Regular expression format conforms most of the POSIX Extended Regular Expressions so it is case sensitive.

See the 'Regular expressions' section in the 'SPARKSEE User Manual' for details.

5.11 sparksee.ConnectedComponents Class Reference

ConnectedComponents class.

Public Member Functions

- __init__ (self, s, materializedattribute)

  Creates a new instance of ConnectedComponents.
- get_size (self, id_connected_component)

  Returns the number of nodes contained in the given connected component.
- get_count (self)

  Returns the number of connected components found in the graph.
- get_connected_component (self, id_node)

  Returns the connected component where the given node belongs to.
- get_nodes (self, id_connected_component)

  Returns the collection of nodes contained in the given connected component.
- close (self)

  Closes the ConnectedComponents instance.
- is_closed (self)

  Gets if ConnectedComponents has been closed or not.

5.11.1 Detailed Description

ConnectedComponents class.

This class contains the results processed on a Connectivity algorithm.

These results contain information related to the connected components found. We must consider that each connected component has a number in order to identify it. These number identifiers are values from 0 to N-1, where N is the number of different connected components found.

When executing any implementation of the Connectivity, it is possible to indicate whether the results of the execution must be stored persistently using the class Connectivity setMaterializedAttribute method. In case the results are set to be materialized, users can retrieve this data whenever they want, even if the graph has been closed and opened again, just by creating a new instance of this class.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.11.2 Constructor & Destructor Documentation

__init__()

```
sparksee.ConnectedComponents.__init__ (
              self,
              s,
              materializedattribute)
```

Creates a new instance of ConnectedComponents.

This constructor method can only be called when a previous execution of any implementation of the Connectivity class has materialized the results in a common attribute type for all the nodes in the graph. For further information about materializing the results processed on any Connectivity execution see the documentation of the Connectivity#SetMaterializedAttribute method.

Parameters

s	[in] Session to get the graph Graph on which the information will be retrieved just by getting the values contained in the given common attribute type for all the nodes in the graph and processing them.
materializedattribute	[in] The common attribute type for all the nodes in the graph where data will be retrieved in order to process the results related to the connected components found in the graph.

5.11.3 Member Function Documentation

close()

```
sparksee.ConnectedComponents.close (
              self)
```

Closes the ConnectedComponents instance.

It must be called to ensure the integrity of all data.

get_connected_component()

```
sparksee.ConnectedComponents.get_connected_component (
              self,
              id_node)
```

Returns the connected component where the given node belongs to.

Parameters

id_node	[in] The node identifier for which the connected component identifier where it belongs will be returned.

Returns

The connected component identifier where the given node identifier belongs to.

get_count()

```
sparksee.ConnectedComponents.get_count (
              self)
```

Returns the number of connected components found in the graph.

Returns

The number of connected components found in the graph.

get_nodes()

```
sparksee.ConnectedComponents.get_nodes (
              self,
              id_connected_component)
```

Returns the collection of nodes contained in the given connected component.

Parameters

id_connected_component	The connected component for which the collection of nodes contained in it will be returned.

Returns

The collection of node identifiers contained in the given connected component.

get_size()

```
sparksee.ConnectedComponents.get_size (
              self,
              id_connected_component)
```

Returns the number of nodes contained in the given connected component.

Parameters

id_connected_component	The connected component for which the number of nodes contained in it will be returned.

Returns

The number of nodes contained in the given connected component.

is_closed()

```
sparksee.ConnectedComponents.is_closed (
            self)
```

Gets if ConnectedComponents has been closed or not.

See also

close()

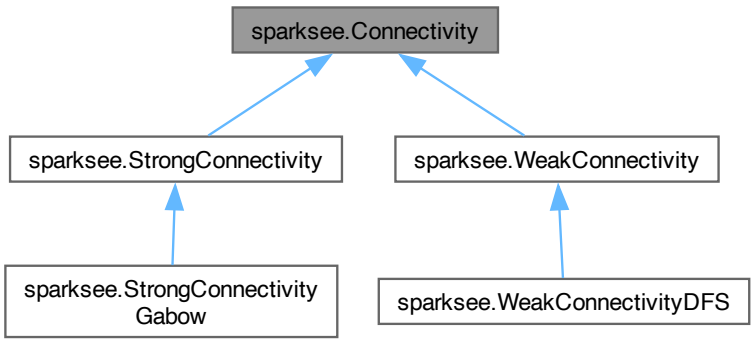
Returns

TRUE if the ConnectedComponents instance has been closed, FALSE otherwise.

5.12 sparksee.Connectivity Class Reference

Connectivity class.

Inheritance diagram for sparksee.Connectivity:



Public Member Functions

- run (self)

  Runs the algorithm in order to find the connected components.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- get_connected_components (self)

  Returns the results generated by the execution of the algorithm.
- exclude_edges (self, edges)

  Set which edges can't be used.
- add_node_type (self, t)

  Allows connectivity through nodes of the given type.
- set_materialized_attribute (self, attribute_name)

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

- close (self)

  Closes the Connectivity instance.

- **add_all_node_types** (self)

  Allows connectivity through all node types of the graph.

- is_closed (self)

  Gets if Connectivity has been closed or not.

5.12.1 Detailed Description

Connectivity class.

Any class implementing this abstract class can be used to solve a problem related to graph connectivity as finding the strongly connected components, finding the weakly connected components.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.12.2 Member Function Documentation

add_node_type()

```
sparksee.Connectivity.add_node_type (
              self,
              t)
```

Allows connectivity through nodes of the given type.

Parameters

t	null

Reimplemented in sparksee.StrongConnectivity, sparksee.StrongConnectivityGabow, sparksee.WeakConnectivity, and sparksee.WeakConnectivityDFS.

close()

```
sparksee.Connectivity.close (
              self)
```

Closes the Connectivity instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.Connectivity.exclude_edges (
            self,
            edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented in sparksee.StrongConnectivity, sparksee.StrongConnectivityGabow, sparksee.WeakConnectivity, and sparksee.WeakConnectivityDFS.

exclude_nodes()

```
sparksee.Connectivity.exclude_nodes (
            self,
            nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented in sparksee.StrongConnectivity, sparksee.StrongConnectivityGabow, sparksee.WeakConnectivity, and sparksee.WeakConnectivityDFS.

get_connected_components()

```
sparksee.Connectivity.get_connected_components (
            self)
```

Returns the results generated by the execution of the algorithm.

These results contain information related to the connected components found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class ConnectedComponents which contain information related to the connected components found.

Reimplemented in sparksee.StrongConnectivity, sparksee.StrongConnectivityGabow, sparksee.WeakConnectivity, and sparksee.WeakConnectivityDFS.

is_closed()

```
sparksee.Connectivity.is_closed (
              self)
```

Gets if Connectivity has been closed or not.

See also

close()

Returns

TRUE if the Connectivity instance has been closed, FALSE otherwise.

run()

```
sparksee.Connectivity.run (
              self)
```

Runs the algorithm in order to find the connected components.

This method can be called only once.

Reimplemented in sparksee.StrongConnectivity, sparksee.StrongConnectivityGabow, sparksee.WeakConnectivity, and sparksee.WeakConnectivityDFS.

set_materialized_attribute()

```
sparksee.Connectivity.set_materialized_attribute (
              self,
              attribute_name)
```

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class ConnectedComponents indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the connected components found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

attribute_name	[in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm.

Reimplemented in sparksee.StrongConnectivity, sparksee.StrongConnectivityGabow, sparksee.WeakConnectivity, and sparksee.WeakConnectivityDFS.

5.13 sparksee.Context Class Reference

Context class.

Public Member Functions

- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- __init__ (self, session, node)

  Creates a new instance.
- exclude_edges (self, edges)

  Set which edges can't be used.
- compute (self)

  Gets the resulting collection of nodes.
- add_edge_type (self, t, d)

  Allows for traversing edges of the given type.
- compute (self, session, node, node_types, edge_types, dir, maxhops, include)

  Helper method to easily compute a context from a node.
- set_maximum_hops (self, maxhops, include)

  Sets the maximum hops restriction.
- add_node_type (self, t)

  Allows for traversing nodes of the given type.
- close (self)

  Closes the Context instance.
- add_all_edge_types (self, d)

  Allows for traversing all edge types of the graph.
- **add_all_node_types** (self)

  Allows for traversing all node types of the graph.
- is_closed (self)

  Gets if Context has been closed or not.

5.13.1 Detailed Description

Context class.

It provides a very similar functionality than the Traversal classes. The main difference is Context returns a resulting collection whereas Traversal provides an iterator behaviour.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.13.2 Constructor & Destructor Documentation

__init__()

```
sparksee.Context.__init__ (
              self,
              session,
              node)
```

Creates a new instance.

Parameters

session	[in] Session to get the graph from and perform operation.
node	[in] Node to start traversal from.

5.13.3 Member Function Documentation

add_all_edge_types()

```
sparksee.Context.add_all_edge_types (
              self,
              d)
```

Allows for traversing all edge types of the graph.

Parameters

d	[in] Edge direction.

add_edge_type()

```
sparksee.Context.add_edge_type (
              self,
              t,
              d)
```

Allows for traversing edges of the given type.

Parameters

t	[in] Edge type.
d	[in] Edge direction.

add_node_type()

```
sparksee.Context.add_node_type (
              self,
              t)
```

Allows for traversing nodes of the given type.

Parameters

t	null

close()

```
sparksee.Context.close (
              self)
```

Closes the Context instance.

It must be called to ensure the integrity of all data.

compute() [1/2]

```
sparksee.Context.compute (
              self)
```

Gets the resulting collection of nodes.

Returns

The resulting collection of nodes.

compute() [2/2]

```
sparksee.Context.compute (
              self,
              session,
              node,
              node_types,
              edge_types,
              dir,
              maxhops,
              include)
```

Helper method to easily compute a context from a node.

Parameters

session	[in] Session to get the graph from and perform operation.

node	[in] Node to start traversal from.
node_types	[in] Allowed node type list. NULL means all node types are allowed.
edge_types	[in] Allowed edge type list. NULL means all edge types are allowed.
dir	[in] Allowed direction for the allowed edge types.
maxhops	[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.
include	[in] If TRUE, the resulting collection will include those nodes at distance less or equal than the given one, otherwise it will just include those nodes at distance equal than the given one. This parameter just makes sense if maxhops is different from 0; in that case it includes all nodes no matters the distance.

Returns

Returns an Objects with the computed context of a node.

exclude_edges()

```
sparksee.Context.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

exclude_nodes()

```
sparksee.Context.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

is_closed()

```
sparksee.Context.is_closed (
              self)
```

Gets if Context has been closed or not.

See also

close()

Returns

TRUE if the Context instance has been closed, FALSE otherwise.

set_maximum_hops()

```
sparksee.Context.set_maximum_hops (
              self,
              maxhops,
              include)
```

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

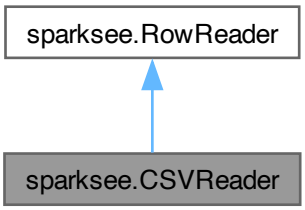
Parameters

maxhops	[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.
include	[in] If TRUE, the resulting collection will include those nodes at distance less or equal than the given one, otherwise it will just include those nodes at distance equal than the given one. This parameter just makes sense if maxhops is different from 0; in that case it includes all nodes no matters the distance.

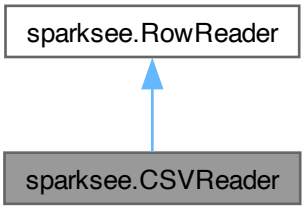
5.14 sparksee.CSVReader Class Reference

CSVReader interface.

Inheritance diagram for sparksee.CSVReader:



Collaboration diagram for sparksee.CSVReader:



Public Member Functions

- set_separator (self, sep)

  Sets the character used to separate fields in the file.
- set_num_lines (self, num_lines)

  Used to limit the number of lines that will be read.
- close (self)

  Closes the reader.
- set_locale (self, locale_str)

  Sets the locale that will be used to read the file.
- read (self, row)

  Reads the next row as a string array.
- set_multilines (self, num_extralines)

  Allows the use of fields with more than one line.
- open (self, file_path)

  Opens the source file path.
- reset (self)

  Moves the reader to the beginning.
- set_start_line (self, start_line)

  Sets the number of lines to be skiped from the beginning.
- **set_single_line** (self)

  Only allows single line fields.
- get_row (self)

  The row number for the current row.
- **__init__** (self)

  Constructs CSVReader.
- set_quotes (self, quotes)

  Sets the character used to quote fields.

5.14.1 Detailed Description

CSVReader interface.

A very simple CSV reader.

It works as any other RowReader, but open must be called once before the first read operation.

Using the format RFC 4180.

Except: leading and trailing spaces, adjacent to CSV separator character, are trimmed.

You can use your own separators and quote characters. By default the separator is the comma (,) and the quote character is the double quotes ("). Fields with multiple lines can be allowed (and the maximum lines specified), but the default is a single line. The locale string can be used to set the language, country and the file encoding. The format must be "[language_territory][.codeset]". But only the file encoding is being used in the current version. The languages supported are: "en_US", "es_ES" and "ca_ES".
 The file encodings supported are: "utf8" and "iso88591". For example: To don't change the default locales, use an empty string: "".
 To read a file in utf8 with the default language settings use ".utf8".
 To read a file in iso88591 with English language use: "en_US.iso88591".

Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.14.2 Member Function Documentation

close()

```
sparksee.CSVReader.close (
              self)
```

Closes the reader.

Exceptions

IOError	If the close fails.

Reimplemented from sparksee.RowReader.

get_row()

```
sparksee.CSVReader.get_row (
              self)
```

The row number for the current row.

Returns

The current row number; 0 if there is no current row.

Exceptions

IOError	If it fails.

Reimplemented from sparksee.RowReader.

open()

```
sparksee.CSVReader.open (
              self,
              file_path)
```

Opens the source file path.

File can be optionally compressed in GZIP format.

Parameters

file_path	[in] CSV file path.

Exceptions

IOError	If bad things happen opening the file.

read()

```
sparksee.CSVReader.read (
              self,
              row)
```

Reads the next row as a string array.

Parameters

row	[out] A string list with each comma-separated element as a separate entry.

Returns

Returns true if a row had been read or false otherwise.

Exceptions

IOError	If bad things happen during the read.

Reimplemented from sparksee.RowReader.

reset()

```
sparksee.CSVReader.reset (
              self)
```

Moves the reader to the beginning.

Restarts the reader.

Returns

true if the reader can be restarted, false otherwise.

Exceptions

IOError	If bad things happen during the restart.

Reimplemented from sparksee.RowReader.

set_locale()

```
sparksee.CSVReader.set_locale (
              self,
              locale_str)
```

Sets the locale that will be used to read the file.

Parameters

locale_str	[in] The locale string for the file encoding.

set_multilines()

```
sparksee.CSVReader.set_multilines (
              self,
              num_extralines)
```

Allows the use of fields with more than one line.

Parameters

num_extralines	[in] Maximum number of extra lines for each column (0==unlimited, N==N+1 total rows).

set_num_lines()

```
sparksee.CSVReader.set_num_lines (
              self,
              num_lines)
```

Used to limit the number of lines that will be read.

Parameters

num_lines	[in] The maximum number of lines to read (0 == unlimited)

set_quotes()

```
sparksee.CSVReader.set_quotes (
              self,
              quotes)
```

Sets the character used to quote fields.

Parameters

quotes	[in] Quote character.

Exceptions

RuntimeError	For any issue

set_separator()

```
sparksee.CSVReader.set_separator (
              self,
              sep)
```

Sets the character used to separate fields in the file.

Parameters

sep	[in] Separator character.

Exceptions

RuntimeError	For any issue

set_start_line()

```
sparksee.CSVReader.set_start_line (
              self,
              start_line)
```

Sets the number of lines to be skiped from the beginning.

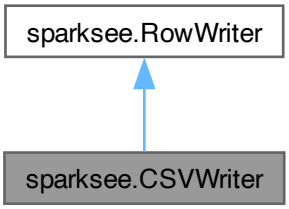
Parameters

start_line	[in] The line number to skip for start reading

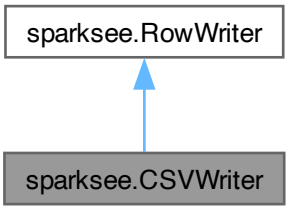
5.15 sparksee.CSVWriter Class Reference

CSVWriter interface.

Inheritance diagram for sparksee.CSVWriter:



Collaboration diagram for sparksee.CSVWriter:



Public Member Functions

- **__init__** (self)
  Creates a new instance.
- set_separator (self, sep)
  Sets the character used to separate fields in the file.
- set_auto_quotes (self, autoquotes)
  Sets on/off the automatic quote mode.
- open (self, f)
  Opens the output file path.
- write (self, row)
  Writes the next row.
- set_forced_quotes (self, forcequotes)
  Disables the automatic quote mode and forces to be quoted those positions set to TRUE in the given vector.
- close (self)
  Closes the writer.
- set_locale (self, locale_str)
  Sets the locale that will be used to write the file.
- set_quotes (self, quotes)
  Sets the character used to quote fields.

5.15.1 Detailed Description

CSVWriter interface.

A very simple CSV writer implementing RowWriter.

It works as any other RowWriter, but open must be called once before the first write operation.

It uses the format RFC 4180: http://tools.ietf.org/html/rfc4180

You can use your own separators and quote characters. By default the separator is the comma (,) and the quote character is the double quotes (") and autoquote is enabled.

See the CSVReader locale documentation or the SPARKSEE User Manual.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.15.2 Member Function Documentation

close()

```
sparksee.CSVWriter.close (
              self)
```

Closes the writer.

ErrorFor any issue

Exceptions

RuntimeError	null
IOError	If the close fails.

Reimplemented from sparksee.RowWriter.

open()

```
sparksee.CSVWriter.open (
              self,
              f)
```

Opens the output file path.

Parameters

f	[in] Output file path.

Exceptions

IOError	If bad things happen opening the file.

set_auto_quotes()

```
sparksee.CSVWriter.set_auto_quotes (
              self,
              autoquotes)
```

Sets on/off the automatic quote mode.

If there are forced quotes, setting autoquotes on will clear them. If the autoquotes is set to off and no forced quotes are provided, there will not be any quote.

Parameters

autoquotes	[in] If TRUE it enables the automatic quote mode, if FALSE it disables it.

set_forced_quotes()

```
sparksee.CSVWriter.set_forced_quotes (
              self,
              forcequotes)
```

Disables the automatic quote mode and forces to be quoted those positions set to TRUE in the given vector.

Parameters

forcequotes	[in] A booleanList with the position for each column that must be quoted set to true.

set_locale()

```
sparksee.CSVWriter.set_locale (
              self,
              locale_str)
```

Sets the locale that will be used to write the file.

Parameters

locale_str	[in] The locale string for the file encoding.

set_quotes()

```
sparksee.CSVWriter.set_quotes (
              self,
              quotes)
```

Sets the character used to quote fields.

Parameters

quotes	[in] Quote character.

Exceptions

RuntimeError	For any issue

set_separator()

```
sparksee.CSVWriter.set_separator (
              self,
              sep)
```

Sets the character used to separate fields in the file.

Parameters

sep	[in] Separator character.

Exceptions

RuntimeError	For any issue

write()

```
sparksee.CSVWriter.write (
              self,
              row)
```

Writes the next row.

ErrorFor any issue

Parameters

row	[in] Row of data.

Exceptions

RuntimeError	null
IOError	If bad things happen during the write.

Reimplemented from sparksee.RowWriter.

5.16 sparksee.Database Class Reference

Database class.

Public Member Functions

- dump_schema (self, file_path)

  Dump the database schema to the given file using the Sparksee scripting format.
- add_query_stream_manager (self, query_stream_manager)

  Add a QueryStreamManager to the database.
- new_session (self)

  Creates a new Session.
- get_path (self)

  Gets the path of the Database.
- **enable_rollback** (self)

  Enables the rollback mechanism.
- **fix_current_cache_max_size** (self)

  Sets the cache maximum size to the current cache size in use.
- redo_precommitted (self, tx_id)

  Redo a pending precommitted transaction recovered.
- get_statistics (self, stats)

  Gets Database statistics.
- set_cache_max_size (self, mega_bytes)

  Sets the cache maximum size (in MB).
- **disable_rollback** (self)

  Disables the rollback mechanism.
- close (self)

  Closes the Database instance.
- get_alias (self)

  Gets the alias of the Database.
- get_cache_max_size (self)

  Gets the cache maximum size (in MB).
- is_closed (self)

  Gets if Database instance has been closed or not.

5.16.1 Detailed Description

Database class.

All the data of the Database is stored into a persistent file which just can be created or open through a Sparksee instance.

Also, all the manipulation of a Database must be done by means of a Session which can be initiated from a Database instance.

Multiple Databases do not share the memory, that is there is no negotiation among them. In those cases, memory must be prefixed for each Database. To do that, use the SPARKSEEConfig.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.16.2 Member Function Documentation

add_query_stream_manager()

```
sparksee.Database.add_query_stream_manager (
              self,
              query_stream_manager)
```

Add a QueryStreamManager to the database.

The user is responsible of keeping the manager alive for the duration of the Database. The Managers will be used in reverse order of addition while searching for a matching Stream.

Parameters

query_stream_manager	[In] A reference to a QueryStreamManager

close()

```
sparksee.Database.close (
              self)
```

Closes the Database instance.

It must be called to ensure the integrity of all data.

dump_schema()

```
sparksee.Database.dump_schema (
              self,
              file_path)
```

Dump the database schema to the given file using the Sparksee scripting format.

Parameters

file_path	null

get_alias()

```
sparksee.Database.get_alias (
              self)
```

Gets the alias of the Database.

Returns

The alias of the Database.

get_cache_max_size()

```
sparksee.Database.get_cache_max_size (
              self)
```

Gets the cache maximum size (in MB).

Returns

Returns the current cache max size.

get_path()

```
sparksee.Database.get_path (
              self)
```

Gets the path of the Database.

Returns

The path of the Database.

get_statistics()

```
sparksee.Database.get_statistics (
              self,
              stats)
```

Gets Database statistics.

Parameters

stats	[out] The DatabaseStatistics instance.

is_closed()

```
sparksee.Database.is_closed (
              self)
```

Gets if Database instance has been closed or not.

See also

close()

Returns

TRUE if the Database instance has been closed, FALSE otherwise.

new_session()

```
sparksee.Database.new_session (
              self)
```

Creates a new Session.

Returns

A new Session instance.

redo_precommitted()

```
sparksee.Database.redo_precommitted (
              self,
              tx_id)
```

Redo a pending precommitted transaction recovered.

YOU SHOULD NOT USE THIS METHOD. This method only exists for a specific service.

Parameters

tx↩ _id	null

set_cache_max_size()

```
sparksee.Database.set_cache_max_size (
              self,
              mega_bytes)
```

Sets the cache maximum size (in MB).

0 means unlimited which is all the physical memory of the computer minus a small margin.

Parameters

mega_bytes	[in] The new cache max size.

5.17 sparksee.DatabaseStatistics Class Reference

Database statistics.

Public Member Functions

- get_read (self)
  Gets total read data in KBytes.
- get_data (self)
  Gets database size in KBytes.
- get_temp (self)
  Gets temporary storage file size in KBytes.
- get_write (self)
  Gets total written data in KBytes.
- get_cache (self)
  Gets cache size in KBytes.
- get_sessions (self)
  Gets the number of sessions.

5.17.1 Detailed Description

Database statistics.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.17.2 Member Function Documentation

get_cache()

```
sparksee.DatabaseStatistics.get_cache (
              self)
```

Gets cache size in KBytes.

Returns

Cache size in KBytes.

get_data()

```
sparksee.DatabaseStatistics.get_data (
              self)
```

Gets database size in KBytes.

Returns

Database size in KBytes.

get_read()

```
sparksee.DatabaseStatistics.get_read (
              self)
```

Gets total read data in KBytes.

Returns

Total read data in KBytes.

get_sessions()

```
sparksee.DatabaseStatistics.get_sessions (
              self)
```

Gets the number of sessions.

Returns

The number of sessions.

get_temp()

```
sparksee.DatabaseStatistics.get_temp (
              self)
```

Gets temporary storage file size in KBytes.

Returns

Temporary storage file size in KBytes.

get_write()

```
sparksee.DatabaseStatistics.get_write (
              self)
```

Gets total written data in KBytes.

Returns

Total read written in KBytes.

5.18 sparksee.DataType Class Reference

Data type (domain) enumeration.

Static Public Attributes

- int **BOOLEAN** = 0

  Boolean data type.
- int **INTEGER** = 1

  32-bit signed integer data type.
- int **LONG** = 2

  64-bit signed integer data type.
- int **DOUBLE** = 3

  64-bit signed double data type.
- int TIMESTAMP = 4

  Distance from Epoch (UTC) time in milliseconds precision.
- int STRING = 5

  Unicode string data type.
- int TEXT = 6

  Large unicode character object (CLOB) data type.
- int **OID** = 7

  Object identifier (OID) data type.

5.18.1 Detailed Description

Data type (domain) enumeration.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.18.2 Member Data Documentation

STRING

```
int sparksee.DataType.STRING = 5  [static]
```

Unicode string data type.

2048 characters maximum length.

TEXT

```
int sparksee.DataType.TEXT = 6  [static]
```

Large unicode character object (CLOB) data type.

TextStream

TIMESTAMP

```
int sparksee.DataType.TIMESTAMP = 4  [static]
```

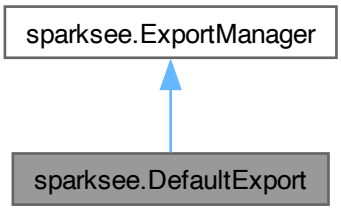
Distance from Epoch (UTC) time in milliseconds precision.

It just works properly with timestamps in the range ['1970-01-01 00:00:01' UTC, '2038-01-19 03:14:07' UTC].

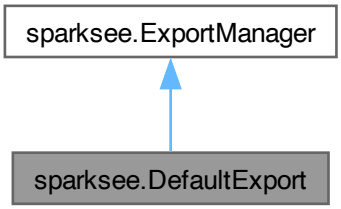
5.19 sparksee.DefaultExport Class Reference

Default implementation for ExportManager class.

Inheritance diagram for sparksee.DefaultExport:



Collaboration diagram for sparksee.DefaultExport:



Public Member Functions

- next_edge (self)

  Gets the next edge to export or InvalidOID.
- get_graph (self, graph_export)

  Default implementation of the ExportManager class method GetGraph.
- **release** (self)

Default implementation of the ExportManager class method Release.

- export_all (self)

  Get if all the graph object should be exported or only a subgraph It should return true (default option), to export all the nodes and edges in the Graph as with previous verions.
- subgraph_nodes (self, nodes)

  Limit the exported nodes to the nodes in the given node set.
- set_title_with_attributes (self, encoded)

  Include additional information including all the node/edge attributes as Title instead of the default title text (just the type name).
- set_max_text_characters (self, size)

  Sets the maximum number of Text Attribute characters to show.
- get_node (self, node, node_export)

  Default implementation of the ExportManager class method GetNode.
- get_edge (self, edge, edge_export)

  Default implementation of the ExportManager class method GetEdge.
- get_edge_type (self, type, edge_export)

  Default implementation of the ExportManager class method GetEdgeType.
- set_max_array_elements (self, size)

  Sets the maximum number of elements of an Array Attribute to show.
- set_type_label_attribute (self, type, label_attr)

  Set the attribute that stores the label for a specific node/edge type.
- subgraph_edges (self, edges)

  Limit the exported edges to the edges in the given edges set.
- prepare (self, graph)

  Default implementation of the ExportManager class method Prepare.
- next_node (self)

  Gets the next node to export or InvalidOID.
- set_node_type_image_attribute (self, node_type, image_attr)

  Set the attribute that stores the image for a specific node type.
- **__init__** (self)

  Creates a new instance.
- get_node_type (self, type, node_export)

  Default implementation of the ExportManager class method GetNodeType.
- enable_type (self, type)

  Default implementation of the ExportManager class method EnableType.
- set_type_color_rgb (self, type, color)

  Set the color for a specific node/edge type.
- set_node_type_image (self, node_type, image)

  Set an image for all the nodes of a specific node type.
- set_node_type_shape (self, node_type, shape)

  Set the shape for all nodes of this type.

5.19.1 Detailed Description

Default implementation for ExportManager class.

It uses the default values from GraphExport, NodeExport and EdgeExport to export all node and edge types.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.19.2 Member Function Documentation

enable_type()

```
sparksee.DefaultExport.enable_type (
              self,
              type)
```

Default implementation of the ExportManager class method EnableType.

This enables all node and edge types to be exported.

Parameters

type	[in] The type to enable.

Returns

TRUE.

Reimplemented from sparksee.ExportManager.

export_all()

```
sparksee.DefaultExport.export_all (
              self)
```

Get if all the graph object should be exported or only a subgraph It should return true (default option), to export all the nodes and edges in the Graph as with previous verions.

Or false to use the new Next() method of this class to get the only nodes/edges that should be exported. It will return true to export all the graph unless a subgraph had been selected by calling the SubgraphNodes or SubgraphEdges methods.

Returns

bool_t Returns true to export all the graph or false otherwise.

Reimplemented from sparksee.ExportManager.

get_edge()

```
sparksee.DefaultExport.get_edge (
              self,
              edge,
              edge_export)
```

Default implementation of the ExportManager class method GetEdge.

This sets the default EdgeExport values and sets the OID as the label. Also, it exports the edge as directed just if the edge is directed.

Parameters

edge	[in] An edge.
edge_export	[out] The EdgeExport that will store the information.

Returns

TRUE.

Reimplemented from sparksee.ExportManager.

get_edge_type()

```
sparksee.DefaultExport.get_edge_type (
              self,
              type,
              edge_export)
```

Default implementation of the ExportManager class method GetEdgeType.

This sets de default EdgeExport values.

Parameters

type	[in] An edge type.
edge_export	[out] The EdgeExport that will store the information.

Returns

TRUE.

Reimplemented from sparksee.ExportManager.

get_graph()

```
sparksee.DefaultExport.get_graph (
              self,
              graph_export)
```

Default implementation of the ExportManager class method GetGraph.

This sets the default GraphExport values including the default background color and the database alias as the label. The default GraphExport label could be empty (because the database did not have an alias) but this method will add " graph" or " subgraph" to the label anyway to easily know if the hole graph was exported or just some nodes/edges.

Parameters

graph_export	[out] The GraphExport that will store the information.

Returns

TRUE.

Reimplemented from sparksee.ExportManager.

get_node()

```
sparksee.DefaultExport.get_node (
              self,
              node,
              node_export)
```

Default implementation of the ExportManager class method GetNode.

This sets the default NodeExport values and sets the OID as the label.

Parameters

node	[in] A node.
node_export	[out] The NodeExport that will store the information.

Returns

TRUE.

Reimplemented from sparksee.ExportManager.

get_node_type()

```
sparksee.DefaultExport.get_node_type (
              self,
              type,
              node_export)
```

Default implementation of the ExportManager class method GetNodeType.

This sets de default NodeExport values.

Parameters

type	[in] A node type.
node_export	[out] The NodeExport that will store the information.

Returns

TRUE.

Reimplemented from sparksee.ExportManager.

next_edge()

```
sparksee.DefaultExport.next_edge (
              self)
```

Gets the next edge to export or InvalidOID.

It will return valid edge OIDs to export and InvalidOID when there is not any more edges to export. ExportAll returns false.

Returns

oid_t Return an edge OID or InvalidOID

Reimplemented from sparksee.ExportManager.

next_node()

```
sparksee.DefaultExport.next_node (
              self)
```

Gets the next node to export or InvalidOID.

It will return valid node OIDs to export and InvalidOID when there is not any more nodes to export. ExportAll returns false.

Returns

oid_t Return a node OID or InvalidOID

Reimplemented from sparksee.ExportManager.

prepare()

```
sparksee.DefaultExport.prepare (
              self,
              graph)
```

Default implementation of the ExportManager class method Prepare.

Parameters

graph	null

Reimplemented from sparksee.ExportManager.

set_max_array_elements()

```
sparksee.DefaultExport.set_max_array_elements (
              self,
              size)
```

Sets the maximum number of elements of an Array Attribute to show.

If an Array Attribute has more than the given size elements, only the first ones will be used and "..." will be added.

Parameters

size	null

Returns

Returns True if successful or False if the give value is not valid.

set_max_text_characters()

```
sparksee.DefaultExport.set_max_text_characters (
              self,
              size)
```

Sets the maximum number of Text Attribute characters to show.

If a Text Attribute has more than the given size characters it will be truncated and "..." will be added.

Parameters

size	null

Returns

Returns True if successful or False if the give value is not valid.

set_node_type_image()

```
sparksee.DefaultExport.set_node_type_image (
              self,
              node_type,
              image)
```

Set an image for all the nodes of a specific node type.

Parameters

node_type	[in] A valid node type.
image	[in] A path or URL of the image to use for all nodes of this type.

set_node_type_image_attribute()

```
sparksee.DefaultExport.set_node_type_image_attribute (
              self,
              node_type,
              image_attr)
```

Set the attribute that stores the image for a specific node type.

Parameters

node_type	[in] A valid node type.
image_attr	[in] The attribute to use for nodes of this type to get the image url/path.

set_node_type_shape()

```
sparksee.DefaultExport.set_node_type_shape (
              self,
              node_type,
              shape)
```

Set the shape for all nodes of this type.

Parameters

node_type	[in] A valid node type.
shape	[in] The shape for all the nodes of the given type.

set_title_with_attributes()

```
sparksee.DefaultExport.set_title_with_attributes (
              self,
              encoded)
```

Include additional information including all the node/edge attributes as Title instead of the default title text (just the type name).

Parameters

encoded	[in] If true the tooltip data will be added as a base64 encoded JSON instead of very basic html text format for easy parsing of the additional data.

set_type_color_rgb()

```
sparksee.DefaultExport.set_type_color_rgb (
              self,
              type,
              color)
```

Set the color for a specific node/edge type.

Parameters

type	[in] A valid node/edge type.
color	[in] The main color for the type. The label color will be set as a complementary color from the given color to guarantee that it will contrast.

set_type_label_attribute()

```
sparksee.DefaultExport.set_type_label_attribute (
              self,
              type,
              label_attr)
```

Set the attribute that stores the label for a specific node/edge type.

Parameters

type	[in] A valid node/edge type.
label_attr	[in] The attribute to use for nodes/edges of this type to get the label. If a valid label can be obttained from the attribute it will be used instead of the default OID.

subgraph_edges()

```
sparksee.DefaultExport.subgraph_edges (
              self,
              edges)
```

Limit the exported edges to the edges in the given edges set.

If only the subgraph edges are specified, the exported nodes will be calculated as all the nodes present in the selected edges.

Parameters

edges	[in] Subgraph edges to export. Must be kept alive by the user until the export is finished. The given Objects set can also contain nodes, but will be ignored. This is useful to use a signle Objects set to limit both subgraph nodes and edges. But if an Objects containing both nodes and edges is used, then both SubgraphNodes and SubgraphEdges methods must be called.

Returns

Returns true if successful or false otherwise.

subgraph_nodes()

```
sparksee.DefaultExport.subgraph_nodes (
              self,
              nodes)
```

Limit the exported nodes to the nodes in the given node set.

If only the subgraph nodes are specified, the exported edges will be calculated as all the edges between the selected nodes.

Parameters

nodes	[in] Subgraph nodes to export. Must be kept alive by the user until the export is finished. The given Objects set can also contain edges, but will be ignored. This is useful to use a signle Objects set to limit both subgraph nodes and edges. But if an Objects containing both nodes and edges is used, then both SubgraphNodes and SubgraphEdges methods must be called.

Returns

Returns true if successful or false otherwise.

5.20 sparksee.DisjointCommunities Class Reference

DisjointCommunities class.

Public Member Functions

- get_nodes (self, id_community)

  Returns the collection of nodes contained in the given community.
- get_community (self, id_node)

  Returns the disjoint community where the given node belongs to.
- get_count (self)

  Returns the number of communities found in the graph.
- __init__ (self, session, materializedattribute)

  Creates a new instance of DisjointCommunities.
- get_size (self, id_community)

  Returns the number of nodes contained in the given community.
- close (self)

  Closes the DisjointCommunities instance.
- is_closed (self)

  Gets if DisjointCommunities has been closed or not.

5.20.1 Detailed Description

DisjointCommunities class.

This class contains the results processed on a DisjointCommunityDetection algorithm.

These results contain information related to the communities found. We must consider that each community has a number in order to identify it. These number identifiers are values from 0 to N-1, where N is the number of different communities found.

When executing any implementation of the DisjointCommunityDetection, it is possible to indicate whether the results of the execution must be stored persistently using the class DisjointCommunityDetection setMaterializedAttribute method. In case the results are set to be materialized, users can retrieve this data whenever they want, even if the graph has been closed and opened again, just by creating a new instance of this class.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.20.2 Constructor & Destructor Documentation

__init__()

```
sparksee.DisjointCommunities.__init__ (
              self,
              session,
              materializedattribute)
```

Creates a new instance of DisjointCommunities.

This constructor method can only be called when a previous execution of any implementation of the DisjointCommunityDetection class has materialized the results in a common attribute type for all the nodes in

the graph. For further information about materializing the results processed on any DisjointCommunityDetection execution see the documentation of the DisjointCommunityDetection#SetMaterializedAttribute method.

Parameters

session	[in] Session to get the graph Graph on which the information will be retrieved just by getting the values contained in the given common attribute type for all the nodes in the graph and processing them.
materializedattribute	[in] The common attribute type for all the nodes in the graph where data will be retrieved in order to process the results related to the communities found in the graph.

5.20.3 Member Function Documentation

close()

```
sparksee.DisjointCommunities.close (
              self)
```

Closes the DisjointCommunities instance.

It must be called to ensure the integrity of all data.

get_community()

```
sparksee.DisjointCommunities.get_community (
              self,
              id_node)
```

Returns the disjoint community where the given node belongs to.

Parameters

id_node	[in] The node identifier for which the disjoint community identifier where it belongs will be returned.

Returns

The disjoint community identifier where the given node identifier belongs to.

get_count()

```
sparksee.DisjointCommunities.get_count (
              self)
```

Returns the number of communities found in the graph.

Returns

The number of communities found in the graph.

get_nodes()

```
sparksee.DisjointCommunities.get_nodes (
              self,
              id_community)
```

Returns the collection of nodes contained in the given community.

Parameters

id_community	The community for which the collection of nodes contained in it will be returned.

Returns

The collection of node identifiers contained in the given community.

get_size()

```
sparksee.DisjointCommunities.get_size (
              self,
              id_community)
```

Returns the number of nodes contained in the given community.

Parameters

id_community	The community for which the number of nodes contained in it will be returned.

Returns

The number of nodes contained in the given community.

is_closed()

```
sparksee.DisjointCommunities.is_closed (
              self)
```

Gets if DisjointCommunities has been closed or not.

See also

close()

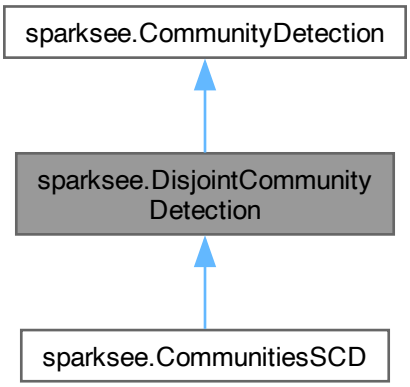
Returns

TRUE if the DisjointCommunities instance has been closed, FALSE otherwise.

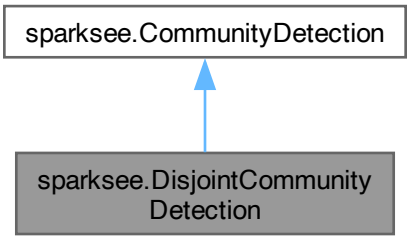
5.21 sparksee.DisjointCommunityDetection Class Reference

DisjointCommunityDetection class.

Inheritance diagram for sparksee.DisjointCommunityDetection:



Collaboration diagram for sparksee.DisjointCommunityDetection:



Public Member Functions

- run (self)

  Runs the algorithm in order to find the communities.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- add_all_edge_types (self)

  Allows connectivity through all edge types of the graph.
- get_communities (self)

  Returns the results generated by the execution of the algorithm.

- add_edge_type (self, type)

  Allows connectivity through edges of the given type.
- add_node_type (self, type)

  Allows connectivity through nodes of the given type.
- exclude_edges (self, edges)

  Set which edges can't be used.
- set_materialized_attribute (self, attribute_name)

  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the disjoint communities found while executing this algorithm.
- include_nodes (self, nodes)

  Set additional nodes that can be used.
- **add_all_node_types** (self)

  Allows connectivity through all node types of the graph.
- include_edges (self, edges)

  Set additional edges that can be used.
- close (self)

  Closes the CommunityDetection instance.
- is_closed (self)

  Gets if CommunityDetection has been closed or not.

5.21.1 Detailed Description

DisjointCommunityDetection class.

Any class implementing this abstract class can be used to solve a problem related to graph connectivity as finding the strongly connected components, finding the weakly connected components.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.21.2 Member Function Documentation

add_all_edge_types()

```
sparksee.DisjointCommunityDetection.add_all_edge_types (
              self)
```

Allows connectivity through all edge types of the graph.

The edges can be used in Any direction.

Reimplemented in sparksee.CommunitiesSCD.

add_edge_type()

```
sparksee.DisjointCommunityDetection.add_edge_type (
              self,
              type)
```

Allows connectivity through edges of the given type.

The edges can be used in Any direction.

Parameters

type	[in] Edge type.

Reimplemented in sparksee.CommunitiesSCD.

add_node_type()

```
sparksee.DisjointCommunityDetection.add_node_type (
              self,
              type)
```

Allows connectivity through nodes of the given type.

Parameters

type	null

Reimplemented from sparksee.CommunityDetection.

Reimplemented in sparksee.CommunitiesSCD.

close()

```
sparksee.CommunityDetection.close (
              self)  [inherited]
```

Closes the CommunityDetection instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.DisjointCommunityDetection.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.CommunityDetection.

Reimplemented in sparksee.CommunitiesSCD.

exclude_nodes()

```
sparksee.DisjointCommunityDetection.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.CommunityDetection.

Reimplemented in sparksee.CommunitiesSCD.

get_communities()

```
sparksee.DisjointCommunityDetection.get_communities (
              self)
```

Returns the results generated by the execution of the algorithm.

These results contain information related to the disjoint communities found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class DisjointCommunities which contain information related to the disjoint communities found.

Reimplemented in sparksee.CommunitiesSCD.

include_edges()

```
sparksee.DisjointCommunityDetection.include_edges (
              self,
              edges)
```

Set additional edges that can be used.

This will replace any previously specified set of include edges. Using this optional method adds valid edges to the edges of any edge type explicitly set as a valid type. Should only be used to include specific small sets of edges because it's less efficient than just using an edge type. For any edge to be used, both nodes must be also valid.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.CommunityDetection.

Reimplemented in sparksee.CommunitiesSCD.

include_nodes()

```
sparksee.DisjointCommunityDetection.include_nodes (
              self,
              nodes)
```

Set additional nodes that can be used.

This will replace any previously specified set of include nodes. Using this optional method adds valid nodes to the nodes of any node type explicitly set as a valid type. Should only be used to include specific small sets of nodes because it's less efficient than just using a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.CommunityDetection.

Reimplemented in sparksee.CommunitiesSCD.

is_closed()

```
sparksee.CommunityDetection.is_closed (
              self)  [inherited]
```

Gets if CommunityDetection has been closed or not.

See also

close()

Returns

TRUE if the CommunityDetection instance has been closed, FALSE otherwise.

run()

```
sparksee.DisjointCommunityDetection.run (
              self)
```

Runs the algorithm in order to find the communities.

This method can be called only once.

Reimplemented from sparksee.CommunityDetection.

Reimplemented in sparksee.CommunitiesSCD.

set_materialized_attribute()

```
sparksee.DisjointCommunityDetection.set_materialized_attribute (
              self,
              attribute_name)
```

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the disjoint communities found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class DisjointCommunities indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the disjoint communities found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

attribute_name	[in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm.

Reimplemented in sparksee.CommunitiesSCD.

5.22 sparksee.EdgeData Class Reference

Public Member Functions

- get_head (self)

  Gets the head of the edge.
- get_edge (self)

  Gets the edge identifier.
- get_tail (self)

  Gets the tail of the edge.

5.22.1 Member Function Documentation

get_edge()

```
sparksee.EdgeData.get_edge (
              self)
```

Gets the edge identifier.

Returns

The Sparksee edge identifier.

get_head()

```
sparksee.EdgeData.get_head (
              self)
```

Gets the head of the edge.

Returns

The Sparksee edge identifier of the head of the edge.

get_tail()

```
sparksee.EdgeData.get_tail (
              self)
```

Gets the tail of the edge.

Returns

The Sparksee edge identifier of the tail of the edge.

5.23 sparksee.EdgeExport Class Reference

Stores edge exporting values.

Public Member Functions

- set_as_directed (self, directed)
  Sets if the edge should be managed as directed.
- get_labelcolor_rgb (self)
  Gets the edge label color.
- set_label (self, label)
  Sets the edge label.
- set_width (self, width)
  Sets the edge width.
- set_color_rgb (self, color)
  Sets the edge color.
- set_font_size (self, size)
  Sets the edge label font size.
- set_labelcolor_rgb (self, color)
  Sets the edge label color.
- get_color_rgb (self)
  Gets the edge color.
- get_font_size (self)
  Gets the edge label font size.
- **set_defaults** (self)
  Sets to default values.
- **__init__** (self)
  Creates a new instance.
- get_label (self)
  Gets the edge label.
- get_title (self)
  Gets the edge title.
- get_width (self)
  Gets the edge width.
- set_title (self, title)
  Sets the edge title.
- as_directed (self)
  Gets if the edge should be managed as directed.
- copy_defaults (self, source)
  Sets the default values using a refrerence EdgeExport.

5.23.1 Detailed Description

Stores edge exporting values.

Some properties may be ignored depending on the exportation type.

Default values are:

Label: "" (empty string).

As directed: TRUE.

Color: 13882323 (OxD3D3D3, Light gray).

Label color: 0 (Ox000000, Black).

Width: 5px.

Font size: 10.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.23.2 Member Function Documentation

as_directed()

```
sparksee.EdgeExport.as_directed (
              self)
```

Gets if the edge should be managed as directed.

TRUE is the default value. If TRUE, use as directed, otherwise use as undirected.

Returns

The edge direction.

copy_defaults()

```
sparksee.EdgeExport.copy_defaults (
              self,
              source)
```

Sets the default values using a refrerence EdgeExport.

Parameters

source	null

get_color_rgb()

```
sparksee.EdgeExport.get_color_rgb (
              self)
```

Gets the edge color.

Returns

The edge color.

get_font_size()

```
sparksee.EdgeExport.get_font_size (
              self)
```

Gets the edge label font size.

Returns

The edge label font size.

get_label()

```
sparksee.EdgeExport.get_label (
              self)
```

Gets the edge label.

Returns

The edge label.

get_labelcolor_rgb()

```
sparksee.EdgeExport.get_labelcolor_rgb (
              self)
```

Gets the edge label color.

Returns

The edge label color.

get_title()

```
sparksee.EdgeExport.get_title (
              self)
```

Gets the edge title.

Returns

The edge title.

get_width()

```
sparksee.EdgeExport.get_width (
              self)
```

Gets the edge width.

Returns

The edge width.

set_as_directed()

```
sparksee.EdgeExport.set_as_directed (
              self,
              directed)
```

Sets if the edge should be managed as directed.

Parameters

directed	[in] If TRUE, use as directed, otherwise use as undirected.

set_color_rgb()

```
sparksee.EdgeExport.set_color_rgb (
              self,
              color)
```

Sets the edge color.

Parameters

color	[in] The edge color.

set_font_size()

```
sparksee.EdgeExport.set_font_size (
              self,
              size)
```

Sets the edge label font size.

Parameters

size	[in] The edge label font size.

set_label()

```
sparksee.EdgeExport.set_label (
              self,
              label)
```

Sets the edge label.

Parameters

label	[in] The edge label.

set_labelcolor_rgb()

```
sparksee.EdgeExport.set_labelcolor_rgb (
              self,
              color)
```

Sets the edge label color.

Parameters

color	[in] The edge label color.

set_title()

```
sparksee.EdgeExport.set_title (
              self,
              title)
```

Sets the edge title.

Parameters

title	[in] The edge title.

set_width()

```
sparksee.EdgeExport.set_width (
              self,
              width)
```

Sets the edge width.

Parameters

width	[in] The edge width.

5.24 sparksee.EdgesDirection Class Reference

Edges direction enumeration.

Static Public Attributes

- int **INGOING** = 0

  In-going edges.
- int **OUTGOING** = 1

  Out-going edges.
- int **ANY** = 2

  In-going or out-going edges.

5.24.1 Detailed Description

Edges direction enumeration.

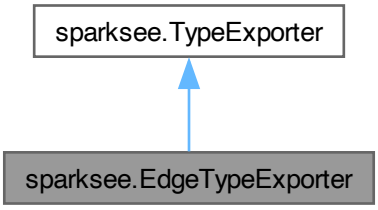
Author

Sparsity Technologies http://www.sparsity-technologies.com

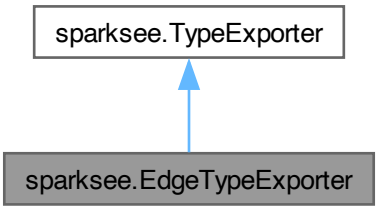
5.25 sparksee.EdgeTypeExporter Class Reference

EdgeTypeExporter class.

Inheritance diagram for sparksee.EdgeTypeExporter:



Collaboration diagram for sparksee.EdgeTypeExporter:



Public Member Functions

- run (self)

  See the TypeExporter class Run method.
- set_type (self, type)

  Sets the type to be exported.
- set_head_position (self, pos)

Sets the position (index column) of the head attribute in the exported data.

- set_attributes (self, attrs)

  Sets the list of Attributes.
- set_frequency (self, freq)

  Sets the frequency of listener notification.
- set_head_attribute (self, attr)

  Sets the attribute that will be used to get the value to be dumped for the head of the edge.
- set_tail_attribute (self, attr)

  Sets the attribute that will be used to get the value to be dumped for the tail of the edge.
- set_tail_position (self, pos)

  Sets the position (index column) of the tail attribute in the exported data.
- **__init__** (self)

  Creates a new instance.
- set_header (self, header)

  Sets the presence of a header row.
- set_row_writer (self, rw)

  Sets the output data destination.
- set_graph (self, graph)

  Sets the graph that will be exported.
- __init__ (self, row_writer, graph, type, attrs, h_pos, t_pos, h_attr, t_attr)

  Creates a new instance.
- register (self, tel)

  Registers a new listener.

5.25.1 Detailed Description

EdgeTypeExporter class.

Specific TypeExporter implementation for edge types.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.25.2 Constructor & Destructor Documentation

__init__()

```
sparksee.EdgeTypeExporter.__init__ (
            self,
            row_writer,
            graph,
            type,
            attrs,
            h_pos,
            t_pos,
            h_attr,
            t_attr)
```

Creates a new instance.

Parameters

row_writer	[in] Output RowWriter.
graph	[in] Graph.
type	[in] Type identifier.
attrs	[in] Attribute identifiers to be exported.
h_pos	[in] The position (index column) for the head value.
t_pos	[in] The position (index column) for the tail value.
h_attr	[in] The attribute identifier to get the value to be dumped for the head.
t_attr	[in] The attribute identifier to get the value to be dumped for the tail.

5.25.3 Member Function Documentation

register()

```
sparksee.EdgeTypeExporter.register (
              self,
              tel)
```

Registers a new listener.

Parameters

tel	[in] TypeExporterListener to be registered.

Reimplemented from sparksee.TypeExporter.

run()

```
sparksee.EdgeTypeExporter.run (
              self)
```

See the TypeExporter class Run method.

WrongArgumentErrorIn case of bad settings ErrorFor other issues

Exceptions

RuntimeError	null
ValueError	null
IOError	If bad things happen writting to the RowWriter.

Reimplemented from sparksee.TypeExporter.

set_attributes()

```
sparksee.EdgeTypeExporter.set_attributes (
              self,
              attrs)
```

Sets the list of Attributes.

Parameters

attrs	[in] Attribute identifiers to be exported

Reimplemented from sparksee.TypeExporter.

set_frequency()

```
sparksee.EdgeTypeExporter.set_frequency (
              self,
              freq)
```

Sets the frequency of listener notification.

Parameters

freq	[in] Frequency in number of rows managed to notify progress to all listeners

Reimplemented from sparksee.TypeExporter.

set_graph()

```
sparksee.EdgeTypeExporter.set_graph (
              self,
              graph)
```

Sets the graph that will be exported.

Parameters

graph	[in] Graph.

Reimplemented from sparksee.TypeExporter.

set_head_attribute()

```
sparksee.EdgeTypeExporter.set_head_attribute (
              self,
              attr)
```

Sets the attribute that will be used to get the value to be dumped for the head of the edge.

Parameters

attr	[in] Head Attribute

set_head_position()

```
sparksee.EdgeTypeExporter.set_head_position (
              self,
              pos)
```

Sets the position (index column) of the head attribute in the exported data.

Parameters

pos	[in] Head position

set_header()

```
sparksee.EdgeTypeExporter.set_header (
              self,
              header)
```

Sets the presence of a header row.

Parameters

header	[in] If TRUE, a header row is dumped with the name of the attributes.

Reimplemented from sparksee.TypeExporter.

set_row_writer()

```
sparksee.EdgeTypeExporter.set_row_writer (
              self,
              rw)
```

Sets the output data destination.

Parameters

rw	[in] Input RowWriter.

Reimplemented from sparksee.TypeExporter.

set_tail_attribute()

```
sparksee.EdgeTypeExporter.set_tail_attribute (
              self,
              attr)
```

Sets the attribute that will be used to get the value to be dumped for the tail of the edge.

Parameters

attr	[in] Tail Attribute

set_tail_position()

```
sparksee.EdgeTypeExporter.set_tail_position (
              self,
              pos)
```

Sets the position (index column) of the tail attribute in the exported data.

Parameters

pos	[in] Tail position

set_type()

```
sparksee.EdgeTypeExporter.set_type (
              self,
              type)
```

Sets the type to be exported.

Parameters

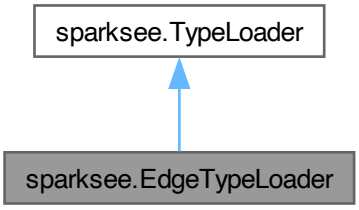
type	[in] Type identifier.

Reimplemented from sparksee.TypeExporter.

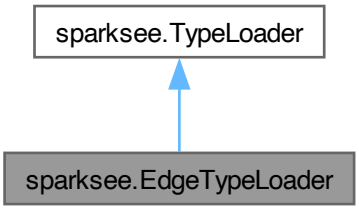
5.26 sparksee.EdgeTypeLoader Class Reference

EdgeTypeLoader class.

Inheritance diagram for sparksee.EdgeTypeLoader:



Collaboration diagram for sparksee.EdgeTypeLoader:



Public Member Functions

- run (self)

  See the TypeLoader class Run method.
- set_log_error (self, path)

  Sets a log error file.
- set_type (self, type)

  Sets the type to be loaded.
- set_head_position (self, pos)

  Sets the position of the head attribute in the source data.
- set_locale (self, locale_str)

  Sets the locale that will be used to read the data.
- set_timestamp_format (self, timestamp_format)

  Sets a specific timestamp format.
- set_attributes (self, attrs)

  Sets the list of Attributes.

- **__init__** (self)

  Creates a new instance.
- register (self, tel)

  Registers a new listener.
- __init__ (self, row_reader, graph, type, attrs, attrs_pos, h_pos, t_pos, h_attr, t_attr, head_mep, tail_mep)

  Creates a new instance.
- set_frequency (self, freq)

  Sets the frequency of listener notification.
- run_two_phases (self)

  See the TypeLoader class RunTwoPhases method.
- set_head_attribute (self, attr)

  Sets the attribute that will be used to find the head of the edge.
- set_tail_attribute (self, attr)

  Sets the attribute that will be used to find the tail of the edge.
- set_tail_position (self, pos)

  Sets the position of the tail attribute in the source data.
- set_row_reader (self, rr)

  Sets the input data source.
- set_tail_mep (self, mep)

  Sets the flag to create missing tail nodes when loading nodes or edges.
- set_attribute_positions (self, attrs_pos)

  Sets the list of attribute positions.
- set_graph (self, graph)

  Sets the graph where the data will be loaded.
- run_n_phases (self, partitions)

  See the TypeLoader class RunNPhases method.
- set_head_mep (self, mep)

  Sets the flag to create missing head nodes when loading nodes or edges.
- set_log_off (self)

  Truns off all the error reporting.

5.26.1 Detailed Description

EdgeTypeLoader class.

Specific TypeLoader implementation for edge types.

Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.26.2 Constructor & Destructor Documentation

__init__()

```
sparksee.EdgeTypeLoader.__init__ (
              self,
              row_reader,
              graph,
              type,
              attrs,
              attrs_pos,
              h_pos,
              t_pos,
              h_attr,
              t_attr,
              head_mep,
              tail_mep)
```

Creates a new instance.

Parameters

row_reader	[in] Input RowReader.
graph	[in] Graph.
type	[in] Type identifier.
attrs	[in] Attribute identifiers to be loaded.
attrs_pos	[in] Attribute positions (column index >=0). to all listeners.
h_pos	[in] The position (index column) for the head value.
t_pos	[in] The position (index column) for the tail value.
h_attr	[in] The attribute identifier for the head.
t_attr	[in] The attribute identifier for the tail.
head_mep	[in] The MissingEndpoint policy for the head nodes
tail_mep	[in] The MissingEndpoint policy for the tail nodes

5.26.3 Member Function Documentation

register()

```
sparksee.EdgeTypeLoader.register (
              self,
              tel)
```

Registers a new listener.

Parameters

tel	TypeLoaderListener to be registered.

Reimplemented from sparksee.TypeLoader.

run()

```
sparksee.EdgeTypeLoader.run (
              self)
```

See the TypeLoader class Run method.

ErrorFor other issues

Exceptions

RuntimeError	null
IOError	If bad things happen using to the RowReader.

Reimplemented from sparksee.TypeLoader.

run_n_phases()

```
sparksee.EdgeTypeLoader.run_n_phases (
              self,
              partitions)
```

See the TypeLoader class RunNPhases method.

ErrorFor other issues

Parameters

partitions	null

Exceptions

RuntimeError	null
IOError	If bad things happen using to the RowReader.

Reimplemented from sparksee.TypeLoader.

run_two_phases()

```
sparksee.EdgeTypeLoader.run_two_phases (
              self)
```

See the TypeLoader class RunTwoPhases method.

ErrorFor other issues

Exceptions

RuntimeError	null
IOError	If bad things happen using to the RowReader.

Reimplemented from sparksee.TypeLoader.

set_attribute_positions()

```
sparksee.EdgeTypeLoader.set_attribute_positions (
              self,
              attrs_pos)
```

Sets the list of attribute positions.

Parameters

attrs_pos	[in] Attribute positions (column index >=0).

Reimplemented from sparksee.TypeLoader.

set_attributes()

```
sparksee.EdgeTypeLoader.set_attributes (
              self,
              attrs)
```

Sets the list of Attributes.

Parameters

attrs	[in] Attribute identifiers to be loaded

Reimplemented from sparksee.TypeLoader.

set_frequency()

```
sparksee.EdgeTypeLoader.set_frequency (
              self,
              freq)
```

Sets the frequency of listener notification.

Parameters

freq	[in] Frequency in number of rows managed to notify progress to all listeners

Reimplemented from sparksee.TypeLoader.

set_graph()

```
sparksee.EdgeTypeLoader.set_graph (
              self,
              graph)
```

Sets the graph where the data will be loaded.

Parameters

graph	[in] Graph.

Reimplemented from sparksee.TypeLoader.

set_head_attribute()

```
sparksee.EdgeTypeLoader.set_head_attribute (
              self,
              attr)
```

Sets the attribute that will be used to find the head of the edge.

This method is protected because only the Edge loaders should have it.

Parameters

attr	[in] Head Attribute

set_head_mep()

```
sparksee.EdgeTypeLoader.set_head_mep (
              self,
              mep)
```

Sets the flag to create missing head nodes when loading nodes or edges.

flagTrue if the nodes need to be created. False otherwise

Parameters

mep	null

set_head_position()

```
sparksee.EdgeTypeLoader.set_head_position (
              self,
              pos)
```

Sets the position of the head attribute in the source data.

This method is protected because only the Edge loaders should have it.

Parameters

pos	[in] Head position

set_locale()

```
sparksee.EdgeTypeLoader.set_locale (
              self,
              locale_str)
```

Sets the locale that will be used to read the data.

It should match the locale used in the rowreader.

Parameters

locale_str	[in] The locale string for the read data. See CSVReader.

Reimplemented from sparksee.TypeLoader.

set_log_error()

```
sparksee.EdgeTypeLoader.set_log_error (
              self,
              path)
```

Sets a log error file.

By default errors are thrown as a exception and the load process ends. If a log file is set, errors are logged there and the load process does not stop.

Parameters

path	[in] The path to the error log file.

Exceptions

IOError	If bad things happen opening the file.

Reimplemented from sparksee.TypeLoader.

set_log_off()

```
sparksee.EdgeTypeLoader.set_log_off (
              self)
```

Truns off all the error reporting.

The log file will not be created and no exceptions for invalid data will be thrown. If you just want to turn off the logs, but abort at the first error what you should do is not call this method and not set a logError file.

Reimplemented from sparksee.TypeLoader.

set_row_reader()

```
sparksee.EdgeTypeLoader.set_row_reader (
              self,
              rr)
```

Sets the input data source.

Parameters

rr	[in] Input RowReader.

Reimplemented from sparksee.TypeLoader.

set_tail_attribute()

```
sparksee.EdgeTypeLoader.set_tail_attribute (
              self,
              attr)
```

Sets the attribute that will be used to find the tail of the edge.

This method is protected because only the Edge loaders should have it.

Parameters

attr	[in] Tail Attribute

set_tail_mep()

```
sparksee.EdgeTypeLoader.set_tail_mep (
              self,
              mep)
```

Sets the flag to create missing tail nodes when loading nodes or edges.

flagTrue if the nodes need to be created. False otherwise

Parameters

mep	null

set_tail_position()

```
sparksee.EdgeTypeLoader.set_tail_position (
              self,
              pos)
```

Sets the position of the tail attribute in the source data.

This method is protected because only the Edge loaders should have it.

Parameters

pos	[in] Tail position

set_timestamp_format()

```
sparksee.EdgeTypeLoader.set_timestamp_format (
              self,
              timestamp_format)
```

Sets a specific timestamp format.

Parameters

timestamp_format	[in] A string with the timestamp format definition.

Reimplemented from sparksee.TypeLoader.

set_type()

```
sparksee.EdgeTypeLoader.set_type (
              self,
              type)
```

Sets the type to be loaded.

Parameters

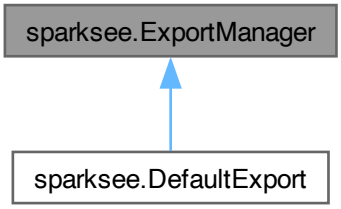
type	[in] Type identifier.

Reimplemented from sparksee.TypeLoader.

5.27 sparksee.ExportManager Class Reference

Defines how to export a graph to an external format.

Inheritance diagram for sparksee.ExportManager:



Public Member Functions

- get_edge (self, edge, edge_export)
  Gets the edge export definition for the given edge.
- release (self)
  Ends the export process.
- get_graph (self, graph_export)
  Gets the graph export definition.
- next_edge (self)
  Gets the next edge to export or InvalidOID.
- get_node (self, node, node_export)
  Gets the node export definition for the given node.
- next_node (self)
  Gets the next node to export or InvalidOID.
- get_node_type (self, type, node_export)
  Gets the default node export definition for the given node type.
- export_all (self)

Get if all the graph object should be exported or only a subgraph It should return true (default option), to export all the nodes and edges in the Graph as with previous verions.

- enable_type (self, type)

  Gets whether a node or edge type must be exported or not.
- prepare (self, graph)

  Prepares the graph for the export process.
- get_edge_type (self, type, edge_export)

  Gets the default node export definition for the given edge type.

5.27.1 Detailed Description

Defines how to export a graph to an external format.

This is an interface which must be implemented by the user. While the export proces, a call for each node or edge type and node or edge object is done to get how to export that element.

It is possible to export a Graph to a diferent fortmats. Nowadays, available formats are defined in the ExportType enum.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.27.2 Member Function Documentation

enable_type()

```
sparksee.ExportManager.enable_type (
              self,
              type)
```

Gets whether a node or edge type must be exported or not.

Parameters

type	Node or edge type identifier.

Returns

If TRUE all objects of the given type will be exported, otherwise they will not be exported.

Reimplemented in sparksee.DefaultExport.

export_all()

```
sparksee.ExportManager.export_all (
              self)
```

Get if all the graph object should be exported or only a subgraph It should return true (default option), to export all the nodes and edges in the Graph as with previous verions.

Or false to use the new NextNode() and NextEdge() methods of this class to get the only nodes/edges that should be exported.

Returns

bool_t Returns true to export all the graph or false otherwise.

Reimplemented in sparksee.DefaultExport.

get_edge()

```
sparksee.ExportManager.get_edge (
              self,
              edge,
              edge_export)
```

Gets the edge export definition for the given edge.

Parameters

edge	Edge identifier.
edge_export	[out] The EdgeExport which defines how to export given edge.

Returns

TRUE if the given EdgeExport has been updated, otherwise FALSE will be returned and the default EdgeExport for the type the edge belongs to will be used.

Reimplemented in sparksee.DefaultExport.

get_edge_type()

```
sparksee.ExportManager.get_edge_type (
              self,
              type,
              edge_export)
```

Gets the default node export definition for the given edge type.

GetEdge has a higher priority than this function. That is, only if GetEdge returns FALSE, the EdgeExport of this function will be used.

Parameters

type	[in] Edge type identifier.
edge_export	[out] The EdgeExport which defines how to export the edges of the given type.

Returns

TRUE.

Reimplemented in sparksee.DefaultExport.

get_graph()

```
sparksee.ExportManager.get_graph (
              self,
              graph_export)
```

Gets the graph export definition.

Parameters

graph_export	[out] The GraphExport which defines how to export the graph.

Returns

TRUE.

Reimplemented in sparksee.DefaultExport.

get_node()

```
sparksee.ExportManager.get_node (
              self,
              node,
              node_export)
```

Gets the node export definition for the given node.

Parameters

node	Node identifier.
node_export	[out] The NodeExport which defines how to export given node.

Returns

TRUE if the given NodeExport has been updated, otherwise FALSE will be returned and the default NodeExport for the type the node belongs to will be used.

Reimplemented in sparksee.DefaultExport.

get_node_type()

```
sparksee.ExportManager.get_node_type (
              self,
              type,
              node_export)
```

Gets the default node export definition for the given node type.

GetNode has a higher priority than this function. That is, only if GetNode returns FALSE, the NodeExport of this function will be used.

Parameters

type	[in] Node type identifier.
node_export	[out] The NodeExport which defines how to export the nodes of the given type.

Returns

TRUE.

Reimplemented in sparksee.DefaultExport.

next_edge()

```
sparksee.ExportManager.next_edge (
              self)
```

Gets the next edge to export or InvalidOID.

It will return valid edge OIDs to export and InvalidOID when there is not any more edges to export. ExportAll returns false.

Returns

oid_t Return an edge OID or InvalidOID

Reimplemented in sparksee.DefaultExport.

next_node()

```
sparksee.ExportManager.next_node (
              self)
```

Gets the next node to export or InvalidOID.

It will return valid node OIDs to export and InvalidOID when there is not any more nodes to export. ExportAll returns false.

Returns

oid_t Return a node OID or InvalidOID

Reimplemented in sparksee.DefaultExport.

prepare()

```
sparksee.ExportManager.prepare (
              self,
              graph)
```

Prepares the graph for the export process.

It is called once before the export process.

Parameters

graph	Graph to be exported.

Reimplemented in sparksee.DefaultExport.

release()

```
sparksee.ExportManager.release (
              self)
```

Ends the export process.

It is called once after the export process.

Reimplemented in sparksee.DefaultExport.

5.28 sparksee.ExportType Class Reference

Export type.

Static Public Attributes

- int GRAPHVIZ = 0

  Export to Graphviz format.
- int GRAPHML = 1

  Export to GraphML format.
- int YGRAPHML = 2

  Export to YGRAPHML format.
- int HTML = 3

  Export to HTML format.

5.28.1 Detailed Description

Export type.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.28.2 Member Data Documentation

GRAPHML

```
int sparksee.ExportType.GRAPHML = 1  [static]
```

Export to GraphML format.

Deprectated! GraphML home page: http://graphml.graphdrawing.org/

GRAPHVIZ

```
int sparksee.ExportType.GRAPHVIZ = 0  [static]
```

Export to Graphviz format.

This "dot" file format is the recommended export format. Graphviz home page: http://www.graphviz.org

HTML

```
int sparksee.ExportType.HTML = 3  [static]
```

Export to HTML format.

Simple html export for quick visualization of a small graph. Directly export to HTML using vis.js (https://visjs.github.io/vis-network/docs/network/index.html).

YGRAPHML

```
int sparksee.ExportType.YGRAPHML = 2  [static]
```

Export to YGRAPHML format.

Deprectated! It is an GraphML format extended with a set of yWorks ("http://www.yworks.com") extensions. Thus, it allows for the visualization of the exported graph with the public yEd visualization tool ("http://www.yworks.com/products/yed").

5.29 sparksee.Graph Class Reference

Graph class.

Public Member Functions

- find_or_create_edge (self, etype, tail, head)

  Gets any of the edges of the specified type between two given nodes (tail and head).
- find_edge_types (self)

  Gets all existing Sparksee edge type identifiers.
- get_attribute_interval_count (self, attr, lower, include_lower, higher, include_higher)

  Gets how many objects have a value into the given range for the given attribute.
- neighbors (self, objs, etype, dir)

  Selects all neighbor nodes from or to each of the node OID in the given collection and for the given edge type.
- backup (self, file)

  Dumps all the data to a backup file.
- remove_attribute (self, attr)

  Removes the given attribute.
- remove_type (self, type)

  Removes the given type.
- degree (self, oid, etype, dir)

  Gets the number of edges from or to the given node OID and for the given edge type.
- find_types (self)

  Gets all existing Sparksee node and edge type identifiers.
- new_array_attribute (self, type, name, dt, size)

  Creates a new array attribute.
- find_type (self, name)

  Gets the Sparksee type identifier for the given type name.
- top_k (self, attribute, operation, lower, order, k)

  Gets a KeyValues iterator as a result of the TopK operation.
- get_edge_peer (self, edge, node)

  Gets the other end for the given edge.
- set_attribute_text (self, oid, attr, tstream)

  Sets the writable TextStream for the given text attribute and OID.
- set_attribute (self, attr, value)

  Sets the value of the given attribute for all the objects of the types the attribute applies to.
- index_neighbors (self, edge_type, neighbors)

  Creates or destroys the neighbors index of an edge type.
- get_object_type (self, oid)

  Gets the Sparksee node or edge type identifier for the given OID.
- index_attribute (self, attr, kind)

  Updates the index of the given attribute.
- set_array_attribute (self, attr, value)

  Sets all the values of the array of the given array attribute for all the objects of the types the attribute applies to.
- top_k (self, attribute, order, k)

  Gets a KeyValues iterator as a result of the TopK operation.
- get_attribute (self, oid, attr, value)

  Gets the Value for the given attribute and OID.
- get_attribute_statistics (self, attr, basic)

  Gets statistics from the given attribute.
- select (self, attr, cond, lower, higher, restriction)

  Selects all OIDs satisfying the given condition for the given attribute.
- count_nodes (self)

  Gets the number of nodes.
- top_k (self, attribute, operation, lower, order, k, restriction)

Gets a KeyValues iterator as a result of the TopK operation.

- set_attribute_default_value (self, attr, value)

  Sets a default value for an attribute.
- find_object (self, attr, value)

  Finds one object having the given Value for the given attribute.
- neighbors (self, oid, etype, dir)

  Selects all neighbor nodes from or to the given node OID and for the given edge type.
- new_session_attribute (self, type, dt, kind)

  Creates a new Session attribute.
- encrypted_backup (self, file, key_in_hex, iv_in_hex)

  Dumps all the data to a backup file.
- drop (self, oid)

  Drops the given OID.
- set_array_attribute_void (self, oid, attr, value)

  Sets all the elements of the ValueArray for the given array attribute and OID.
- edges (self, etype, tail, head)

  Gets all the edges of the given type between two given nodes (tail and head).
- set_attribute (self, oid, attr, value)

  Sets the Value for the given attribute and OID.
- find_edge (self, etype, tail, head)

  Gets any of the edges of the given type between two given nodes (tail and head).
- tails (self, edges)

  Gets all the tails from the given edges collection.
- select (self, attr, cond, lower, higher)

  Selects all OIDs satisfying the given condition for the given attribute.
- get_values (self, attr)

  Gets the Value collection for the given attribute.
- export (self, file, type, em)

  Exports the Graph.
- new_restricted_edge_type (self, name, tail, head, neighbors)

  Creates a new restricted edge type.
- get_attribute_text (self, oid, attr)

  Gets the read-only TextStream for the given text attribute and OID.
- explode (self, objs, etype, dir)

  Selects all edges from or to each of the node OID in the given collection and for the given edge type.
- new_edge_type (self, name, directed, neighbors)

  Creates a new edge type.
- top_k (self, attribute, operation, lower, higher, order, k)

  Gets a KeyValues iterator as a result of the TopK operation.
- top_k (self, attribute, order, k, restriction)

  Gets a KeyValues iterator as a result of the TopK operation.
- dump_data (self, file)

  Dumps logical data to a file.
- set_array_attribute (self, oid, attr, value)

  Sets all the elements of the ValueArray for the given array attribute and OID and returns it.
- top_k (self, attribute, operation, lower, higher, order, k, restriction)

  Gets a KeyValues iterator as a result of the TopK operation.
- find_attribute (self, type, name)

  Gets the Sparksee attribute identifier for the given type identifier and attribute name.
- new_node (self, type)

  Creates a new node instance.

- find_node_types (self)

  Gets all existing Sparksee node type identifiers.
- new_attribute (self, type, name, dt, kind)

  Creates a new attribute.
- new_session_array_attribute (self, type, dt, size)

  Creates a new Session array attribute.
- get_attributes (self, oid, with_values, with_names)

  Gets all Sparksee attribute identifiers with a non-NULL value for the given Sparksee OID.
- new_edge (self, type, tail_attr, tail_v, head_attr, head_v)

  Creates a new edge instance.
- get_attribute (self, attr)

  Gets information about the given attribute.
- get_session (self)

  Gets the Session this Graph belongs to.
- new_node_type (self, name)

  Creates a new node type.
- dump_storage (self, file)

  Dumps internal storage data to a file.
- select (self, type)

  Selects all OIDs belonging to the given type.
- select (self, attr, cond, value, restriction)

  Selects all OIDs satisfying the given condition for the given attribute.
- select (self, attr, cond, value)

  Selects all OIDs satisfying the given condition for the given attribute.
- rename_attribute (self, attr, new_name)

  Renames an attribute.
- count_edges (self)

  Gets the number of edges.
- rename_type (self, type, new_name)

  Renames a type.
- tails_and_heads (self, edges, tails, heads)

  Gets all the tails and heads from the given edges collection.
- new_session_attribute (self, type, dt, kind, default_value)

  Creates a new Session attribute with a default value.
- heads (self, edges)

  Gets all the heads from the given edges collection.
- new_attribute (self, type, name, dt, kind, default_value)

  Creates a new attribute with a default value.
- rename_type (self, old_name, new_name)

  Renames a type.
- explode (self, oid, etype, dir)

  Selects all edges from or to the given node OID and for the given edge type.
- get_attributes (self, oid)

  Gets all Sparksee attribute identifiers with a non-NULL value for the given Sparksee OID.
- find_attributes (self, type)

  Gets all existing Sparksee attribute identifiers for the given type identifier.
- get_array_attribute (self, oid, attr)

  Gets the ValueArray for the given array attribute and OID or NULL if it does not exist.
- get_attribute (self, oid, attr)

  Gets the Value for the given attribute and OID.
- find_or_create_object (self, attr, value)

Finds one object having the given Value for the attribute or it creates one does not exist any.

- get_edge_data (self, edge)

  Gets information about an edge.
- drop (self, objs)

  Drops all the OIDs from the given collection.
- new_edge (self, type, tail, head)

  Creates a new edge instance.
- get_type (self, type)

  Gets information about the given type.

5.29.1 Detailed Description

Graph class.

Each Database has a Graph associated, which is the persistent graph which contains all data stored into the graph database and is retrieved from a Session.

Check out the 'API' and the 'SPARKSEE graph database' sections in the SPARKSEE User Manual for more details on the use of this class.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.29.2 Member Function Documentation

backup()

```
sparksee.Graph.backup (
            self,
            file)
```

Dumps all the data to a backup file.

See the Sparksee class Restore methods. Errorfor other issues.

Parameters

file	[in] Output backup file path.

Exceptions

RuntimeError	null
IOError	If the given file cannot be created.

count_edges()

```
sparksee.Graph.count_edges (
              self)
```

Gets the number of edges.

Returns

The number of edges.

count_nodes()

```
sparksee.Graph.count_nodes (
              self)
```

Gets the number of nodes.

Returns

The number of nodes.

degree()

```
sparksee.Graph.degree (
              self,
              oid,
              etype,
              dir)
```

Gets the number of edges from or to the given node OID and for the given edge type.

Parameters

oid	[in] Sparksee node OID.
etype	[in] Sparksee edge type identifier.
dir	[in] Direction.

Returns

The number of edges.

drop() [1/2]

```
sparksee.Graph.drop (
              self,
              objs)
```

Drops all the OIDs from the given collection.

See Drop method with the single OID parameter. This performs the same operation for each object in the given set.

Parameters

objs	[in] Objects collection with the OIDs to be removed.

drop() [2/2]

```
sparksee.Graph.drop (
              self,
              oid)
```

Drops the given OID.

It also removes its egdges as well as its attribute values.

Parameters

oid	[in] Sparksee OID to be removed.

dump_data()

```
sparksee.Graph.dump_data (
              self,
              file)
```

Dumps logical data to a file.

Errorfor other issues.

Parameters

file	[in] Output file path.

Exceptions

RuntimeError	null
IOError	If the given file cannot be created.

dump_storage()

```
sparksee.Graph.dump_storage (
              self,
              file)
```

Dumps internal storage data to a file.

Errorfor other issues.

Parameters

file	[in] Output file path.

Exceptions

RuntimeError	null
IOError	If the given file cannot be created.

edges()

```
sparksee.Graph.edges (
              self,
              etype,
              tail,
              head)
```

Gets all the edges of the given type between two given nodes (tail and head).

Parameters

etype	[in] Sparksee edge type identifier.
tail	[in] Tail node identifier.
head	[in] Head node identifier.

Returns

Objects instance.

encrypted_backup()

```
sparksee.Graph.encrypted_backup (
              self,
              file,
              key_in_hex,
              iv_in_hex)
```

Dumps all the data to a backup file.

See the Sparksee class RestoreEncryptedBackup methods. Errorfor other issues.

Parameters

file	[in] Output backup file path.
key_in_hex	[In] The AES encryption Key as an hexadecimal string (8, 16 or 32 bytes).
iv_in_hex	[In] The AES Initialization Vector as an hexadecimal string (16 bytes).

Exceptions

RuntimeError	null
IOError	If the given file cannot be created.

explode() [1/2]

```
sparksee.Graph.explode (
            self,
            objs,
            etype,
            dir)
```

Selects all edges from or to each of the node OID in the given collection and for the given edge type.

Parameters

objs	[in] Sparksee node OID collection.
etype	[in] Sparksee edge type identifier.
dir	[in] Direction.

Returns

Objects instance.

explode() [2/2]

```
sparksee.Graph.explode (
            self,
            oid,
            etype,
            dir)
```

Selects all edges from or to the given node OID and for the given edge type.

Parameters

oid	[in] Sparksee node OID.

etype	[in] Sparksee edge type identifier.
dir	[in] Direction.

Returns

Objects instance.

export()

```
sparksee.Graph.export (
                self,
                file,
                type,
                em)
```

Exports the Graph.

Parameters

file	[in] Output file.
type	[in] Export type.
em	[in] Defines how to do the export for each graph object.

Exceptions

IOError	for errors writing to the file.

find_attribute()

```
sparksee.Graph.find_attribute (
                self,
                type,
                name)
```

Gets the Sparksee attribute identifier for the given type identifier and attribute name.

Parameters

type	[in] Sparksee type identifier.
name	[in] Unique attribute name.

Returns

The Sparksee attribute identifier for the given type and attribute name or InvalidAttribute if there is no attribute with the given name for the given type.

find_attributes()

```
sparksee.Graph.find_attributes (
              self,
              type)
```

Gets all existing Sparksee attribute identifiers for the given type identifier.

Parameters

type	[in] Sparksee type identifier.

Returns

Sparksee attribute identifier list.

find_edge()

```
sparksee.Graph.find_edge (
              self,
              etype,
              tail,
              head)
```

Gets any of the edges of the given type between two given nodes (tail and head).

If there are more than one, then any of them will be returned. And in case there are no edge between the given tail and head, the Objects InvalidOID will be returned.

Parameters

etype	[in] Sparksee edge type identifier.
tail	[in] Tail node identifier.
head	[in] Head node identifier.

Returns

Any of the edges or the Objects InvalidOID.

find_edge_types()

```
sparksee.Graph.find_edge_types (
              self)
```

Gets all existing Sparksee edge type identifiers.

Returns

Sparksee edge type identifier list.

find_node_types()

```
sparksee.Graph.find_node_types (
              self)
```

Gets all existing Sparksee node type identifiers.

Returns

Sparksee node type identifier list.

find_object()

```
sparksee.Graph.find_object (
              self,
              attr,
              value)
```

Finds one object having the given Value for the given attribute.

If there are more than one, then any of them will be returned. And in case there are no object having this Value, the Objects InvalidOID will be returned.

Parameters

attr	[in] Sparksee attribute identifier.
value	[in] Value.

Returns

Sparksee OID or the Objects InvalidOID.

find_or_create_edge()

```
sparksee.Graph.find_or_create_edge (
              self,
              etype,
              tail,
              head)
```

Gets any of the edges of the specified type between two given nodes (tail and head).

If it can not find any edge of this type between them it tries to create a new one.

Parameters

etype	[in] Sparksee edge type identifier.
tail	[in] Tail node identifier.

head	[in] Head node identifier.

Returns

Any of the edges or the Objects InvalidOID.

find_or_create_object()

```
sparksee.Graph.find_or_create_object (
              self,
              attr,
              value)
```

Finds one object having the given Value for the attribute or it creates one does not exist any.

If the attribute is a node or edge attribute and at least one node/edge with that value is found, it returns one of them. But if it does not exist, then: If it's a node attribute it will create it and set the attribute. If it's an edge attribute it will return the InvalidOID.

Using this method with a global attribute will return the InvalidOID.

Parameters

attr	[in] Sparksee attribute identifier.
value	[in] Value.

Returns

Sparksee OID or the Objects InvalidOID.

find_type()

```
sparksee.Graph.find_type (
              self,
              name)
```

Gets the Sparksee type identifier for the given type name.

Parameters

name	[in] Unique type name.

Returns

The Sparksee type identifier for the given type name or the Type InvalidType if there is no type with the given name.

find_types()

```
sparksee.Graph.find_types (
              self)
```

Gets all existing Sparksee node and edge type identifiers.

Returns

Sparksee node and edge type identifier list.

get_array_attribute()

```
sparksee.Graph.get_array_attribute (
              self,
              oid,
              attr)
```

Gets the ValueArray for the given array attribute and OID or NULL if it does not exist.

Parameters

oid	[in] Sparksee OID.
attr	[in] Sparksee array attribute identifier.

Returns

A ValueArray. This returns a ValueArray, a NULL or throws an exception.

get_attribute() [1/3]

```
sparksee.Graph.get_attribute (
              self,
              attr)
```

Gets information about the given attribute.

Parameters

attr	[in] Sparksee attribute identifier.

Returns

The Attribute for the given Sparksee attribute identifier.

get_attribute() [2/3]

```
sparksee.Graph.get_attribute (
              self,
              oid,
              attr)
```

Gets the Value for the given attribute and OID.

The other version of this call, where the Value is an output parameter instead of the return, is better because it allows the user to reuse an existing Value instance, whereas this call always creates a new Value instance to be returned.

It never returns NULL. Thus, in case the OID has a NULL value for the attribute it returns a NULL Value instance.

Parameters

oid	[in] Sparksee OID.
attr	[in] Sparksee attribute identifier.

Returns

A new Value instance having the attribute value for the given OID.

get_attribute() [3/3]

```
sparksee.Graph.get_attribute (
              self,
              oid,
              attr,
              value)
```

Gets the Value for the given attribute and OID.

Parameters

oid	[in] Sparksee OID.
attr	[in] Sparksee attribute identifier.
value	[in|out] Value for the given attribute and for the given OID.

get_attribute_interval_count()

```
sparksee.Graph.get_attribute_interval_count (
              self,
              attr,
              lower,
              include_lower,
              higher,
              include_higher)
```

Gets how many objects have a value into the given range for the given attribute.

This only works for the attributes with the AttributeKind Indexed or Unique.

Given values must belong to the same DataType than the attribute.

Parameters

attr	[in] Sparksee attribute identifier.
lower	[in] Lower bound Value of the range.
include_lower	[in] If TRUE, include lower bound Value of the range.
higher	[in] Higher bound Value of the range.
include_higher	[in] If TRUE, include higher bound Value of the range.

Returns

Number of objects having a value into the given range.

get_attribute_statistics()

```
sparksee.Graph.get_attribute_statistics (
              self,
              attr,
              basic)
```

Gets statistics from the given attribute.

The statistics can only be obtained from Indexed or Unique attributes.

Parameters

attr	[in] Sparksee attribute identifier.
basic	[in] If FALSE all statistics are computed, if TRUE just those statistics marked as basic will be computed (see description of the AttributeStatistics class). Of course, computing just basic statistics will be faster than computing all of them.

Returns

An AttributeStatistics instace.

get_attribute_text()

```
sparksee.Graph.get_attribute_text (
              self,
              oid,
              attr)
```

Gets the read-only TextStream for the given text attribute and OID.

Parameters

oid	[in] Sparksee OID.
attr	[in] Sparksee attribute identifier.

Returns

A TextStream. This returns a TextStream to read.

get_attributes() [1/2]

```
sparksee.Graph.get_attributes (
              self,
              oid)
```

Gets all Sparksee attribute identifiers with a non-NULL value for the given Sparksee OID.

Parameters

oid	[in] Sparksee OID.

Returns

Sparksee attribute identifier list.

get_attributes() [2/2]

```
sparksee.Graph.get_attributes (
              self,
              oid,
              with_values,
              with_names)
```

Gets all Sparksee attribute identifiers with a non-NULL value for the given Sparksee OID.

Additional options can be set to also get it's Values and the names of the attributes. The Text and Array attributes will have the Value set to Null because they can't be represented as a simple Value object.

Parameters

oid	[in] Sparksee OID.
with_values	[in] True to also get the Values of each attribute.
with_names	[in] True to also get the Names of each attribute.

Returns

Sparksee attribute identifier list.

get_edge_data()

```
sparksee.Graph.get_edge_data (
              self,
              edge)
```

Gets information about an edge.

Parameters

edge	[in] Sparksee edge identifier.

Returns

An EdgeData instance.

get_edge_peer()

```
sparksee.Graph.get_edge_peer (
              self,
              edge,
              node)
```

Gets the other end for the given edge.

Parameters

edge	[in] Sparksee edge identifier.
node	[in] Sparksee node identifier. It must be one of the ends of the edge.

Returns

The other end of the edge.

get_object_type()

```
sparksee.Graph.get_object_type (
              self,
              oid)
```

Gets the Sparksee node or edge type identifier for the given OID.

Parameters

oid	[in] Sparksee OID.

Returns

Sparksee node or edge type identifier.

get_session()

```
sparksee.Graph.get_session (
              self)
```

Gets the Session this Graph belongs to.

Returns

The owning Session instance.

get_type()

```
sparksee.Graph.get_type (
              self,
              type)
```

Gets information about the given type.

Parameters

type	[in] Sparksee type identifier.

Returns

The Type for the given Sparksee type identifier.

get_values()

```
sparksee.Graph.get_values (
              self,
              attr)
```

Gets the Value collection for the given attribute.

Parameters

attr	[in] Sparksee attribute identifier.

Returns

Returns a Values object.

heads()

```
sparksee.Graph.heads (
                self,
                edges)
```

Gets all the heads from the given edges collection.

Parameters

edges	[in] Sparksee edge identifier collection.

Returns

The heads collection.

index_attribute()

```
sparksee.Graph.index_attribute (
                self,
                attr,
                kind)
```

Updates the index of the given attribute.

This just works if the current index of the attribute corresponds to the AttributeKind Basic and the new one is Indexed or Unique.

Parameters

attr	[in] Sparksee attribute identifier.
kind	[in] Attribute kind.

index_neighbors()

```
sparksee.Graph.index_neighbors (
                self,
                edge_type,
                neighbors)
```

Creates or destroys the neighbors index of an edge type.

Parameters

edge_type	[in] Sparksee Edge type identifier.
neighbors	[in] If TRUE, it indexes the neighbor nodes, otherwise it removes the index.

neighbors() [1/2]

```
sparksee.Graph.neighbors (
                self,
                objs,
                etype,
                dir)
```

Selects all neighbor nodes from or to each of the node OID in the given collection and for the given edge type.

Parameters

objs	[in] Sparksee node OID collection.
etype	[in] Sparksee edge type identifier.
dir	[in] Direction.

Returns

Objects instance.

neighbors() [2/2]

```
sparksee.Graph.neighbors (
                self,
                oid,
                etype,
                dir)
```

Selects all neighbor nodes from or to the given node OID and for the given edge type.

Parameters

oid	[in] Sparksee node OID.
etype	[in] Sparksee edge type identifier.
dir	[in] Direction.

Returns

Objects instance.

new_array_attribute()

```
sparksee.Graph.new_array_attribute (
                self,
                type,
                name,
                dt,
                size)
```

Creates a new array attribute.

Parameters

type	[in] Sparksee node or edge type identifier.
name	[in] Unique name for the new attribute.
dt	[in] Base Data type for the new array attribute elements.
size	[in] The array size.

Returns

Unique Sparksee attribute identifier.

new_attribute() [1/2]

```
sparksee.Graph.new_attribute (
              self,
              type,
              name,
              dt,
              kind)
```

Creates a new attribute.

Parameters

type	[in] Sparksee node or edge type identifier.
name	[in] Unique name for the new attribute.
dt	[in] Data type for the new attribute.
kind	[in] Attribute kind.

Returns

Unique Sparksee attribute identifier.

new_attribute() [2/2]

```
sparksee.Graph.new_attribute (
              self,
              type,
              name,
              dt,
              kind,
              default_value)
```

Creates a new attribute with a default value.

Parameters

type	[in] Sparksee node or edge type identifier.
name	[in] Unique name for the new attribute.
dt	[in] Data type for the new attribute.
kind	[in] Attribute kind.
default_value	[in] The default value to use in each new node/edge.

Returns

Unique Sparksee attribute identifier.

new_edge() [1/2]

```
sparksee.Graph.new_edge (
              self,
              type,
              tail,
              head)
```

Creates a new edge instance.

Parameters

type	[in] Sparksee type identifier.
tail	[in] Source Sparksee OID.
head	[in] Target Sparksee OID.

Returns

Unique OID of the new edge instance.

new_edge() [2/2]

```
sparksee.Graph.new_edge (
              self,
              type,
              tail_attr,
              tail_v,
              head_attr,
              head_v)
```

Creates a new edge instance.

The tail of the edge will be any node having the given tailV Value for the given tailAttr attribute identifier, and the head of the edge will be any node having the given headV Value for the given headAttr attribute identifier.

Parameters

type	[in] Sparksee type identifier.
tail_attr	[in] Sparksee attribute identifier.
tail_v	[in] Value.
head_attr	[in] Sparksee attribute identifier.
head_v	[in] Value.

Returns

Unique OID of the new edge instance.

new_edge_type()

```
sparksee.Graph.new_edge_type (
              self,
              name,
              directed,
              neighbors)
```

Creates a new edge type.

Parameters

name	[in] Unique name for the new edge type.
directed	[in] If TRUE, this creates a directed edge type, otherwise this creates a undirected edge type.
neighbors	[in] If TRUE, this indexes neighbor nodes, otherwise not.

Returns

Unique Sparksee type identifier.

new_node()

```
sparksee.Graph.new_node (
              self,
              type)
```

Creates a new node instance.

Parameters

type	[in] Sparksee type identifier.

Returns

Unique OID of the new node instance.

new_node_type()

```
sparksee.Graph.new_node_type (
              self,
              name)
```

Creates a new node type.

Parameters

name	[in] Unique name for the new node type.

Returns

Unique Sparksee type identifier.

new_restricted_edge_type()

```
sparksee.Graph.new_restricted_edge_type (
              self,
              name,
              tail,
              head,
              neighbors)
```

Creates a new restricted edge type.

Parameters

name	[in] Unique name for the new edge type.
tail	[in] Tail Sparksee node type identifier.
head	[in] Head Sparksee node type identifier.
neighbors	[in] If TRUE, this indexes neighbor nodes, otherwise not.

Returns

Unique Sparksee type identifier.

new_session_array_attribute()

```
sparksee.Graph.new_session_array_attribute (
              self,
              type,
              dt,
              size)
```

Creates a new Session array attribute.

Session attributes are exclusive for the Session (just its Session can use the attribute) and are automatically removed when the Session is closed (thus, attribute data is not persistent into the database).

Since they are not persistent, they cannot be retrieved from the database, so they do not have an identifier name.

Parameters

type	[in] Sparksee node or edge type identifier.
dt	[in] Base Data type for the new array attribute elements.
size	[in] The array size.

Returns

Unique Sparksee attribute identifier.

new_session_attribute() [1/2]

```
sparksee.Graph.new_session_attribute (
              self,
              type,
              dt,
              kind)
```

Creates a new Session attribute.

Session attributes are exclusive for the Session (just its Session can use the attribute) and are automatically removed when the Session is closed (thus, attribute data is not persistent into the database).

Since they are not persistent, they cannot be retrieved from the database, so they do not have an identifier name.

Parameters

type	[in] Sparksee node or edge type identifier.
dt	[in] Data type for the new attribute.
kind	[in] Attribute kind.

Returns

Unique Sparksee attribute identifier.

new_session_attribute() [2/2]

```
sparksee.Graph.new_session_attribute (
              self,
              type,
              dt,
              kind,
              default_value)
```

Creates a new Session attribute with a default value.

Session attributes are exclusive for the Session (just its Session can use the attribute) and are automatically removed when the Session is closed (thus, attribute data is not persistent into the database).

Since they are not persistent, they cannot be retrieved from the database, so they do not have an identifier name.

Parameters

type	[in] Sparksee node or edge type identifier.
dt	[in] Data type for the new attribute.
kind	[in] Attribute kind.
default_value	[in] The default value to use in each new node/edge.

Returns

Unique Sparksee attribute identifier.

remove_attribute()

```
sparksee.Graph.remove_attribute (
              self,
              attr)
```

Removes the given attribute.

Parameters

attr	[in] Sparksee attribute identifier.

remove_type()

```
sparksee.Graph.remove_type (
              self,
              type)
```

Removes the given type.

This fails if there exist attributes defined for the type or if there exist restricted edges referencing this type.

Parameters

type	[in] Sparksee type identifier.

rename_attribute()

```
sparksee.Graph.rename_attribute (
              self,
              attr,
              new_name)
```

Renames an attribute.

The new name must be available.

Parameters

attr	[in] Sparksee attribute identifier.
new_name	[in] The new name for the attribute.

rename_type() [1/2]

```
sparksee.Graph.rename_type (
              self,
              old_name,
              new_name)
```

Renames a type.

The new name must be available.

Parameters

old_name	[in] The current name of the type to be renamed.
new_name	[in] The new name for the type.

rename_type() [2/2]

```
sparksee.Graph.rename_type (
              self,
              type,
              new_name)
```

Renames a type.

The new name must be available.

Parameters

type	[in] The type to be renamed.
new_name	[in] The new name for the type.

select() [1/5]

```
sparksee.Graph.select (
                self,
                attr,
                cond,
                lower,
                higher)
```

Selects all OIDs satisfying the given condition for the given attribute.

This allows to perform the Between operation, thus it has two Value arguments.

Parameters

attr	[in] Sparksee attribute identifier.
cond	[in] Condition to be satisfied. It must be the Between Condition.
lower	[in] Lower-bound Value to be satisfied.
higher	[in] Higher-bound Value to be satisfied.

Returns

Objects instance.

select() [2/5]

```
sparksee.Graph.select (
                self,
                attr,
                cond,
                lower,
                higher,
                restriction)
```

Selects all OIDs satisfying the given condition for the given attribute.

This allows to perform the Between operation, thus it has two Value arguments.

Parameters

attr	[in] Sparksee attribute identifier.
cond	[in] Condition to be satisfied. It must be the Between Condition.
lower	[in] Lower-bound Value to be satisfied.
higher	[in] Higher-bound Value to be satisfied.
restriction	[in] Objects to limit the select in this set of objects.

Returns

Objects instance.

select() [3/5]

```
sparksee.Graph.select (
                self,
                attr,
                cond,
                value)
```

Selects all OIDs satisfying the given condition for the given attribute.

Parameters

attr	[in] Sparksee attribute identifier.
cond	[in] Condition to be satisfied.
value	[in] Value to be satisfied.

Returns

Objects instance.

select() [4/5]

```
sparksee.Graph.select (
                self,
                attr,
                cond,
                value,
                restriction)
```

Selects all OIDs satisfying the given condition for the given attribute.

Parameters

attr	[in] Sparksee attribute identifier.
cond	[in] Condition to be satisfied.
value	[in] Value to be satisfied.
restriction	[in] Objects to limit the select in this set of objects.

Returns

Objects instance.

select() [5/5]

```
sparksee.Graph.select (
              self,
              type)
```

Selects all OIDs belonging to the given type.

Parameters

type	[in] Sparksee type identifier.

Returns

Objects instance.

set_array_attribute() [1/2]

```
sparksee.Graph.set_array_attribute (
              self,
              attr,
              value)
```

Sets all the values of the array of the given array attribute for all the objects of the types the attribute applies to.

Parameters

attr	[in] Sparksee array attribute identifier.
value	[in] Value for all the array elements of the arrays.

set_array_attribute() [2/2]

```
sparksee.Graph.set_array_attribute (
              self,
              oid,
              attr,
              value)
```

Sets all the elements of the ValueArray for the given array attribute and OID and returns it.

Initializing the array with one of these SetArrayAttribute methods is the only way to create the array. But once created it can be updated using the ValueArray. And you can get it again with the GetArrayAttribute operation.

Parameters

oid	[in] Sparksee OID.
attr	[in] Sparksee array attribute identifier.
value	[in] Value for all the array elements of the given attribute and OID.

Returns

A ValueArray. This returns a ValueArray, a NULL or throws an exception.

set_array_attribute_void()

```
sparksee.Graph.set_array_attribute_void (
                self,
                oid,
                attr,
                value)
```

Sets all the elements of the ValueArray for the given array attribute and OID.

Initializing the array with one of these SetArrayAttribute methods is the only way to create the array. But once created it can be updated using the ValueArray which can be obtained using the GetArrayAttribute operation.

Parameters

oid	[in] Sparksee OID.
attr	[in] Sparksee array attribute identifier.
value	[in] Value for all the array elements of the given attribute and OID.

set_attribute() [1/2]

```
sparksee.Graph.set_attribute (
                self,
                attr,
                value)
```

Sets the value of the given attribute for all the objects of the types the attribute applies to.

Does not work for TEXT attribute types

txThe Transaction this operation belongs to attributeThe attribute to initialize

Parameters

attr	null
value	The value to initialize the attribute to

set_attribute() [2/2]

```
sparksee.Graph.set_attribute (
                self,
                oid,
                attr,
                value)
```

Sets the Value for the given attribute and OID.

Parameters

oid	[in] Sparksee OID.
attr	[in] Sparksee attribute identifier.
value	[in] Value for the given attribute and for the given OID.

set_attribute_default_value()

```
sparksee.Graph.set_attribute_default_value (
                self,
                attr,
                value)
```

Sets a default value for an attribute.

The default value will be applied to all the new nodes or edges.

The given value must have the same DataType as the attribute or be a NULL value to remove the current default value.

Parameters

attr	[in] The attribute.
value	[in] The default value to use for this attribute.

set_attribute_text()

```
sparksee.Graph.set_attribute_text (
                self,
                oid,
                attr,
                tstream)
```

Sets the writable TextStream for the given text attribute and OID.

Parameters

oid	[in] Sparksee OID.
attr	[in] Sparksee attribute identifier.
tstream	[in] New Text value. This corresponds to a TextStream to write.

tails()

```
sparksee.Graph.tails (
                self,
                edges)
```

Gets all the tails from the given edges collection.

Parameters

edges	[in] Sparksee edge identifier collection.

Returns

The tails collection.

tails_and_heads()

```
sparksee.Graph.tails_and_heads (
                self,
                edges,
                tails,
                heads)
```

Gets all the tails and heads from the given edges collection.

Parameters

edges	[in] Sparksee edge identifier collection.
tails	[in|out] If not NULL, all the tails will be stored here.
heads	[in|out] If not NULL, all the heads will be stored here.

top_k() [1/6]

```
sparksee.Graph.top_k (
                self,
                attribute,
                operation,
                lower,
                higher,
                order,
                k)
```

Gets a KeyValues iterator as a result of the TopK operation.

Parameters

attribute	The attribute to perform the TopK on

operation	The operation to perform in the top k
lower	The lower bound Value to be satisfied
higher	The upper bound Value to be satisfied
order	The ordering semantincs of the top-k
k	The size of the TopK

Returns

A KeyValues instance

top_k() [2/6]

```
sparksee.Graph.top_k (
                self,
                attribute,
                operation,
                lower,
                higher,
                order,
                k,
                restriction)
```

Gets a KeyValues iterator as a result of the TopK operation.

restrictonObjects to limit the topk to this set of objects

Parameters

attribute	The attribute to perform the TopK on
operation	The operation to perform in the top k
lower	The lower bound Value to be satisfied
higher	The upper bound Value to be satisfied
order	The ordering semantincs of the top-k
k	The size of the TopK
restriction	null

Returns

A KeyValues instance

top_k() [3/6]

```
sparksee.Graph.top_k (
              self,
              attribute,
              operation,
              lower,
              order,
              k)
```

Gets a KeyValues iterator as a result of the TopK operation.

Parameters

attribute	The attribute to perform the TopK on
operation	The operation to perform in the top k
lower	The lower bound Value to be satisfied
order	The ordering semantincs of the top-k
k	The size of the TopK

Returns

A KeyValues instance

top_k() [4/6]

```
sparksee.Graph.top_k (
              self,
              attribute,
              operation,
              lower,
              order,
              k,
              restriction)
```

Gets a KeyValues iterator as a result of the TopK operation.

restrictonObjects to limit the topk to this set of objects

Parameters

attribute	The attribute to perform the TopK on
operation	The operation to perform in the top k
lower	The lower bound Value to be satisfied
order	The ordering semantincs of the top-k
k	The size of the TopK
restriction	null

Returns

A KeyValues instance

top_k() [5/6]

```
sparksee.Graph.top_k (
                self,
                attribute,
                order,
                k)
```

Gets a KeyValues iterator as a result of the TopK operation.

Parameters

attribute	The attribute to perform the TopK on
order	The ordering semantincs of the top-k
k	The size of the TopK

Returns

A KeyValues instance

top_k() [6/6]

```
sparksee.Graph.top_k (
                self,
                attribute,
                order,
                k,
                restriction)
```

Gets a KeyValues iterator as a result of the TopK operation.

restrictonObjects to limit the topk to this set of objects

Parameters

attribute	The attribute to perform the TopK on
order	The ordering semantincs of the top-k
k	The size of the TopK
restriction	null

Returns

A KeyValues instance

5.30 sparksee.GraphExport Class Reference

Stores the graph exporting values.

Public Member Functions

- **set_defaults** (self)

  Sets to default values.
- set_label (self, label)

  Sets the graph label.
- get_label (self)

  Gets the graph label.
- set_color_rgb (self, color)

  Sets the background color.
- get_color_rgb (self)

  Gets the background color.
- **__init__** (self)

  Creates a new GraphExport instance.

5.30.1 Detailed Description

Stores the graph exporting values.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.30.2 Member Function Documentation

get_color_rgb()

```
sparksee.GraphExport.get_color_rgb (
              self)
```

Gets the background color.

Returns

The background color.

get_label()

```
sparksee.GraphExport.get_label (
              self)
```

Gets the graph label.

Returns

The graph label.

set_color_rgb()

```
sparksee.GraphExport.set_color_rgb (
              self,
              color)
```

Sets the background color.

Parameters

color	The background color.

set_label()

```
sparksee.GraphExport.set_label (
              self,
              label)
```

Sets the graph label.

Parameters

label	[in] The graph label.

5.31 sparksee.Int32List Class Reference

Sparksee 32-bit signed integer list.

Public Member Functions

- **clear** (self)
  Clears the list.
- __init__ (self)
  Constructor.
- __iter__ (self)
  Gets a new TypeListIterator.
- iterator (self)
  Gets a new Int32ListIterator.
- add (self, value)
  Adds an 32-bit signed integer at the end of the list.
- count (self)
  Number of elements in the list.

5.31.1 Detailed Description

Sparksee 32-bit signed integer list.

It stores a 32-bit signed integer list.

Use Int32ListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.31.2 Constructor & Destructor Documentation

__init__()

```
sparksee.Int32List.__init__ (
              self)
```

Constructor.

This creates an empty list.

5.31.3 Member Function Documentation

__iter__()

```
sparksee.Int32List.__iter__ (
              self)
```

Gets a new TypeListIterator.

Returns

TypeListIterator instance

add()

```
sparksee.Int32List.add (
              self,
              value)
```

Adds an 32-bit signed integer at the end of the list.

Parameters

value	[in] The integer.

count()

```
sparksee.Int32List.count (
              self)
```

Number of elements in the list.

Returns

Number of elements in the list.

iterator()

```
sparksee.Int32List.iterator (
              self)
```

Gets a new Int32ListIterator.

Returns

Int32ListIterator instance.

5.32 sparksee.Int32ListIterator Class Reference

Int32List iterator class.

Public Member Functions

- next (self)
  Moves to the next element.
- has_next (self)
  Gets if there are more elements.
- __next__ (self)
  Return the next item from the container.

5.32.1 Detailed Description

Int32List iterator class.

Iterator to traverse all the integer into a Int32List instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.32.2 Member Function Documentation

__next__()

```
sparksee.Int32ListIterator.__next__ (
              self)
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

has_next()

```
sparksee.Int32ListIterator.has_next (
              self)
```

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

next()

```
sparksee.Int32ListIterator.next (
              self)
```

Moves to the next element.

Returns

The next element.

5.33 sparksee.KeyValue Class Reference

Public Member Functions

- get_key (self)
  Gets the key (OID) of this KeyValue pair.
- get_value (self)
  Gets the value of this KeyValue pair.
- **to_string** (self)
  *Returns a String representation of the KeyValue, used in **unicode** and **str**.*

5.33.1 Member Function Documentation

get_key()

```
sparksee.KeyValue.get_key (
              self)
```

Gets the key (OID) of this KeyValue pair.

Returns

The Sparksee object identifier for this entry.

get_value()

```
sparksee.KeyValue.get_value (
              self)
```

Gets the value of this KeyValue pair.

Returns

The Value associated with this entry.

5.34 sparksee.KeyValues Class Reference

Value set class.

Public Member Functions

- next (self)
  Gets the next KeyValue pair.
- next (self, kv)
  Gets the next KeyValue pair.
- has_next (self)
  Checks if the KeyValues has more KeyValue pairs.
- close (self)
  Closes the KeyValues instance.
- __next__ (self)
  Return the next item from the container.
- is_closed (self)
  Gets if KeyValues instance has been closed or not.

5.34.1 Detailed Description

Value set class.

This is a set of Value instances, that is there is no duplicated elements.

Use a ValuesIterator to traverse all the elements into the set.

When the Values instance is closed, it closes all existing and non-closed ValuesIterator instances too.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.34.2 Member Function Documentation

__next__()

```
sparksee.KeyValues.__next__ (
              self)
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

close()

```
sparksee.KeyValues.close (
              self)
```

Closes the KeyValues instance.

It must be called to ensure the integrity of all data.

has_next()

```
sparksee.KeyValues.has_next (
              self)
```

Checks if the KeyValues has more KeyValue pairs.

Returns

Returns ture if there are more KeyValue pairs

is_closed()

```
sparksee.KeyValues.is_closed (
              self)
```

Gets if KeyValues instance has been closed or not.

See also

close()

Returns

TRUE if the KeyValues instance has been closed, FALSE otherwise.

next() [1/2]

```
sparksee.KeyValues.next (
              self)
```

Gets the next KeyValue pair.

Returns

Returns the next KeyValue pair

next() [2/2]

```
sparksee.KeyValues.next (
              self,
              kv)
```

Gets the next KeyValue pair.

Parameters

kv	Returns the next KeyValue pair

5.35 sparksee.KOpt Class Reference

KOpt class.

5.35.1 Detailed Description

KOpt class.

Implements the 2-Opt and 3-Opt algorithms

Author

Sparsity Technologies http://www.sparsity-technologies.com

ValueList iterator class.

Iterator to traverse all the values into a ValueList instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.36 sparksee.LogLevel Class Reference

Log level enumeration.

Static Public Attributes

- int **OFF** = 0

  Disable log.
- int SEVERE = 1

  Severe log level.
- int WARNING = 2

  Warning log level.
- int INFO = 3

  Info log level.
- int CONFIG = 4

  Config log level.
- int FINE = 5

  Fine log level.
- int DEBUG = 6

  Debug log level.

5.36.1 Detailed Description

Log level enumeration.

Log level priority order is as follows, from minimum to maximum log information: Off (log is disabled), Severe, Warning, Info, Config, Fine, Debug.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.36.2 Member Data Documentation

CONFIG

```
int sparksee.LogLevel.CONFIG = 4  [static]
```

Config log level.

Errors, warnings, information messages and configuration details of the different components are shown.

DEBUG

```
int sparksee.LogLevel.DEBUG = 6  [static]
```

Debug log level.

This is for Sparksee development purposes and just works with debug versions of the library.

FINE

```
int sparksee.LogLevel.FINE = 5  [static]
```

Fine log level.

This is the higher and finest public log level, everything is dumped to the log.

INFO

```
int sparksee.LogLevel.INFO = 3  [static]
```

Info log level.

Errors, warnings and information messages are shown.

SEVERE

```
int sparksee.LogLevel.SEVERE = 1  [static]
```

Severe log level.

This is the lower log level, just errors are shown.

WARNING

```
int sparksee.LogLevel.WARNING = 2  [static]
```

Warning log level.

Errors and warnings are shown.

5.37 sparksee.MissingEndpoint Class Reference

The policy to follow whenever and edge endpoint is missing during a loading.

Static Public Attributes

- int **IS_ERROR** = 0
  Throw an error.
- int **CREATE** = 1
  Create the endpoint.
- int **IGNORE** = 2
  Ignore the edge.

5.37.1 Detailed Description

The policy to follow whenever and edge endpoint is missing during a loading.

5.38 sparksee.NodeExport Class Reference

Stores the node exporting values.

Public Member Functions

- get_image (self)
  Gets the node image.
- get_labelcolor_rgb (self)
  Gets the node label color.
- set_label (self, label)
  Sets the node label.
- set_width (self, width)
  Gets the node width.
- set_image (self, image)
  Sets the node image.
- set_color_rgb (self, color)
  Sets the node color.
- set_font_size (self, size)
  Sets the node label font size.
- set_height (self, height)
  Sets the node height.
- get_shape (self)
  Gets the node shape.
- set_labelcolor_rgb (self, color)
  Sets the node label color.
- is_fit (self)
  Gets whether the node size is fitted to the label or not.
- get_color_rgb (self)
  Gets the node color.
- set_fit (self, fit)
  Sets the node fit property.
- **__init__** (self)
  Creates a new instance.
- get_font_size (self)
  Gets the node label font size.
- **set_defaults** (self)
  Sets to default values.
- get_label (self)
  Gets the node label.
- get_height (self)
  Gets the node height.
- get_title (self)
  Gets the node title.
- get_width (self)

Gets the node width.

- set_title (self, title)

  Sets the node title.

- copy_defaults (self, source)

  Sets the default values using a refrerence NodeExport.

- set_shape (self, shape)

  Sets the node shape.

5.38.1 Detailed Description

Stores the node exporting values.

When 'fit' is set to TRUE, then 'height' and 'width' will be ignored.

Some properties may be ignored depending on the exportation type.

Default values are:

Label: "" (empty string).

Shape: Box.

Color: 10863606 (0xa5c3f6).

Label color: 0 (Ox000000, Black).

Height: 25px.

Width: 25px.

Fit: TRUE.

Font size: 10.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.38.2 Member Function Documentation

copy_defaults()

```
sparksee.NodeExport.copy_defaults (
              self,
              source)
```

Sets the default values using a refrerence NodeExport.

Parameters

source	null

get_color_rgb()

```
sparksee.NodeExport.get_color_rgb (
              self)
```

Gets the node color.

Returns

The node color.

get_font_size()

```
sparksee.NodeExport.get_font_size (
              self)
```

Gets the node label font size.

Returns

The node label font size.

get_height()

```
sparksee.NodeExport.get_height (
              self)
```

Gets the node height.

Returns

The node height in pixels.

get_image()

```
sparksee.NodeExport.get_image (
              self)
```

Gets the node image.

Returns

The node image.

get_label()

```
sparksee.NodeExport.get_label (
              self)
```

Gets the node label.

Returns

The node label.

get_labelcolor_rgb()

```
sparksee.NodeExport.get_labelcolor_rgb (
              self)
```

Gets the node label color.

Returns

The node label color.

get_shape()

```
sparksee.NodeExport.get_shape (
              self)
```

Gets the node shape.

Returns

The node shape.

get_title()

```
sparksee.NodeExport.get_title (
              self)
```

Gets the node title.

Returns

The node title.

get_width()

```
sparksee.NodeExport.get_width (
              self)
```

Gets the node width.

Returns

The node width in pixels.

is_fit()

```
sparksee.NodeExport.is_fit (
              self)
```

Gets whether the node size is fitted to the label or not.

Returns

If TRUE, then the node size is fitted to the label, otherwise the size is fixed with the values of 'height' and 'width'.

set_color_rgb()

```
sparksee.NodeExport.set_color_rgb (
              self,
              color)
```

Sets the node color.

Parameters

color	The node color.

set_fit()

```
sparksee.NodeExport.set_fit (
              self,
              fit)
```

Sets the node fit property.

Parameters

fit	[in] If TRUE, then the node size is fitted to the label ('height' and 'width' will be ignored), otherwise the size is fixed with the values of 'height' and 'width'.

set_font_size()

```
sparksee.NodeExport.set_font_size (
              self,
              size)
```

Sets the node label font size.

Parameters

size	[in] The node label font size.

set_height()

```
sparksee.NodeExport.set_height (
              self,
              height)
```

Sets the node height.

Parameters

height	[in] The node height in pixels.

set_image()

```
sparksee.NodeExport.set_image (
              self,
              image)
```

Sets the node image.

Parameters

image	[in] The node image. The image should be a path to a local file or an URL. The image usage in the export depends on the exported format and shape. For Graphviz: Graphviz expects only local image paths as images in dot files unless you plan on using something like it's "dot_url_resolve.py" script to automatically download them. The Image shape is not directly supported for Graphviz. For Html: The vis.js html exports only show the image if the node shape selected is Image or Circle (that will be translated as "circularimage").

set_label()

```
sparksee.NodeExport.set_label (
              self,
              label)
```

Sets the node label.

Parameters

label	[in] The node label.

set_labelcolor_rgb()

```
sparksee.NodeExport.set_labelcolor_rgb (
              self,
              color)
```

Sets the node label color.

Parameters

color	[in] The node label color.

set_shape()

```
sparksee.NodeExport.set_shape (
              self,
              shape)
```

Sets the node shape.

Parameters

shape	[in] The node shape.

set_title()

```
sparksee.NodeExport.set_title (
              self,
              title)
```

Sets the node title.

Parameters

title	[in] The node title.

set_width()

```
sparksee.NodeExport.set_width (
              self,
              width)
```

Gets the node width.

Parameters

width	The node width in pixels.

5.39 sparksee.NodeShape Class Reference

Node shape.

Static Public Attributes

- int **BOX** = 0
  Box Shape.
- int **ROUND** = 1
  Round Shape.
- int **ELLIPSE** = 2
  Ellipse Shape.
- int **CIRCLE** = 3
  Circle Shape.
- int **CYLINDER** = 4
  Cylinder Shape.
- int **TEXT_SHAPE** = 5
  Plain Text Shape.
- int **DIAMOND** = 6
  Diamond Shape.
- int **DOT** = 7
  Dot Shape.
- int **STAR** = 8
  Star Shape.
- int **TRIANGLE** = 9
  Upwards Triangle Shape.
- int **TRIANGLE_DOWN** = 10
  Downwards Triangle Shape.
- int **SQUARE** = 11
  Square Shape.
- int **IMAGE** = 12
  Image Shape.

5.39.1 Detailed Description

Node shape.

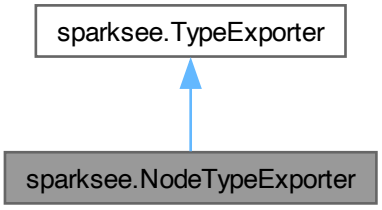
Author

Sparsity Technologies http://www.sparsity-technologies.com

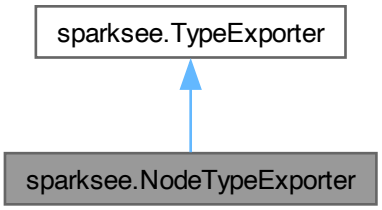
5.40 sparksee.NodeTypeExporter Class Reference

NodeTypeExporter class.

Inheritance diagram for sparksee.NodeTypeExporter:



Collaboration diagram for sparksee.NodeTypeExporter:



Public Member Functions

- **__init__** (self)

  Creates a new instance.
- run (self)

  See the TypeExporter class Run method.
- __init__ (self, row_writer, graph, type, attrs)

Creates a new instance.

- set_type (self, type)

  Sets the type to be exported.
- set_header (self, header)

  Sets the presence of a header row.
- set_row_writer (self, rw)

  Sets the output data destination.
- set_graph (self, graph)

  Sets the graph that will be exported.
- register (self, tel)

  Registers a new listener.
- set_attributes (self, attrs)

  Sets the list of Attributes.
- set_frequency (self, freq)

  Sets the frequency of listener notification.

5.40.1 Detailed Description

NodeTypeExporter class.

Specific TypeExporter implementation for node types.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.40.2 Constructor & Destructor Documentation

__init__()

```
sparksee.NodeTypeExporter.__init__ (
              self,
              row_writer,
              graph,
              type,
              attrs)
```

Creates a new instance.

Parameters

row_writer	[in] Output RowWriter.
graph	[in] Graph.
type	[in] Type identifier.
attrs	[in] Attribute identifiers to be exported.

5.40.3 Member Function Documentation

register()

```
sparksee.NodeTypeExporter.register (
              self,
              tel)
```

Registers a new listener.

Parameters

tel	[in] TypeExporterListener to be registered.

Reimplemented from sparksee.TypeExporter.

run()

```
sparksee.NodeTypeExporter.run (
              self)
```

See the TypeExporter class Run method.

WrongArgumentErrorIn case of bad settings ErrorFor other issues

Exceptions

RuntimeError	null
ValueError	null
IOError	If bad things happen writting to the RowWriter.

Reimplemented from sparksee.TypeExporter.

set_attributes()

```
sparksee.NodeTypeExporter.set_attributes (
              self,
              attrs)
```

Sets the list of Attributes.

Parameters

attrs	[in] Attribute identifiers to be exported

Reimplemented from sparksee.TypeExporter.

set_frequency()

```
sparksee.NodeTypeExporter.set_frequency (
              self,
              freq)
```

Sets the frequency of listener notification.

Parameters

freq	[in] Frequency in number of rows managed to notify progress to all listeners

Reimplemented from sparksee.TypeExporter.

set_graph()

```
sparksee.NodeTypeExporter.set_graph (
              self,
              graph)
```

Sets the graph that will be exported.

Parameters

graph	[in] Graph.

Reimplemented from sparksee.TypeExporter.

set_header()

```
sparksee.NodeTypeExporter.set_header (
              self,
              header)
```

Sets the presence of a header row.

Parameters

header	[in] If TRUE, a header row is dumped with the name of the attributes.

Reimplemented from sparksee.TypeExporter.

set_row_writer()

```
sparksee.NodeTypeExporter.set_row_writer (
              self,
              rw)
```

Sets the output data destination.

Parameters

rw	[in] Input RowWriter.

Reimplemented from sparksee.TypeExporter.

set_type()

```
sparksee.NodeTypeExporter.set_type (
              self,
              type)
```

Sets the type to be exported.

Parameters

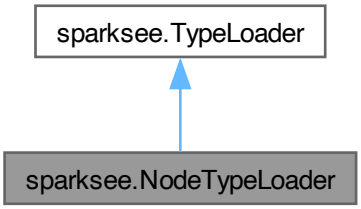
type	[in] Type identifier.

Reimplemented from sparksee.TypeExporter.

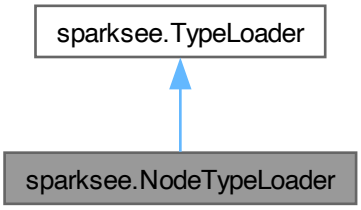
5.41 sparksee.NodeTypeLoader Class Reference

NodeTypeLoader class.

Inheritance diagram for sparksee.NodeTypeLoader:



Collaboration diagram for sparksee.NodeTypeLoader:



Public Member Functions

- run (self)

  See the TypeLoader class Run method.
- set_log_error (self, path)

  Sets a log error file.
- set_type (self, type)

  Sets the type to be loaded.
- set_locale (self, locale_str)

  Sets the locale that will be used to read the data.
- set_timestamp_format (self, timestamp_format)

  Sets a specific timestamp format.
- set_attributes (self, attrs)

  Sets the list of Attributes.
- **__init__** (self)

  Creates a new instance.
- register (self, tel)

  Registers a new listener.
- set_frequency (self, freq)

  Sets the frequency of listener notification.
- __init__ (self, row_reader, graph, type, attrs, attrs_pos)

  Creates a new instance.
- run_two_phases (self)

  See the TypeLoader class RunTwoPhases method.
- set_row_reader (self, rr)

  Sets the input data source.
- set_attribute_positions (self, attrs_pos)

  Sets the list of attribute positions.
- set_graph (self, graph)

  Sets the graph where the data will be loaded.
- run_n_phases (self, partitions)

  See the TypeLoader class RunNPhases method.
- set_log_off (self)

  Truns off all the error reporting.

5.41.1 Detailed Description

NodeTypeLoader class.

Specific TypeLoader implementation for node types.

Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.41.2 Constructor & Destructor Documentation

__init__()

```
sparksee.NodeTypeLoader.__init__ (
                self,
                row_reader,
                graph,
                type,
                attrs,
                attrs_pos)
```

Creates a new instance.

Parameters

row_reader	[in] Input RowReader.
graph	[in] Graph.
type	[in] Type identifier.
attrs	[in] Attribute identifiers to be loaded.
attrs_pos	[in] Attribute positions (column index >=0).

5.41.3 Member Function Documentation

register()

```
sparksee.NodeTypeLoader.register (
                self,
                tel)
```

Registers a new listener.

Parameters

tel	TypeLoaderListener to be registered.

Reimplemented from sparksee.TypeLoader.

run()

```
sparksee.NodeTypeLoader.run (
              self)
```

See the TypeLoader class Run method.

ErrorFor other issues

Exceptions

RuntimeError	null
IOError	If bad things happen using to the RowReader.

Reimplemented from sparksee.TypeLoader.

run_n_phases()

```
sparksee.NodeTypeLoader.run_n_phases (
              self,
              partitions)
```

See the TypeLoader class RunNPhases method.

ErrorFor other issues

Parameters

partitions	null

Exceptions

RuntimeError	null
IOError	If bad things happen using to the RowReader.

Reimplemented from sparksee.TypeLoader.

run_two_phases()

```
sparksee.NodeTypeLoader.run_two_phases (
              self)
```

See the TypeLoader class RunTwoPhases method.

ErrorFor other issues

Exceptions

RuntimeError	null
IOError	If bad things happen using to the RowReader.

Reimplemented from sparksee.TypeLoader.

set_attribute_positions()

```
sparksee.NodeTypeLoader.set_attribute_positions (
              self,
              attrs_pos)
```

Sets the list of attribute positions.

Parameters

attrs_pos	[in] Attribute positions (column index >=0).

Reimplemented from sparksee.TypeLoader.

set_attributes()

```
sparksee.NodeTypeLoader.set_attributes (
              self,
              attrs)
```

Sets the list of Attributes.

Parameters

attrs	[in] Attribute identifiers to be loaded

Reimplemented from sparksee.TypeLoader.

set_frequency()

```
sparksee.NodeTypeLoader.set_frequency (
              self,
              freq)
```

Sets the frequency of listener notification.

Parameters

freq	[in] Frequency in number of rows managed to notify progress to all listeners

Reimplemented from sparksee.TypeLoader.

set_graph()

```
sparksee.NodeTypeLoader.set_graph (
              self,
              graph)
```

Sets the graph where the data will be loaded.

Parameters

graph	[in] Graph.

Reimplemented from sparksee.TypeLoader.

set_locale()

```
sparksee.NodeTypeLoader.set_locale (
              self,
              locale_str)
```

Sets the locale that will be used to read the data.

It should match the locale used in the rowreader.

Parameters

locale_str	[in] The locale string for the read data. See CSVReader.

Reimplemented from sparksee.TypeLoader.

set_log_error()

```
sparksee.NodeTypeLoader.set_log_error (
              self,
              path)
```

Sets a log error file.

By default errors are thrown as a exception and the load process ends. If a log file is set, errors are logged there and the load process does not stop.

Parameters

path	[in] The path to the error log file.

Exceptions

IOError	If bad things happen opening the file.

Reimplemented from sparksee.TypeLoader.

set_log_off()

```
sparksee.NodeTypeLoader.set_log_off (
              self)
```

Truns off all the error reporting.

The log file will not be created and no exceptions for invalid data will be thrown. If you just want to turn off the logs, but abort at the first error what you should do is not call this method and not set a logError file.

Reimplemented from sparksee.TypeLoader.

set_row_reader()

```
sparksee.NodeTypeLoader.set_row_reader (
              self,
              rr)
```

Sets the input data source.

Parameters

rr	[in] Input RowReader.

Reimplemented from sparksee.TypeLoader.

set_timestamp_format()

```
sparksee.NodeTypeLoader.set_timestamp_format (
              self,
              timestamp_format)
```

Sets a specific timestamp format.

Parameters

timestamp_format	[in] A string with the timestamp format definition.

Reimplemented from sparksee.TypeLoader.

set_type()

```
sparksee.NodeTypeLoader.set_type (
              self,
              type)
```

Sets the type to be loaded.

Parameters

type	[in] Type identifier.

Reimplemented from sparksee.TypeLoader.

5.42 sparksee.Objects Class Reference

Object identifier set class.

Public Member Functions

- copy (self, objs)
  Performs the copy operation.
- count (self)
  Gets the number of elements into the collection.
- equals (self, objs)
  Checks if the given Objects contains the same information.
- any (self)
  Gets an element from the collection.
- contains (self, objs)
  Check if this objects contains the other one.
- combine_intersection (self, objs1, objs2)
  Creates a new Objects instance which is the intersection of the two given.
- __iter__ (self)
  Gets a new ObjectsIterator.
- iterator (self)
  Gets an ObjectsIterator.
- union (self, objs)
  Performs the union operation.
- combine_union (self, objs1, objs2)
  Creates a new Objects instance which is the union of the two given.
- intersection (self, objs)
  Performs the intersection operation.
- **clear** (self)
  Clears the collection removing all its elements.
- iterator_from_index (self, index)
  Gets an ObjectsIterator skipping index elements.
- exists (self, e)
  Gets if the given element exists into the collection.
- copy (self)
  Creates a new Objects instance as a copy of the given one.

- difference (self, objs)

  Performs the difference operation.
- combine_difference (self, objs1, objs2)

  Creates a new Objects instance which is the difference of the two given.
- sample (self, exclude, samples)

  Creates a new Objects instance which is a sample of the calling one.
- close (self)

  Closes the Objects instance.
- add (self, e)

  Adds an element into the collection.
- remove (self, e)

  Removes an element from the collection.
- iterator_from_element (self, e)

  Gets an ObjectsIterator starting from the given element.
- is_closed (self)

  Gets if Objects instance has been closed or not.

Static Public Attributes

- int **INVALID_OID** = 0

  Invalid OID constant.

5.42.1 Detailed Description

Object identifier set class.

It stores a collection of Sparksee object identifiers as a set. As a set, there is no order and no duplicated elements.

This class should be used just to store large collections. Otherwise, it is strongly recommended to use common classes from the language API.

This class is not thread-safe.

ObjectsIterator must be used to traverse all the elements into the set.

When the Objects instance is closed, it closes all existing and non-closed ObjectsIterator instances too.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.42.2 Member Function Documentation

__iter__()

```
sparksee.Objects.__iter__ (
                self)
```

Gets a new ObjectsIterator.

Returns

ObjectsIterator instance

add()

```
sparksee.Objects.add (
              self,
              e)
```

Adds an element into the collection.

Parameters

e	[in] Element to be added.

Returns

TRUE if the element is added, FALSE if the element was already into the collection.

any()

```
sparksee.Objects.any (
              self)
```

Gets an element from the collection.

Errorfor other issues.

Returns

Any element from the collection.

Exceptions

RuntimeError	null
RuntimeError	whether the collection is empty.

close()

```
sparksee.Objects.close (
              self)
```

Closes the Objects instance.

It must be called to ensure the integrity of all data.

combine_difference()

```
sparksee.Objects.combine_difference (
              self,
              objs1,
              objs2)
```

Creates a new Objects instance which is the difference of the two given.

Two given Objects belong to the same Session.

Parameters

objs1	[in] Objects instance.
objs2	[in] Objects instance.

Returns

New Objects instance.

combine_intersection()

```
sparksee.Objects.combine_intersection (
              self,
              objs1,
              objs2)
```

Creates a new Objects instance which is the intersection of the two given.

Two given Objects belong to the same Session.

Parameters

objs1	[in] Objects instance.
objs2	[in] Objects instance.

Returns

New Objects instance.

combine_union()

```
sparksee.Objects.combine_union (
              self,
              objs1,
              objs2)
```

Creates a new Objects instance which is the union of the two given.

Two given Objects belong to the same Session.

Parameters

objs1	[in] Objects instance.
objs2	[in] Objects instance.

Returns

New Objects instance.

contains()

```
sparksee.Objects.contains (
              self,
              objs)
```

Check if this objects contains the other one.

Parameters

objs	Objects collection.

Returns

True if it contains the given object.

copy() [1/2]

```
sparksee.Objects.copy (
              self)
```

Creates a new Objects instance as a copy of the given one.

Returns

The new Objects instance.

copy() [2/2]

```
sparksee.Objects.copy (
              self,
              objs)
```

Performs the copy operation.

This updates the Objects calling instance and copies the given Objects instance.

Parameters

objs	[in] Objects instance.

Returns

Number of elements into the collection once the operation has been executed.

count()

```
sparksee.Objects.count (
              self)
```

Gets the number of elements into the collection.

Returns

The number of elements into the collection.

difference()

```
sparksee.Objects.difference (
              self,
              objs)
```

Performs the difference operation.

This updates the Objects calling instance removing those existing elements at the given Objects instance.

Parameters

objs	[in] Objects instance.

Returns

Number of elements into the collection once the operation has been executed.

equals()

```
sparksee.Objects.equals (
              self,
              objs)
```

Checks if the given Objects contains the same information.

Parameters

objs	[in] Objects instance.

Returns

True if the objects are equal or false otherwise.

exists()

```
sparksee.Objects.exists (
              self,
              e)
```

Gets if the given element exists into the collection.

Parameters

e	[in] Element.

Returns

TRUE if the element exists into the collection, FALSE otherwise.

intersection()

```
sparksee.Objects.intersection (
              self,
              objs)
```

Performs the intersection operation.

Updates the Objects calling instance setting those existing elements at both two collections and removing all others.

Parameters

objs	[in] Objects instance.

Returns

Number of elements into the collection once the operation has been executed.

is_closed()

```
sparksee.Objects.is_closed (
              self)
```

Gets if Objects instance has been closed or not.

See also

close()

Returns

TRUE if the Objects instance has been closed, FALSE otherwise.

iterator()

```
sparksee.Objects.iterator (
              self)
```

Gets an ObjectsIterator.

Returns

ObjectsIterator instance.

iterator_from_element()

```
sparksee.Objects.iterator_from_element (
              self,
              e)
```

Gets an ObjectsIterator starting from the given element.

Objects collection has no order, so this method is implementation-dependent. e[in] The first element to traverse in the resulting

Parameters

e	[in] The first element to traverse in the resulting ObjectsIterator instance.

Returns

ObjectsIterator instance.

iterator_from_index()

```
sparksee.Objects.iterator_from_index (
              self,
              index)
```

Gets an ObjectsIterator skipping index elements.

Objects collection has no order, so this method is implementation-dependent.

Parameters

index	[in] The number of elements to skip from the beginning. It must be in the range [0..Size).

Returns

ObjectsIterator instance.

remove()

```
sparksee.Objects.remove (
              self,
              e)
```

Removes an element from the collection.

Parameters

e	[in] Element to be removed.

Returns

TRUE if the element is removed, FALSE if the element was not into the collection.

sample()

```
sparksee.Objects.sample (
              self,
              exclude,
              samples)
```

Creates a new Objects instance which is a sample of the calling one.

Parameters

exclude	[in] If not NULL, elements into this collection will be excluded from the resulting one.
samples	[in] Number of elements into the resulting collection.

Returns

Sample collection.

union()

```
sparksee.Objects.union (
              self,
              objs)
```

Performs the union operation.

This adds all existing elements of the given Objects instance to the Objects calling instance

Parameters

objs	[in] Objects instance.

Returns

Number of elements into the collection once the operation has been executed.

5.43 sparksee.ObjectsIterator Class Reference

ObjectsIterator class.

Public Member Functions

- next (self)
  Gets the next element to traverse.
- has_next (self)
  Gets if there are more elements to traverse.
- close (self)
  Closes the ObjectsIterator instance.
- __next__ (self)
  Return the next item from the container.
- is_closed (self)
  Gets if ObjectsIterator instance has been closed or not.

5.43.1 Detailed Description

ObjectsIterator class.

Iterator to traverse all the object identifiers from an Objects instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.43.2 Member Function Documentation

__next__()

```
sparksee.ObjectsIterator.__next__ (
              self)
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

close()

```
sparksee.ObjectsIterator.close (
              self)
```

Closes the ObjectsIterator instance.

It must be called to ensure the integrity of all data.

has_next()

```
sparksee.ObjectsIterator.has_next (
              self)
```

Gets if there are more elements to traverse.

Returns

TRUE if there are more elements to traverse, FALSE otherwise.

is_closed()

```
sparksee.ObjectsIterator.is_closed (
              self)
```

Gets if ObjectsIterator instance has been closed or not.

See also

close()

Returns

TRUE if the ObjectsIterator instance has been closed, FALSE otherwise.

next()

```
sparksee.ObjectsIterator.next (
              self)
```

Gets the next element to traverse.

Returns

The next element.

5.44 sparksee.ObjectType Class Reference

Object type enumeration.

Static Public Attributes

- int **NODE** = 0
  Node object type.
- int **EDGE** = 1
  Edge object type.

5.44.1 Detailed Description

Object type enumeration.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.45 sparksee.OIDList Class Reference

Sparksee object identifier list.

Public Member Functions

- **clear** (self)
  Clears the list.
- __init__ (self, num_invalid_oids)
  Constructor.
- __init__ (self)
  Constructor.
- __iter__ (self)
  Gets a new TypeListIterator.
- set (self, pos, oid)
  Sets a Sparksee object identifier at the specified position of the list.
- iterator (self)
  Gets a new OIDListIterator.
- add (self, attr)
  Adds a Sparksee object identifier at the end of the list.
- count (self)
  Number of elements in the list.

5.45.1 Detailed Description

Sparksee object identifier list.

It stores a Sparksee object identifier list.

Use OIDListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.45.2 Constructor & Destructor Documentation

__init__() [1/2]

```
sparksee.OIDList.__init__ (
              self,
              num_invalid_oids)
```

Constructor.

This creates a list with N invalid oids.

Parameters

num_invalid_oids	[in] The number of invalid oids added to the list.

__init__() [2/2]

```
sparksee.OIDList.__init__ (
              self)
```

Constructor.

This creates an empty list.

5.45.3 Member Function Documentation

__iter__()

```
sparksee.OIDList.__iter__ (
              self)
```

Gets a new TypeListIterator.

Returns

TypeListIterator instance

add()

```
sparksee.OIDList.add (
              self,
              attr)
```

Adds a Sparksee object identifier at the end of the list.

Parameters

attr	[in] Sparksee object identifier.

count()

```
sparksee.OIDList.count (
              self)
```

Number of elements in the list.

Returns

Number of elements in the list.

iterator()

```
sparksee.OIDList.iterator (
              self)
```

Gets a new OIDListIterator.

Returns

OIDListIterator instance.

set()

```
sparksee.OIDList.set (
              self,
              pos,
              oid)
```

Sets a Sparksee object identifier at the specified position of the list.

Parameters

pos	[in] List position [0..Count()-1].
oid	[in] Sparksee object identifier.

5.46 sparksee.OIDListIterator Class Reference

OIDList iterator class.

Public Member Functions

- next (self)
  Moves to the next element.
- has_next (self)
  Gets if there are more elements.
- __next__ (self)
  Return the next item from the container.

5.46.1 Detailed Description

OIDList iterator class.

Iterator to traverse all the Sparksee object identifier into a OIDList instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.46.2 Member Function Documentation

__next__()

```
sparksee.OIDListIterator.__next__ (
              self)
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

has_next()

```
sparksee.OIDListIterator.has_next (
              self)
```

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

next()

```
sparksee.OIDListIterator.next (
              self)
```

Moves to the next element.

Returns

The next element.

5.47 sparksee.Order Class Reference

Order enumeration.

Static Public Attributes

- int **ASCENDENT** = 0
  From lower to higher.
- int **DESCENDENT** = 1
  From higher to lower.

5.47.1 Detailed Description

Order enumeration.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.48 sparksee.PageRank Class Reference

PageRank class.

Public Member Functions

- set_damping (self, damping)
  Sets the damping value for the PageRank.
- set_default_weight (self, weight)
  Sets the default weight for those cases when a given edge does not have a weight attribute set.
- run (self)
  Runs the algorithm.
- add_all_edge_types (self, dir)
  Allows for traversing all edge types of the graph.
- set_output_attribute_type (self, attr)
  Sets the output attribute type.
- set_starting_node (self, start_node)
  Sets the starting node of the page rank to compute the Personalized PageRank variant.

- set_initial_page_rank_value (self, start_value)

  Sets the initial PageRank value.
- set_edge_weight_attribute_type (self, attr)

  Sets the attribute to use as edge weight.
- add_edge_type (self, type, dir)

  Allows for traversing edges of the given type.
- set_num_iterations (self, num_iterations)

  Sets the number of iterations to run the PageRank for.
- set_tolerance (self, tolerance)

  Sets the tolerance threashold to continue computing the PageRank after each iteration.
- add_node_type (self, type)

  Allows for traversing nodes of the given type.
- __init__ (self, session)

  Builds the PageRank.
- **add_all_node_types** (self)

  Allows for traversing all node types of the graph.

5.48.1 Detailed Description

PageRank class.

Implements the PageRank algorithm

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.48.2 Constructor & Destructor Documentation

__init__()

```
sparksee.PageRank.__init__ (
              self,
              session)
```

Builds the PageRank.

Parameters

session	[in] The session to use

5.48.3 Member Function Documentation

add_all_edge_types()

```
sparksee.PageRank.add_all_edge_types (
              self,
              dir)
```

Allows for traversing all edge types of the graph.

The direction is interpreted as in which direction an edge can be followed from a node to influence other nodes.

Parameters

dir	[in] Edge direction.

add_edge_type()

```
sparksee.PageRank.add_edge_type (
              self,
              type,
              dir)
```

Allows for traversing edges of the given type.

If the edge type was already added, the existing direction is overwritten The direction is interpreted as in which direction an edge can be followed from a node to influence other nodes.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Exceptions

RuntimeError	null

add_node_type()

```
sparksee.PageRank.add_node_type (
              self,
              type)
```

Allows for traversing nodes of the given type.

Parameters

type	null

Exceptions

RuntimeError	null

run()

```
sparksee.PageRank.run (
              self)
```

Runs the algorithm.

sparksee::gdb::Error

Exceptions

RuntimeError	null

set_damping()

```
sparksee.PageRank.set_damping (
              self,
              damping)
```

Sets the damping value for the PageRank.

Parameters

damping	[in] The damping value. Default: 0.85

set_default_weight()

```
sparksee.PageRank.set_default_weight (
              self,
              weight)
```

Sets the default weight for those cases when a given edge does not have a weight attribute set.

Default: 0.0

Parameters

weight	[in] The default weight

set_edge_weight_attribute_type()

```
sparksee.PageRank.set_edge_weight_attribute_type (
              self,
              attr)
```

Sets the attribute to use as edge weight.

If the multiple edge are set for traversal, this attribute must be of type GLOBAL_TYPE or EDGES_TYPE. Additionally, the attribute must be of type Double. Finally, negative weights are treated as non existing, so the default weight applies.

sparksee::gdb::Error

Parameters

attr	[in] The attribute type to use as a weight. Default: InvalidAttribute

Exceptions

RuntimeError	null

set_initial_page_rank_value()

```
sparksee.PageRank.set_initial_page_rank_value (
              self,
              start_value)
```

Sets the initial PageRank value.

If a starting node is set, this initial value is only set for the starting node and the rest of nodes are set to 0.0

Parameters

start_value	[in] The initial value to set. Default: 0.0

set_num_iterations()

```
sparksee.PageRank.set_num_iterations (
              self,
              num_iterations)
```

Sets the number of iterations to run the PageRank for.

Parameters

num_iterations	[in] The number of interations to set. Default: 20

set_output_attribute_type()

```
sparksee.PageRank.set_output_attribute_type (
              self,
              attr)
```

Sets the output attribute type.

If the PageRank will run on more than one node type, then the output attribute must be of type GLOBAL_TYPE or NODES_TYPE. Otherwise, it must be a valid attribute for the used node type.

Parameters

attr	[in] The attribute to store the result. Default: InvalidAttribute

Exceptions

RuntimeError	null

set_starting_node()

```
sparksee.PageRank.set_starting_node (
              self,
              start_node)
```

Sets the starting node of the page rank to compute the Personalized PageRank variant.

sparksee::gdb::Error

Parameters

start_node	null

Exceptions

RuntimeError	null

set_tolerance()

```
sparksee.PageRank.set_tolerance (
              self,
              tolerance)
```

Sets the tolerance threashold to continue computing the PageRank after each iteration.

If all the changes to any PPR value after an iteration are below that tolerance threshold, the algorithm finishes.

Parameters

tolerance	[in] The tolerance to use normalized between 0 and 1. Default: 0.000001

5.49 sparksee.Platform Class Reference

Platform class.

Public Member Functions

- get_statistics (self, stats)
  Gets platform data and statistics.

5.49.1 Detailed Description

Platform class.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.49.2 Member Function Documentation

get_statistics()

```
sparksee.Platform.get_statistics (
              self,
              stats)
```

Gets platform data and statistics.

Parameters

stats	[in|out] This updates the given PlatformStatistics.

5.50 sparksee.PlatformStatistics Class Reference

Platform data and statistics.

Public Member Functions

- **__init__** (self)

  Creates a new instance setting all values to 0.
- get_total_mem (self)

  Gets physical memory size in Bytes.
- get_available_mem (self)

  Gets avialable (free) memory size in Bytes.
- get_real_time (self)

  Gets time in microseconds (since epoch).
- get_system_time (self)

  Gets CPU system time.
- get_user_time (self)

  Gets CPU user time.
- get_num_c_p_us (self)

  Gets the number of CPUs.

5.50.1 Detailed Description

Platform data and statistics.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.50.2 Member Function Documentation

get_available_mem()

```
sparksee.PlatformStatistics.get_available_mem (
              self)
```

Gets avialable (free) memory size in Bytes.

Returns

Avialable (free) memory size in Bytes.

get_num_c_p_us()

```
sparksee.PlatformStatistics.get_num_c_p_us (
              self)
```

Gets the number of CPUs.

Returns

The number of CPUs.

get_real_time()

```
sparksee.PlatformStatistics.get_real_time (
              self)
```

Gets time in microseconds (since epoch).

Returns

Time in microseconds (since epoch).

get_system_time()

```
sparksee.PlatformStatistics.get_system_time (
              self)
```

Gets CPU system time.

Returns

CPU system time.

get_total_mem()

```
sparksee.PlatformStatistics.get_total_mem (
              self)
```

Gets physical memory size in Bytes.

Returns

Physical memory size in Bytes.

get_user_time()

```
sparksee.PlatformStatistics.get_user_time (
              self)
```

Gets CPU user time.

Returns

CPU user time.

5.51 sparksee.Query Class Reference

Query class.

Public Member Functions

- set_stream (self, stream, handler)

  Sets a query stream handler Query streams handlers are created and destroyed by the caller.
- set_dynamic (self, name, value)

  Sets the value for a dynamic paramater.
- execute (self, stmt, reiterable)

  Executes the given statement.
- close (self)

  Closes the Query instance.
- is_closed (self)

  Gets if the Query instance has been closed or not.

5.51.1 Detailed Description

Query class.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.51.2 Member Function Documentation

close()

```
sparksee.Query.close (
              self)
```

Closes the Query instance.

It must be called to ensure the integrity of all data.

execute()

```
sparksee.Query.execute (
              self,
              stmt,
              reiterable)
```

Executes the given statement.

Parameters

stmt	[in] Query statement.
reiterable	[in] Whether we want the resultset to be reiterable or not

Returns

A ResultSet instance with the contents of the result of the query.

is_closed()

```
sparksee.Query.is_closed (
              self)
```

Gets if the Query instance has been closed or not.

See also

close()

Returns

TRUE if the Query instance has been closed, FALSE otherwise.

set_dynamic()

```
sparksee.Query.set_dynamic (
              self,
              name,
              value)
```

Sets the value for a dynamic paramater.

Parameters

name	[in] Parameter name
value	[in] Parameter value

set_stream()

```
sparksee.Query.set_stream (
              self,
              stream,
              handler)
```

Sets a query stream handler Query streams handlers are created and destroyed by the caller.

DeprecatedQueryStreams should be handled by a QueryStreamManager setup at database level instead.

Parameters

stream	[in] The stream name
handler	[in] Query stream handler

Returns

The previous handler, or NULL if it does not exists

5.52 sparksee.QueryContext Class Reference

Query context interface.

Public Member Functions

- **new_query** (self)

  Creates a new Query.
- **__init__** (self)

  Default constructor.

5.52.1 Detailed Description

Query context interface.

A QueryContext contains and manages the resources required to run a Query. A Session is one example of a QueryContext connected to a Sparksee database. The applications can implement their own contexts to run queries out of Sparksee.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.53 sparksee.QueryLanguage Class Reference

The supported query languages.

Static Public Attributes

- int **SPARKSEE_ALGEBRA** = 0

  The internal Sparksee Algebra.
- int **SPARKSEE_CYPHER** = 1

  The Sparksee Cypher Language inspired by the OpenCypher Query Language.

5.53.1 Detailed Description

The supported query languages.

5.54 sparksee.QueryStream Class Reference

Query stream interface.

Public Member Functions

- fetch (self, list)

  Gets the next row and moves the iterator forward.
- prepare (self, list)

  Prepares the stream before it is started.
- start (self, list)

  Starts the stream.

5.54.1 Detailed Description

Query stream interface.

A QueryStream is the interface between the application and the STREAM operator. When the operator starts inside a Query, the method is prepared with query-defined arguments. Then, if there are input operations, the STREAM operator builds the ResultSets and starts the iteration. Finally, the operator fetches rows until no more are available.

Application exceptions must be cached by the subclass that implements the interface.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.54.2 Member Function Documentation

fetch()

```
sparksee.QueryStream.fetch (
              self,
              list)
```

Gets the next row and moves the iterator forward.

The end of sequence is denoted by returning TRUE with an empty row. A valid row must contain as many values (even NULL) as expected by the query.

Parameters

list	[out] Storage for the new rows

Returns

TRUE if there is a row or end of sequence, FALSE on error

prepare()

```
sparksee.QueryStream.prepare (
              self,
              list)
```

Prepares the stream before it is started.

Parameters

list	[in] Optional list of arguments

Returns

FALSE on error

start()

```
sparksee.QueryStream.start (
              self,
              list)
```

Starts the stream.

Parameters

list	[in] Optional list of input ResultSets

Returns

FALSE on error

5.55 sparksee.QueryStreamManager Class Reference

QueryStreamManager interface.

Public Member Functions

- release_query_stream (self, qstream)

  Release a QueryStream.
- get_query_stream (self, name, session)

  Returns an instance of a QueryStream with the given name or NULL.

5.55.1 Detailed Description

QueryStreamManager interface.

A QueryStreamManager is the interface to handle objects that will implement user defined query operations through the STREAM operator.

When a query using a custom named STREAM operator requires it it will allocate a QueryStream object using the provided QueryStreamManagers. And once it's finished it will be released through the same QueryStreamManager that was used to allocate it.

Application exceptions must be cached by the subclass that implements the interface.

Maintaining and freeing the QueryStreamManager itself is responsibility of the caller.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.55.2 Member Function Documentation

get_query_stream()

```
sparksee.QueryStreamManager.get_query_stream (
              self,
              name,
              session)
```

Returns an instance of a QueryStream with the given name or NULL.

If this manager handles the given name it will return an instance of a QueryStream that will provide the requested QueryStream operation.

The ReleaseQueryStream will be called once it's usage is finished. It's up to the QueryStreamManager implementation to really allocate/deallocate a new QueryStream each time or share a static one.

Parameters

name	[In] The name of the QueryStream that we want to get
session	[In] A pointer to the Session used in the Query and may be required by the Stream.

Returns

Returns a QueryStream or NULL

release_query_stream()

```
sparksee.QueryStreamManager.release_query_stream (
              self,
              qstream)
```

Release a QueryStream.

The QueryStream must have been reserved by the same QueryStreamManager

The Manager must free the QueryStream if it was allocated or handle it as it requires for it's implementation.

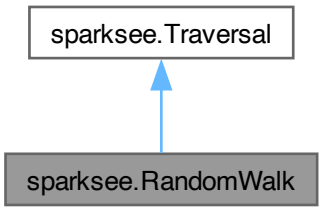
Parameters

qstream	[In] The QueryStream to be released.

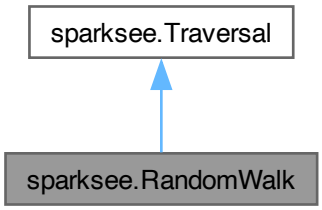
5.56 sparksee.RandomWalk Class Reference

RandomWalk class.

Inheritance diagram for sparksee.RandomWalk:



Collaboration diagram for sparksee.RandomWalk:



Public Member Functions

- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- set_default_weight (self, weight)

  Sets the default weight for those cases when a given edge does not have a weight attribute set.
- add_all_edge_types (self, dir)

  Allows for traversing all edge types of the graph.
- exclude_edges (self, edges)

  Set which edges can't be used.
- set_seed (self, seed)

  Sets the seed of the random walk.
- set_edge_weight_attribute_type (self, attr)

  Sets the attribute to use as edge weight.
- set_in_out_parameter (self, val)

  Sets the In-Out parameter of the RandomWalk.
- add_edge_type (self, type, dir)

  Allows for traversing edges of the given type.
- __init__ (self, session, node)

  Builds the RandomWalk.
- add_node_type (self, type)

  Allows for traversing nodes of the given type.
- set_return_parameter (self, val)

  Sets the return parameter of the RandomWalk.
- next (self)

  Gets the next object of the traversal.
- has_next (self)

  Gets if there are more objects to be traversed.
- get_current_depth (self)

  Returns the depth of the current node.
- reset (self, start_node)

  Sets the starting node of the RandomWalk.
- **add_all_node_types** (self)

  Allows for traversing all node types of the graph.
- set_maximum_hops (self, maxhops)

  Sets the maximum hops restriction.
- __iter__ (self)

  Gets a new TraversalIterator.
- __next__ (self)

  Return the next item from the container.
- close (self)

  Closes the Traversal instance.
- is_closed (self)

  Gets if Traversal has been closed or not.

5.56.1 Detailed Description

RandomWalk class.

Implements the RandomWalk algorithm

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.56.2 Constructor & Destructor Documentation

__init__()

```
sparksee.RandomWalk.__init__ (
              self,
              session,
              node)
```

Builds the RandomWalk.

Parameters

session	[in] The session to use
node	[in] The starting node of the traversal

5.56.3 Member Function Documentation

__iter__()

```
sparksee.Traversal.__iter__ (
              self)  [inherited]
```

Gets a new TraversalIterator.

Returns

TraversalIterator instance

__next__()

```
sparksee.Traversal.__next__ (
              self)  [inherited]
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

add_all_edge_types()

```
sparksee.RandomWalk.add_all_edge_types (
              self,
              dir)
```

Allows for traversing all edge types of the graph.

Parameters

dir	[in] Edge direction.

Reimplemented from sparksee.Traversal.

add_edge_type()

```
sparksee.RandomWalk.add_edge_type (
              self,
              type,
              dir)
```

Allows for traversing edges of the given type.

If the edge type was already added, the existing direction is overwritten

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Reimplemented from sparksee.Traversal.

add_node_type()

```
sparksee.RandomWalk.add_node_type (
              self,
              type)
```

Allows for traversing nodes of the given type.

Parameters

type	The node type to add

Reimplemented from sparksee.Traversal.

close()

```
sparksee.Traversal.close (
              self)  [inherited]
```

Closes the Traversal instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.RandomWalk.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.Traversal.

exclude_nodes()

```
sparksee.RandomWalk.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.Traversal.

get_current_depth()

```
sparksee.RandomWalk.get_current_depth (
              self)
```

Returns the depth of the current node.

That is, it returns the depth of the node returned in the last call to Next().

Returns

The depth of the current node.

Reimplemented from sparksee.Traversal.

has_next()

```
sparksee.RandomWalk.has_next (
              self)
```

Gets if there are more objects to be traversed.

Returns

TRUE if there are more objects, FALSE otherwise.

Reimplemented from sparksee.Traversal.

is_closed()

```
sparksee.Traversal.is_closed (
              self)  [inherited]
```

Gets if Traversal has been closed or not.

See also

close()

Returns

TRUE if the Traversal instance has been closed, FALSE otherwise.

next()

```
sparksee.RandomWalk.next (
              self)
```

Gets the next object of the traversal.

Returns

A node or edge identifier.

Reimplemented from sparksee.Traversal.

reset()

```
sparksee.RandomWalk.reset (
              self,
              start_node)
```

Sets the starting node of the RandomWalk.

This method resets the RandomWalk.

sparksee::gdb::Error

Parameters

start_node	null

set_default_weight()

```
sparksee.RandomWalk.set_default_weight (
              self,
              weight)
```

Sets the default weight for those cases when a given edge does not have a weight attribute set.

Default: 0.0

Parameters

weight	[in] The default weight

set_edge_weight_attribute_type()

```
sparksee.RandomWalk.set_edge_weight_attribute_type (
              self,
              attr)
```

Sets the attribute to use as edge weight.

If the multiple edge are set for traversal, this attribute must be of type GLOBAL_TYPE or EDGES_TYPE. Additionally, the attribute must be of type Double. Finally, negative weights are treated as non existing, so the default weight applies.

Parameters

attr	[in] The attribute type to use as a weight. Default: InvalidAttribute

set_in_out_parameter()

```
sparksee.RandomWalk.set_in_out_parameter (
              self,
              val)
```

Sets the In-Out parameter of the RandomWalk.

Parameters

val	The In-Out parameter to set. Default: 1.0

set_maximum_hops()

```
sparksee.RandomWalk.set_maximum_hops (
              self,
              maxhops)
```

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

Parameters

maxhops	[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.

Reimplemented from sparksee.Traversal.

set_return_parameter()

```
sparksee.RandomWalk.set_return_parameter (
              self,
              val)
```

Sets the return parameter of the RandomWalk.

Parameters

val	The return parameter to set. Default: 1.0

set_seed()

```
sparksee.RandomWalk.set_seed (
              self,
              seed)
```

Sets the seed of the random walk.

Parameters

seed	The seed to generate the random numbers that drive the random walk

5.57 sparksee.ResultSet Class Reference

ResultSet class.

Public Member Functions

- **rewind** (self)
  Positions the cursor before the first row.
- get_column_data_type (self, index)
  Gets the datatype for the given column.
- get_column (self, index)
  Gets the value for the given column.
- get_num_columns (self)
  Gets the number of columns.
- get_column_name (self, index)
  Gets the name for the given column.
- next (self)
  Fetches the next row.
- get_j_s_o_n (self, rows)
  Returns rows in JSON format.
- get_column (self, index, value)
  Gets the value for the given column.
- get_column_index (self, name)
  Gets the column index for the given column name.
- is_closed (self)
  Gets if the ResultSet instance has been closed or not.

5.57.1 Detailed Description

ResultSet class.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.57.2 Member Function Documentation

get_column() [1/2]

```
sparksee.ResultSet.get_column (
                self,
                index)
```

Gets the value for the given column.

QueryExceptionIf a database access error occurs.

Parameters

index	[in] Column index.

Returns

The Value of the given column.

get_column() [2/2]

```
sparksee.ResultSet.get_column (
                self,
                index,
                value)
```

Gets the value for the given column.

QueryExceptionIf a database access error occurs.

Parameters

index	[in] Column index.
value	[in|out] Value.

get_column_data_type()

```
sparksee.ResultSet.get_column_data_type (
                self,
                index)
```

Gets the datatype for the given column.

Parameters

index	[in] Column index.

Returns

DataType for the given column.

get_column_index()

```
sparksee.ResultSet.get_column_index (
              self,
              name)
```

Gets the column index for the given column name.

Parameters

name	[in] Column name.

Returns

Column index.

get_column_name()

```
sparksee.ResultSet.get_column_name (
              self,
              index)
```

Gets the name for the given column.

Parameters

index	[in] Column index.

Returns

Column name.

get_j_s_o_n()

```
sparksee.ResultSet.get_j_s_o_n (
              self,
              rows)
```

Returns rows in JSON format.

Rows are returned from the current position.

Parameters

rows	[in] Maximum number of rows

Returns

JSON representation of the next <rows> rows in the resultset

get_num_columns()

```
sparksee.ResultSet.get_num_columns (
              self)
```

Gets the number of columns.

Columns are in the range [0...COLUMNS).

Returns

The number of columns.

is_closed()

```
sparksee.ResultSet.is_closed (
              self)
```

Gets if the ResultSet instance has been closed or not.

See also

#close()

Returns

TRUE if the ResultSet instance has been closed, FALSE otherwise.

next()

```
sparksee.ResultSet.next (
              self)
```

Fetches the next row.

A ResultSet cursor is initially positioned before the first row; the first call to the method "Next" makes the first row the current row; the second call makes the second row the current row, and so on.

QueryExceptionIf a database access error occurs.

Returns

TRUE if the next row has been successfully fetched, FALSE otherwise.

5.58 sparksee.ResultSetList Class Reference

ResultSet list.

Public Member Functions

- **clear** (self)

  Clears the list.
- get (self, index)

  Returns the ResultSet at the specified position in the list.
- __iter__ (self)

  Gets a new ResultSetListIterator.
- iterator (self)

  Gets a new ResultSetListIterator.
- __init__ (self)

  Constructor.
- count (self)

  Number of elements in the list.

5.58.1 Detailed Description

ResultSet list.

It stores a ResultSet list.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.58.2 Constructor & Destructor Documentation

__init__()

```
sparksee.ResultSetList.__init__ (
              self)
```

Constructor.

This creates an empty list.

5.58.3 Member Function Documentation

__iter__()

```
sparksee.ResultSetList.__iter__ (
              self)
```

Gets a new ResultSetListIterator.

Returns

ResultSetListIterator instance

count()

```
sparksee.ResultSetList.count (
              self)
```

Number of elements in the list.

Returns

Number of elements in the list.

get()

```
sparksee.ResultSetList.get (
              self,
              index)
```

Returns the ResultSet at the specified position in the list.

Parameters

index	[in] Index of the element to return, starting at 0.

iterator()

```
sparksee.ResultSetList.iterator (
              self)
```

Gets a new ResultSetListIterator.

Returns

ResultSetListIterator instance.

5.59 sparksee.ResultSetListIterator Class Reference

ResultSetList iterator class.

Public Member Functions

- next (self)
  Moves to the next element.
- has_next (self)
  Gets if there are more elements.
- __next__ (self)
  Used in next().

5.59.1 Detailed Description

ResultSetList iterator class.

Iterator to traverse all the values into a ResultSetList instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.59.2 Member Function Documentation

__next__()

```
sparksee.ResultSetListIterator.__next__ (
              self)
```

Used in next().

Returns

The next element

has_next()

```
sparksee.ResultSetListIterator.has_next (
              self)
```

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

next()

```
sparksee.ResultSetListIterator.next (
              self)
```

Moves to the next element.

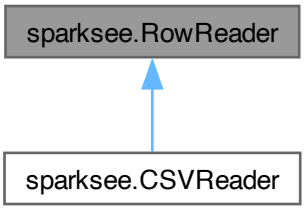
Returns

The next element.

5.60 sparksee.RowReader Class Reference

RowReader interface.

Inheritance diagram for sparksee.RowReader:



Public Member Functions

- close (self)
  Closes the reader.
- reset (self)
  Moves the reader to the beginning.
- read (self, row)
  Reads the next row as a string array.
- get_row (self)
  The row number for the current row.

5.60.1 Detailed Description

RowReader interface.

Common interface for those readers which get the data as an string array.

It works as follows: perform as many read operations as necessary and call close once at the end. Once close is called no more read operations can be executed.

Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.60.2 Member Function Documentation

close()

```
sparksee.RowReader.close (
              self)
```

Closes the reader.

Exceptions

IOError	If the close fails.

Reimplemented in sparksee.CSVReader.

get_row()

```
sparksee.RowReader.get_row (
              self)
```

The row number for the current row.

Returns

The current row number; 0 if there is no current row.

Exceptions

IOError	If it fails.

Reimplemented in sparksee.CSVReader.

read()

```
sparksee.RowReader.read (
              self,
              row)
```

Reads the next row as a string array.

Parameters

row	[out] A string list with each comma-separated element as a separate entry.

Returns

Returns true if a row had been read or false otherwise.

Exceptions

IOError	If bad things happen during the read.

Reimplemented in sparksee.CSVReader.

reset()

```
sparksee.RowReader.reset (
              self)
```

Moves the reader to the beginning.

Restarts the reader.

Returns

true if the reader can be restarted, false otherwise.

Exceptions

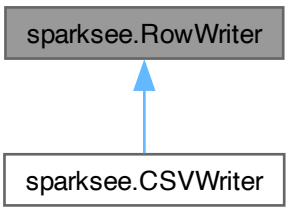
IOError	If bad things happen during the restart.

Reimplemented in sparksee.CSVReader.

5.61 sparksee.RowWriter Class Reference

RowWriter interface.

Inheritance diagram for sparksee.RowWriter:



Public Member Functions

- close (self)

  Closes the writer.
- write (self, row)

  Writes the next row.

5.61.1 Detailed Description

RowWriter interface.

Common interface for those writers which dump the data from an string array.

It works as follows: perform as many write operations as necessary and call close once at the end. Once close is called no more write operations can be executed.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.61.2 Member Function Documentation

close()

```
sparksee.RowWriter.close (
              self)
```

Closes the writer.

Exceptions

RuntimeError	null
IOError	If the close fails.

Reimplemented in sparksee.CSVWriter.

write()

```
sparksee.RowWriter.write (
              self,
              row)
```

Writes the next row.

Parameters

row	[in] Row of data.

Exceptions

RuntimeError	null
IOError	If bad things happen during the write.

Reimplemented in sparksee.CSVWriter.

5.62 sparksee.ScriptParser Class Reference

ScriptParser.

Public Member Functions

- **__init__** (self)
  Constructor.
- set_error_log (self, path)
  Sets the error log.
- generate_schema_script (self, path, db)
  Writes an script with the schema definition for the given database.
- parse (self, path, execute, locale_str)
  Parses the given input file.
- set_output_log (self, path)
  Sets the output log.

5.62.1 Detailed Description

ScriptParser.

The ScriptParser can create schemas and load data from a set of commands in a sparksee script.

A SPARKSEE script contains an ordered list of commands. ScriptParser will execute each one of them in order. Commands may create schemas, define nodes and edges, and load data into a previous defined SPARKSEE schema.

Check out the 'Scripting' chapter in the SPARKSEE User Manual for a comprehensive explanation on the grammar of the SPARKSEE commands and how they work.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.62.2 Member Function Documentation

generate_schema_script()

```
sparksee.ScriptParser.generate_schema_script (
              self,
              path,
              db)
```

Writes an script with the schema definition for the given database.

Parameters

path	[in] Filename of the script to be writen.
db	[in] Database.

Exceptions

IOError	If bad things happen opening or writing the file.

parse()

```
sparksee.ScriptParser.parse (
              self,
              path,
              execute,
              locale_str)
```

Parses the given input file.

Parameters

path	[in] Input file path.
execute	[in] If TRUE the script is executed, if FALSE it is just parsed.
locale_str	[in] The locale string for reading the input file. See CSVReader.

Returns

TRUE if ok, FALSE in case of error.

Exceptions

IOError	If bad things happen opening the file.

set_error_log()

```
sparksee.ScriptParser.set_error_log (
              self,
              path)
```

Sets the error log.

If not set, error log corresponds to standard error output.

Parameters

path	[in] Path of the error log.

Exceptions

IOError	If bad things happen opening the file.

set_output_log()

```
sparksee.ScriptParser.set_output_log (
              self,
              path)
```

Sets the output log.

If not set, output log corresponds to standard output.

Parameters

path	[in] Path of the output log.

Exceptions

IOError	If bad things happen opening the file.

5.63 sparksee.Session Class Reference

Session class.

Public Member Functions

- **rollback** (self)

  Rollbacks a transaction.
- pre_commit (self)

  PreCommits a transaction.
- **begin** (self)

  Begins a transaction.

- **begin_update** (self)

  Begins an update transaction.
- new_query (self, lang)

  Creates a new Query.
- **commit** (self)

  Commits a transaction.
- get_graph (self)

  Gets the Graph instance.
- close (self)

  Closes the Session instance.
- new_objects (self)

  Creates a new Objects instance.
- get_in_memory_pool_capacity (self)

  Gets the capacity of the in-memory pool.
- is_closed (self)

  Gets if Session instance has been closed or not.

5.63.1 Detailed Description

Session class.

A Session is a stateful period of activity of a user with the Database.

All the manipulation of a Database must be enclosed into a Session. A Session can be initiated from a Database instance and allows for getting a Graph instance which represents the persistent graph (the graph database).

Also, temporary data is associated to the Session, thus when a Session is closed, all the temporary data associated to the Session is removed too. Objects or Values instances or even session attributes are an example of temporary data.

Moreover, a Session is exclusive for a thread, thus if it is shared among threads results may be fatal or unexpected.

Check out the 'Processing' and 'Transactions' sections in the SPARKSEE User Manual for details about how Sessions work and the use of transactions.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.63.2 Member Function Documentation

close()

```
sparksee.Session.close (
              self)
```

Closes the Session instance.

It must be called to ensure the integrity of all data.

get_graph()

```
sparksee.Session.get_graph (
              self)
```

Gets the Graph instance.

Returns

The Graph instance.

get_in_memory_pool_capacity()

```
sparksee.Session.get_in_memory_pool_capacity (
              self)
```

Gets the capacity of the in-memory pool.

Returns

Returns the capacity of the in-memory pool

is_closed()

```
sparksee.Session.is_closed (
              self)
```

Gets if Session instance has been closed or not.

See also

close()

Returns

TRUE if the Session instance has been closed, FALSE otherwise.

new_objects()

```
sparksee.Session.new_objects (
              self)
```

Creates a new Objects instance.

Returns

The new Objects instance.

new_query()

```
sparksee.Session.new_query (
            self,
            lang)
```

Creates a new Query.

Parameters

lang	The query language to create the query for

Returns

A new Query instance.

pre_commit()

```
sparksee.Session.pre_commit (
            self)
```

PreCommits a transaction.

YOU SHOULD NOT USE THIS METHOD. This method exists for a specific service and it is used to force that the transaction is written in the recovery log even though it had not been committed yet (and may never be).

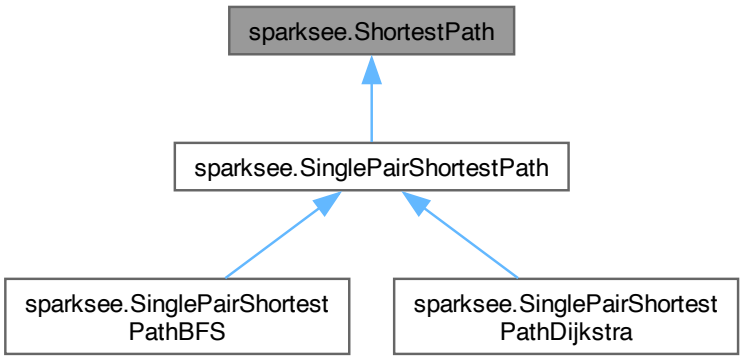
Returns

The transaction id. This has to be used in RedoPrecommited in case of a recovery process.

5.64 sparksee.ShortestPath Class Reference

ShortestPath class.

Inheritance diagram for sparksee.ShortestPath:



Public Member Functions

- add_edge_type (self, type, dir)
  Allows for traversing edges of the given type.
- run (self)
  Runs the algorithm.
- exclude_nodes (self, nodes)
  Set which nodes can't be used.
- add_all_edge_types (self, dir)
  Allows for traversing all edge types of the graph.
- add_node_type (self, type)
  Allows for traversing nodes of the given type.
- exclude_edges (self, edges)
  Set which edges can't be used.
- close (self)
  Closes the ShortestPath instance.
- set_maximum_hops (self, maxhops)
  Sets the maximum hops restriction.
- **add_all_node_types** (self)
  Allows for traversing all node types of the graph.
- is_closed (self)
  Gets if ShortestPath has been closed or not.

5.64.1 Detailed Description

ShortestPath class.

Classes implementing this abstract class solve the shortest path problem in a graph.

The user must set which node and edge types can be used for the traversal.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.64.2 Member Function Documentation

add_all_edge_types()

```
sparksee.ShortestPath.add_all_edge_types (
              self,
              dir)
```

Allows for traversing all edge types of the graph.

Parameters

dir	[in] Edge direction.

Reimplemented in sparksee.SinglePairShortestPath, sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijk

add_edge_type()

```
sparksee.ShortestPath.add_edge_type (
              self,
              type,
              dir)
```

Allows for traversing edges of the given type.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Reimplemented in sparksee.SinglePairShortestPath, sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijk

add_node_type()

```
sparksee.ShortestPath.add_node_type (
              self,
              type)
```

Allows for traversing nodes of the given type.

Parameters

type	null

Reimplemented in sparksee.SinglePairShortestPath, sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijk

close()

```
sparksee.ShortestPath.close (
              self)
```

Closes the ShortestPath instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.ShortestPath.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented in sparksee.SinglePairShortestPath, sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijk

exclude_nodes()

```
sparksee.ShortestPath.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented in sparksee.SinglePairShortestPath, sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijk

is_closed()

```
sparksee.ShortestPath.is_closed (
              self)
```

Gets if ShortestPath has been closed or not.

See also

close()

Returns

TRUE if the ShortestPath instance has been closed, FALSE otherwise.

run()

```
sparksee.ShortestPath.run (
              self)
```

Runs the algorithm.

This method can only be called once.

Reimplemented in sparksee.SinglePairShortestPath, sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijk

set_maximum_hops()

```
sparksee.ShortestPath.set_maximum_hops (
              self,
              maxhops)
```

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

Parameters

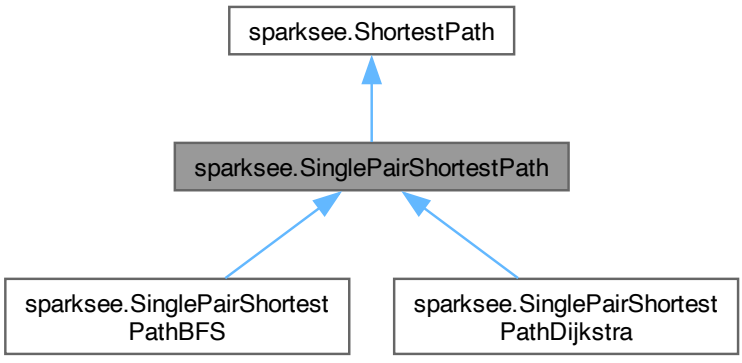
maxhops	[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.

Reimplemented in sparksee.SinglePairShortestPath, sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijk

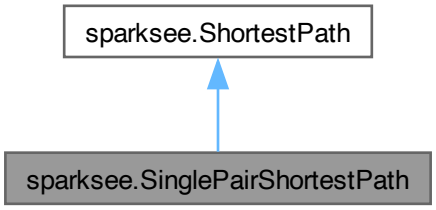
5.65 sparksee.SinglePairShortestPath Class Reference

SinglePairShortestPath class.

Inheritance diagram for sparksee.SinglePairShortestPath:



Collaboration diagram for sparksee.SinglePairShortestPath:



Public Member Functions

- run (self)

  Runs the algorithm.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- add_all_edge_types (self, dir)

  Allows for traversing all edge types of the graph.
- exclude_edges (self, edges)

  Set which edges can't be used.
- add_edge_type (self, type, dir)

  Allows for traversing edges of the given type.
- get_path_as_nodes (self)

  Gets the shortest path between the source node and the destination node as an ordered set of nodes.
- get_path_as_edges (self)

  Gets the shortest path between the source node and the destination node as an ordered set of edges.
- add_node_type (self, type)

  Allows for traversing nodes of the given type.
- **exists** (self)

  Returns TRUE If a path exists or FALSE otherwise.
- get_cost (self)

  Gets the cost of the shortest path.
- set_maximum_hops (self, maxhops)

  Sets the maximum hops restriction.
- **add_all_node_types** (self)

  Allows for traversing all node types of the graph.
- close (self)

  Closes the ShortestPath instance.
- is_closed (self)

  Gets if ShortestPath has been closed or not.

5.65.1 Detailed Description

SinglePairShortestPath class.

Classes implementing this abstract class solve the shortest path problem in a graph from a given source node and to a given destination node.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.65.2 Member Function Documentation

add_all_edge_types()

```
sparksee.SinglePairShortestPath.add_all_edge_types (
              self,
              dir)
```

Allows for traversing all edge types of the graph.

Parameters

dir	[in] Edge direction.

Reimplemented from sparksee.ShortestPath.

Reimplemented in sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijkstra.

add_edge_type()

```
sparksee.SinglePairShortestPath.add_edge_type (
              self,
              type,
              dir)
```

Allows for traversing edges of the given type.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Reimplemented from sparksee.ShortestPath.

Reimplemented in sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijkstra.

add_node_type()

```
sparksee.SinglePairShortestPath.add_node_type (
              self,
              type)
```

Allows for traversing nodes of the given type.

Parameters

type	null

Reimplemented from sparksee.ShortestPath.

Reimplemented in sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijkstra.

close()

```
sparksee.ShortestPath.close (
              self)  [inherited]
```

Closes the ShortestPath instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.SinglePairShortestPath.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.ShortestPath.

Reimplemented in sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijkstra.

exclude_nodes()

```
sparksee.SinglePairShortestPath.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.ShortestPath.

Reimplemented in sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijkstra.

get_cost()

```
sparksee.SinglePairShortestPath.get_cost (
              self)
```

Gets the cost of the shortest path.

The cost for unweighted algorithms is the number of hops of the shortest path. For weighted algoritms, the cost is the sum of the costs of the edges of the shortest path.

Returns

The cost of the shortest path.

Reimplemented in sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijkstra.

get_path_as_edges()

```
sparksee.SinglePairShortestPath.get_path_as_edges (
              self)
```

Gets the shortest path between the source node and the destination node as an ordered set of edges.

Returns

Ordered set of edge identifiers.

Reimplemented in sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijkstra.

get_path_as_nodes()

```
sparksee.SinglePairShortestPath.get_path_as_nodes (
              self)
```

Gets the shortest path between the source node and the destination node as an ordered set of nodes.

Returns

Ordered set of node identifiers.

Reimplemented in sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijkstra.

is_closed()

```
sparksee.ShortestPath.is_closed (
              self)  [inherited]
```

Gets if ShortestPath has been closed or not.

See also

close()

Returns

TRUE if the ShortestPath instance has been closed, FALSE otherwise.

run()

```
sparksee.SinglePairShortestPath.run (
              self)
```

Runs the algorithm.

This method can only be called once.

Reimplemented from sparksee.ShortestPath.

Reimplemented in sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijkstra.

set_maximum_hops()

```
sparksee.SinglePairShortestPath.set_maximum_hops (
            self,
            maxhops)
```

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

Parameters

maxhops	[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.

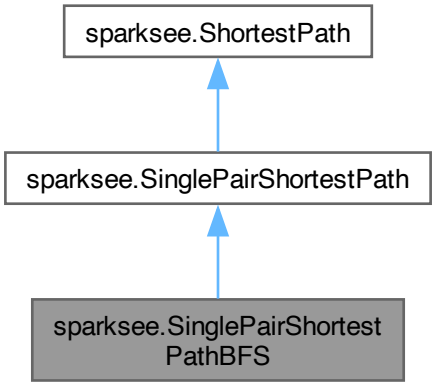
Reimplemented from sparksee.ShortestPath.

Reimplemented in sparksee.SinglePairShortestPathBFS, and sparksee.SinglePairShortestPathDijkstra.

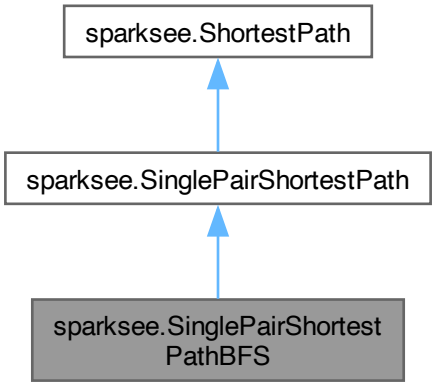
5.66 sparksee.SinglePairShortestPathBFS Class Reference

SinglePairShortestPathBFS class.

Inheritance diagram for sparksee.SinglePairShortestPathBFS:



Collaboration diagram for sparksee.SinglePairShortestPathBFS:



Public Member Functions

- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- get_cost (self)

  Gets the cost of the shortest path.
- **run** (self)

  Executes the algorithm.
- add_all_edge_types (self, dir)

  Allows for traversing all edge types of the graph.
- exclude_edges (self, edges)

  Set which edges can't be used.
- __init__ (self, session, src, dst)

  Creates a new instance.
- add_edge_type (self, type, dir)

  Allows for traversing edges of the given type.
- get_path_as_nodes (self)

  Gets the shortest path between the source node and the destination node as an ordered set of nodes.
- add_node_type (self, type)

  Allows for traversing nodes of the given type.
- **exists** (self)

  Returns TRUE If a path exists or FALSE otherwise.
- check_only_existence (self)

  Set that only the path existence must be calculated and not the path itself.
- get_path_as_edges (self)

  Gets the shortest path between the source node and the destination node as an ordered set of edges.
- set_maximum_hops (self, maxhops)

  Sets the maximum hops restriction.
- **add_all_node_types** (self)

  Allows for traversing all node types of the graph.

- close (self)

  Closes the ShortestPath instance.
- is_closed (self)

  Gets if ShortestPath has been closed or not.

5.66.1 Detailed Description

SinglePairShortestPathBFS class.

It solves the single-pair shortest path problem using a BFS-based implementation.

It is a unweighted algorithm, that is it assumes all edges have the same cost.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.66.2 Constructor & Destructor Documentation

__init__()

```
sparksee.SinglePairShortestPathBFS.__init__ (
              self,
              session,
              src,
              dst)
```

Creates a new instance.

Parameters

session	[in] Session to get the graph from and perform traversal.
src	[in] Source node.
dst	[dst] Destination node.

5.66.3 Member Function Documentation

add_all_edge_types()

```
sparksee.SinglePairShortestPathBFS.add_all_edge_types (
              self,
              dir)
```

Allows for traversing all edge types of the graph.

Parameters

dir	[in] Edge direction.

Reimplemented from sparksee.SinglePairShortestPath.

add_edge_type()

```
sparksee.SinglePairShortestPathBFS.add_edge_type (
              self,
              type,
              dir)
```

Allows for traversing edges of the given type.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Reimplemented from sparksee.SinglePairShortestPath.

add_node_type()

```
sparksee.SinglePairShortestPathBFS.add_node_type (
              self,
              type)
```

Allows for traversing nodes of the given type.

Parameters

type	null

Reimplemented from sparksee.SinglePairShortestPath.

check_only_existence()

```
sparksee.SinglePairShortestPathBFS.check_only_existence (
              self)
```

Set that only the path existence must be calculated and not the path itself.

That method should improve the performance of the algorithm, but a call to GetPathAsNodes or GetPathAsEdges will generate an exception even if the path exists.

close()

```
sparksee.ShortestPath.close (
              self)  [inherited]
```

Closes the ShortestPath instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.SinglePairShortestPathBFS.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.SinglePairShortestPath.

exclude_nodes()

```
sparksee.SinglePairShortestPathBFS.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.SinglePairShortestPath.

get_cost()

```
sparksee.SinglePairShortestPathBFS.get_cost (
              self)
```

Gets the cost of the shortest path.

The cost is the number of hops of the shortest path.

Returns

The cost of the shortest path.

Reimplemented from sparksee.SinglePairShortestPath.

get_path_as_edges()

```
sparksee.SinglePairShortestPathBFS.get_path_as_edges (
              self)
```

Gets the shortest path between the source node and the destination node as an ordered set of edges.

Returns

Ordered set of edge identifiers.

Reimplemented from sparksee.SinglePairShortestPath.

get_path_as_nodes()

```
sparksee.SinglePairShortestPathBFS.get_path_as_nodes (
              self)
```

Gets the shortest path between the source node and the destination node as an ordered set of nodes.

Returns

Ordered set of node identifiers.

Reimplemented from sparksee.SinglePairShortestPath.

is_closed()

```
sparksee.ShortestPath.is_closed (
              self)  [inherited]
```

Gets if ShortestPath has been closed or not.

See also

close()

Returns

TRUE if the ShortestPath instance has been closed, FALSE otherwise.

set_maximum_hops()

```
sparksee.SinglePairShortestPathBFS.set_maximum_hops (
            self,
            maxhops)
```

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

Parameters

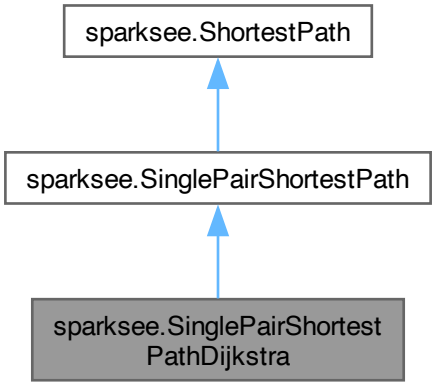
maxhops	[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.

Reimplemented from sparksee.SinglePairShortestPath.

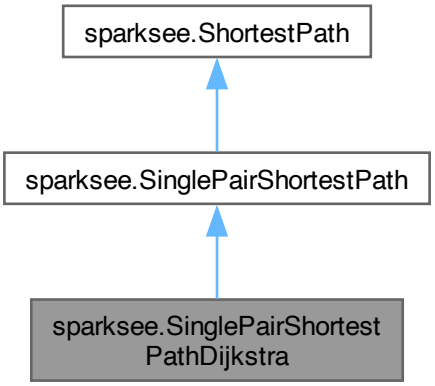
5.67 sparksee.SinglePairShortestPathDijkstra Class Reference

SinglePairShortestPathDijkstra class.

Inheritance diagram for sparksee.SinglePairShortestPathDijkstra:



Collaboration diagram for sparksee.SinglePairShortestPathDijkstra:



Public Member Functions

- __init__ (self, session, src, dst)
  Creates a new instance.
- exclude_nodes (self, nodes)
  Set which nodes can't be used.
- get_cost (self)
  Gets the cost of the shortest path.
- **run** (self)
  Executes the algorithm.
- add_all_edge_types (self, dir)
  Allows for traversing all edge types of the graph.
- exclude_edges (self, edges)
  Set which edges can't be used.
- add_edge_type (self, type, dir)
  Allows for traversing edges of the given type.
- set_unweighted_edge_cost (self, weight)
  Sets the weight assigned to the unweighted edges.
- get_path_as_nodes (self)
  Gets the shortest path between the source node and the destination node as an ordered set of nodes.
- set_dynamic_edge_cost_callback (self, dyn_cost_calculator)
  Set a class callback to dynamically calculate the cost of the edges.
- add_node_type (self, type)
  Allows for traversing nodes of the given type.
- **exists** (self)
  Returns TRUE If a path exists or FALSE otherwise.
- get_path_as_edges (self)
  Gets the shortest path between the source node and the destination node as an ordered set of edges.
- add_weighted_edge_type (self, type, dir, attr)
  Allows for traversing edges of the given type using the given attribute as the weight.

- set_maximum_hops (self, maxhops)

  Sets the maximum hops restriction.
- **add_all_node_types** (self)

  Allows for traversing all node types of the graph.
- close (self)

  Closes the ShortestPath instance.
- is_closed (self)

  Gets if ShortestPath has been closed or not.

5.67.1 Detailed Description

SinglePairShortestPathDijkstra class.

It solves the single-pair shortest path problem using a Dijkstra-based implementation.

It is a weighted algorithm, so it takes into account the cost of the edges to compute a minimum-cost shortest path. That is, the user may set for each edge type which attribute should be used to retrieve the cost of the edge. If no attribute is given for an edge type, this will assume the edge has a fixed cost (the default is 1). Only numerical attribute can be set as weight attributes (that is Long, Integer or Double attributes are allowed).

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.67.2 Constructor & Destructor Documentation

__init__()

```
sparksee.SinglePairShortestPathDijkstra.__init__ (
              self,
              session,
              src,
              dst)
```

Creates a new instance.

Parameters

session	[in] Session to get the graph from and perform traversal.
src	[in] Source node.
dst	[dst] Destination node.

5.67.3 Member Function Documentation

add_all_edge_types()

```
sparksee.SinglePairShortestPathDijkstra.add_all_edge_types (
              self,
              dir)
```

Allows for traversing all edge types of the graph.

Parameters

dir	[in] Edge direction.

Reimplemented from sparksee.SinglePairShortestPath.

add_edge_type()

```
sparksee.SinglePairShortestPathDijkstra.add_edge_type (
              self,
              type,
              dir)
```

Allows for traversing edges of the given type.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Reimplemented from sparksee.SinglePairShortestPath.

add_node_type()

```
sparksee.SinglePairShortestPathDijkstra.add_node_type (
              self,
              type)
```

Allows for traversing nodes of the given type.

Parameters

type	null

Reimplemented from sparksee.SinglePairShortestPath.

add_weighted_edge_type()

```
sparksee.SinglePairShortestPathDijkstra.add_weighted_edge_type (
              self,
              type,
              dir,
              attr)
```

Allows for traversing edges of the given type using the given attribute as the weight.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.
attr	[in] Attribute to be used as the weight. It must be a global attribute or an attribute of the given edge type.

close()

```
sparksee.ShortestPath.close (
              self)  [inherited]
```

Closes the ShortestPath instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.SinglePairShortestPathDijkstra.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.SinglePairShortestPath.

exclude_nodes()

```
sparksee.SinglePairShortestPathDijkstra.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.SinglePairShortestPath.

get_cost()

```
sparksee.SinglePairShortestPathDijkstra.get_cost (
              self)
```

Gets the cost of the shortest path.

The cost is the sum of the weights of the edges in the shortest path.

Returns

The cost of the shortest path.

Reimplemented from sparksee.SinglePairShortestPath.

get_path_as_edges()

```
sparksee.SinglePairShortestPathDijkstra.get_path_as_edges (
              self)
```

Gets the shortest path between the source node and the destination node as an ordered set of edges.

Returns

Ordered set of edge identifiers.

Reimplemented from sparksee.SinglePairShortestPath.

get_path_as_nodes()

```
sparksee.SinglePairShortestPathDijkstra.get_path_as_nodes (
              self)
```

Gets the shortest path between the source node and the destination node as an ordered set of nodes.

Returns

Ordered set of node identifiers.

Reimplemented from sparksee.SinglePairShortestPath.

is_closed()

```
sparksee.ShortestPath.is_closed (
              self)  [inherited]
```

Gets if ShortestPath has been closed or not.

See also

close()

Returns

TRUE if the ShortestPath instance has been closed, FALSE otherwise.

set_dynamic_edge_cost_callback()

```
sparksee.SinglePairShortestPathDijkstra.set_dynamic_edge_cost_callback (
              self,
              dyn_cost_calculator)
```

Set a class callback to dynamically calculate the cost of the edges.

The callback can be set to NULL (the default) to use the normal attribute based cost weights. The given class must be kept alive by the user for as long as the algorithm is running.

Parameters

dyn_cost_calculator	[in] Class callback to calculate the edge costs

set_maximum_hops()

```
sparksee.SinglePairShortestPathDijkstra.set_maximum_hops (
              self,
              maxhops)
```

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

Parameters

maxhops	[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.

Reimplemented from sparksee.SinglePairShortestPath.

set_unweighted_edge_cost()

```
sparksee.SinglePairShortestPathDijkstra.set_unweighted_edge_cost (
              self,
              weight)
```

Sets the weight assigned to the unweighted edges.

All the edges from the types added without an explicit weight attribute will get this weight. The default weight for this edges is 1.

Parameters

weight	[in] The weight value for unweighted edges.

5.68 sparksee.SinglePairShortestPathDijkstraDynamicCost Class Reference

Defines how to calculate an edge weight.

Public Member Functions

- calculate_edge_cost (self, source_node, source_cost, source_level, target_node, edge, edge_weight_attr)

  Dynamic calculation of an edge weight.

5.68.1 Detailed Description

Defines how to calculate an edge weight.

This is an interface which must be implemented by the user. While the algorithm is running, one or more calls for each edge that can be in the shortest path will be issued to get the weight of the edge that could change based on the predecessor node and the current minimum path from the source.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.68.2 Member Function Documentation

calculate_edge_cost()

```
sparksee.SinglePairShortestPathDijkstraDynamicCost.calculate_edge_cost (
              self,
              source_node,
              source_cost,
              source_level,
              target_node,
              edge,
              edge_weight_attr)
```

Dynamic calculation of an edge weight.

This method will be called several times during the algorithm execution. It can be called for the same edge multiple times because the path to the source node (and as a consequence the source weight and levels) might change while the algorithm is running. This method will have the normal default implementation that only uses the specified edge attribute to get the weight or the unweighted edge cost if the edge doesn't have the specified cost attribute. But it can be overriden in a derived user class to implement a dinamic edge weight, that may need the sourceCost and sourceLevels to calculate the weight.

Parameters

source_node	[in] The current node
source_cost	[in] Current total cost up to the source node
source_level	[in] Current path size up to the source node
target_node	[in] The next node
edge	[in] The edge to the next node that this methos must weight
edge_weight_attr	[in] The attribute that stores the edge weight or InvalidAttribute

Returns

Returns the current weight of the edge (>=0) or <0 if the edge can not be used.

5.69 sparksee.Sparksee Class Reference

Sparksee class.

Public Member Functions

- restore_encrypted_backup (self, path, backup_file, config, key_in_hex, iv_in_hex)

  Restores a Database from an encrypted backup file.
- resize_in_memory_pool (self, new_capacity)

  Resizes the in memory pool allocator.
- create (self, path, alias, config)

  Creates a new Database instance.
- verify_checksums (self, path)

  Verifies the Checksum integrity of the given image file and it's recovery log file.

- remove_checksums (self, path)

  Converts a database WITH checksums into a database WITHOUT checksums.
- restore (self, path, backup_file)

  Restores a Database from a backup file.
- add_checksums (self, path)

  Converts a database WITHOUT checksums into a database WITH checksums.
- rebuild_database (self, path, config, minimize_disk)

  Rebuild the database from scratch with the same data.
- restore_encrypted_backup (self, path, backup_file, key_in_hex, iv_in_hex)

  Restores a Database from an encrypted backup file.
- decrypt (self, path)

  Converts a database WITH encryption into a database WITHOUT encryption.
- restore (self, path, backup_file, config)

  Restores a Database from a backup file.
- create (self, path, alias)

  Creates a new Database instance.
- open (self, path, read_only)

  Opens an existing Database instance.
- __init__ (self, config)

  Creates a new instance.
- open (self, path, read_only, config)

  Opens an existing Database instance.
- set_unrecoverable_error_callback (self, callback)

  Sets the unrecoverable error callback.
- encrypt (self, path)

  Converts a database WITHOUT encryption into an encrypted database.
- close (self)

  Closes the Sparksee instance.
- get_in_memory_pool_capacity (self)

  Gets the capacity of the in-memory pool (in Megabytes).

Static Public Attributes

- float **VERSION** = 5.0.0

  Sparksee version.

5.69.1 Detailed Description

Sparksee class.

All Sparksee programs must have one single Sparksee instance to manage one or more Database instances.

This class allows for the creation of new Databases or open an existing one.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.69.2 Constructor & Destructor Documentation

__init__()

```
sparksee.Sparksee.__init__ (
              self,
              config)
```

Creates a new instance.

Parameters

config	[in/out] Sparksee configuration (may be updated).

5.69.3 Member Function Documentation

add_checksums()

```
sparksee.Sparksee.add_checksums (
              self,
              path)
```

Converts a database WITHOUT checksums into a database WITH checksums.

Any error found is logged. The database (and it's recovery log if present) does NOT have checksums. sparksee::↩ gdb::FileNotFoundExceptionsparksee::gdb::Error

Parameters

path	The path to the database image file

Returns

Returns true if the checksum could be added or false otherwise.

Exceptions

RuntimeError	null
IOError	null

close()

```
sparksee.Sparksee.close (
              self)
```

Closes the Sparksee instance.

It must be called to ensure the integrity of all data.

create() [1/2]

```
sparksee.Sparksee.create (
              self,
              path,
              alias)
```

Creates a new Database instance.

Errorfor other issues.

Parameters

path	[in] Database storage file.
alias	[in] Database alias name.

Returns

A Database instance.

Exceptions

RuntimeError	null
IOError	If the given file cannot be created.

create() [2/2]

```
sparksee.Sparksee.create (
              self,
              path,
              alias,
              config)
```

Creates a new Database instance.

Errorfor other issues.

Parameters

path	[in] Database storage file.
alias	[in] Database alias name.
config	[in] Use a specific configuration instead of the one in this Sparksee

Returns

A Database instance.

Exceptions

RuntimeError	null
IOError	If the given file cannot be created.

decrypt()

```
sparksee.Sparksee.decrypt (
              self,
              path)
```

Converts a database WITH encryption into a database WITHOUT encryption.

Any error found is logged. The database (and it's recovery log if present) is encrypted. The current encryption had been configured using SparkseeConfig. sparksee::gdb::FileNotFoundExceptionsparksee::gdb::Error

Parameters

path	The path to the database image file

Returns

Returns true if the database could be unencrypted. The errors are returned with an exception.

Exceptions

RuntimeError	null
IOError	null

encrypt()

```
sparksee.Sparksee.encrypt (
              self,
              path)
```

Converts a database WITHOUT encryption into an encrypted database.

Any error found is logged. The database (and it's recovery log if present) is NOT encrypted. The encryption had already been configured using SparkseeConfig. sparksee::gdb::FileNotFoundExceptionsparksee::gdb::Error

Parameters

path	The path to the database image file

Returns

Returns true if the database could be encrypted. The errors are returned with an exception.

Exceptions

RuntimeError	null
IOError	null

get_in_memory_pool_capacity()

```
sparksee.Sparksee.get_in_memory_pool_capacity (
              self)
```

Gets the capacity of the in-memory pool (in Megabytes).

Returns

Returns the capacity of the in memory pool

open() [1/2]

```
sparksee.Sparksee.open (
              self,
              path,
              read_only)
```

Opens an existing Database instance.

Errorfor other issues.

Parameters

path	[in] Database storage file.
read_only	[in] If TRUE, open Database in read-only mode.

Returns

A Database instance.

Exceptions

RuntimeError	null
IOError	If the given file does not exist.

open() [2/2]

```
sparksee.Sparksee.open (
              self,
              path,
              read_only,
              config)
```

Opens an existing Database instance.

The provided configuration will be used for all the database settings but not for the license, which must already be properly setup in this class. Errorfor other issues.

Parameters

path	[in] Database storage file.
read_only	[in] If TRUE, open Database in read-only mode.
config	[in] Use a specific configuration instead of the one in this Sparksee

Returns

A Database instance.

Exceptions

RuntimeError	null
IOError	If the given file does not exist.

rebuild_database()

```
sparksee.Sparksee.rebuild_database (
              self,
              path,
              config,
              minimize_disk)
```

Rebuild the database from scratch with the same data.

It will create a new database on the same file with the same data but rebuilt from scratch, so the internal OIDs may change and the storage usage may be compacted.

It's the equivalent to extract a backup of the data and restore it. This operation can take a long time and the result file should be smaller than the original if there had been data removals or the data was not loaded in the most optimal order. But the file size could also end up being slightly bigger the the original in some cases (the data creation in the database was optimal and nothing had beed removed).

WARNING: If anything goes wrong while the database is being rebuilt you may lose the database. The data should not be lost, but it may require an additional process to recover it.

To avoid data loss, the next open after a failed rebuild operation will attempt to automatically recover the database by the fastest solution available (restoring the original file, finishing the almost completed rebuild or restoring a backup).

The source database is also kept until the rebuilt database is available to reduce the recovery time after a rebuild failure and avoid data loss risks. But if reducing the temporary disk usage is more important, you can use the minimizeDisk to delete it as soon as possible instead.

The final rebuilt database size will be the same whether you use the minimizeDisk argument or not.

Parameters

path	[In] The Sparksee database to rebuild.
config	[in] Use a specific configuration instead of the one in this Sparksee.
minimize_disk	[in] Use less temporary disk space.

Exceptions

RuntimeError	null
IOError	null

remove_checksums()

```
sparksee.Sparksee.remove_checksums (
                self,
                path)
```

Converts a database WITH checksums into a database WITHOUT checksums.

Any error found is logged. The database (and it's recovery log if present) have checksums. sparksee::gdb::File↩ NotFoundExceptionsparksee::gdb::Error

Parameters

path	The path to the database image file

Returns

Returns true if the checksum could be removed. The errors are returned with an exception.

Exceptions

RuntimeError	null
IOError	null

resize_in_memory_pool()

```
sparksee.Sparksee.resize_in_memory_pool (
                self,
                new_capacity)
```

Resizes the in memory pool allocator.

sparksee::gdb::SystemError

Parameters

new_capacity	The new capacity of the in memory pool allocator

Returns

Returns true if was successful

restore() [1/2]

```
sparksee.Sparksee.restore (
              self,
              path,
              backup_file)
```

Restores a Database from a backup file.

See the Graph class Backup method. Errorfor other issues.

Parameters

path	[in] Database storage file.
backup_file	[in] The Backup file to be restored.

Returns

A Database instance.

Exceptions

RuntimeError	null
IOError	If the given file cannot be created, or the exported data file does not exists.

restore() [2/2]

```
sparksee.Sparksee.restore (
              self,
              path,
              backup_file,
              config)
```

Restores a Database from a backup file.

See the Graph class Backup method. Errorfor other issues.

Parameters

path	[in] Database storage file.
backup_file	[in] The Backup file to be restored.
config	[in] Use a specific configuration instead of the one in this Sparksee

Returns

A Database instance.

Exceptions

RuntimeError	null
IOError	If the given file cannot be created, or the exported data file does not exists.

restore_encrypted_backup() [1/2]

```
sparksee.Sparksee.restore_encrypted_backup (
                self,
                path,
                backup_file,
                config,
                key_in_hex,
                iv_in_hex)
```

Restores a Database from an encrypted backup file.

See the Graph class EncryptedBackup method. Errorfor other issues.

Parameters

path	[in] Database storage file.
backup_file	[in] The Backup file to be restored.
config	[in] Use a specific configuration instead of the one in this Sparksee
key_in_hex	[In] The AES encryption Key of the backup file as an hexadecimal string (8, 16 or 32 bytes).
iv_in_hex	[In] The AES Initialization Vector of the backup file as an hexadecimal string (16 bytes).

Returns

A Database instance.

Exceptions

RuntimeError	null
IOError	If the given file cannot be created, or the exported data file does not exists.

restore_encrypted_backup() [2/2]

```
sparksee.Sparksee.restore_encrypted_backup (
                self,
                path,
                backup_file,
                key_in_hex,
                iv_in_hex)
```

Restores a Database from an encrypted backup file.

See the Graph class EncryptedBackup method. Errorfor other issues.

Parameters

path	[in] Database storage file.
backup_file	[in] The Backup file to be restored.
key_in_hex	[In] The AES encryption Key of the backup file as an hexadecimal string (8, 16 or 32 bytes).
iv_in_hex	[In] The AES Initialization Vector of the backup file as an hexadecimal string (16 bytes).

Returns

A Database instance.

Exceptions

RuntimeError	null
IOError	If the given file cannot be created, or the exported data file does not exists.

set_unrecoverable_error_callback()

```
sparksee.Sparksee.set_unrecoverable_error_callback (
                self,
                callback)
```

Sets the unrecoverable error callback.

Parameters

callback	The unrecoverable error callback to set

verify_checksums()

```
sparksee.Sparksee.verify_checksums (
              self,
              path)
```

Verifies the Checksum integrity of the given image file and it's recovery log file.

The main database file checksums are always checked.

When the recovery log file exists, the recovery log file is also checked for checksum and format errors.

When the checksums of the main file or the recovery file are incorrect, the funcion returns false, in any other error it throws an exception.

Any error found is logged as a WARNING.

sparksee::gdb::FileNotFoundExceptionsparksee::gdb::Error

Parameters

path	The path to the image file

Returns

Returns true if the checksum verification succeeded. Returns false if an invalid checksum was detected.

Exceptions

RuntimeError	null
IOError	null

5.70 sparksee.SparkseeConfig Class Reference

Sparksee configuration class.

Public Member Functions

- set_recovery_enabled (self, status)
  Enables or disables the recovery.
- set_license_pre_download_days (self, days)
  Start trying to automatically download a new license at the specified number of days before expiration.
- get_extent_pages (self)
  Gets the number of pages per extent.
- save_all (self)
  Save all the current configuration settings in the specified config file.
- get_a_e_s_key (self)
  Get the AES encryption key in a hexadecimal encoded string.
- get_pool_persistent_max_size (self)
  Gets the maximum size for the persistent pool in number of frames.

- get_pool_temporary_min_size (self)

  Gets the minimum size for the temporary pool in number of frames.
- get_call_stack_dump (self)

  Gets whether the signals will be captured to dump the call stack or not.
- set_recovery_cache_max_size (self, extents)

  Sets the maximum size for the recovery log cache in extents.
- set_a_e_s_encryption_enabled (self, key_in_hex, iv_in_hex)

  Enables storage encryption using AES with the given key and iv.
- set_plugins_dir (self, dir_path)

  Sets the Plugins directory.
- get_high_availability_enabled (self)

  Gets whether high availability mode is enabled or disabled.
- get_pool_persistent_min_size (self)

  Gets the minimum size for the persistent pool in number of frames.
- set_checksum_enabled (self, status)

  Enables or disables the storage checksum usage.
- set_pool_persistent_min_size (self, frames)

  Sets the minimum size for the persistent pool in number of frames.
- set_cache_statistics_enabled (self, status)

  Enables or disables cache statistics.
- set_recovery_checkpoint_time (self, micro_seconds)

  Sets the delay time (in microseconds) between automatic checkpoints.
- **set_encryption_disabled** (self)

  Disables storage encryption.
- set_recovery_log_file (self, file_path)

  Sets the recovery log file.
- set_pool_frame_size (self, extents)

  Sets the size of a pool frame in number of extents.
- get_recovery_log_file (self)

  Gets the recovery log file.
- set_high_availability_enabled (self, status)

  Enables or disables high availability mode.
- get_cache_statistics_snapshot_time (self)

  Gets the cache statistics snapshot time in microseconds.
- get_extent_size (self)

  Gets the size of a extent.
- get_sparksee_config_file (self)

  Gets the config file path.
- set_extent_size (self, k_bytes)

  Sets the size of the extents in KB.
- get_plugins_dir (self)

  Gets the Plugins directory.
- set_high_availability_i_p (self, ip)

  Sets the IP address and port of the instance.
- get_pool_temporary_max_size (self)

  Gets the maximum size for the temporary pool in number of frames.
- set_in_mem_alloc_size (self, size)

  Sets the in-memory allocator size.
- __init__ (self, path)

  Creates a new instance with a specific config file.
- set_high_availability_master_history (self, file_path)

Sets the master's history log.

- set_tmp_enabled (self, enabled)

  Sets whether to use temporary storage for computations or not.
- get_log_file (self)

  Gets the log file.
- get_tmp_enabled (self)

  Gets whether using temporary storage for computations is enabled.
- get_pool_frame_size (self)

  Gets the size of a pool frame in number of extents.
- get_cache_statistics_enabled (self)

  Gets whether cache statistics are enabled or disabled.
- set_high_availability_coordinators (self, ip)

  Sets the coordinators address and port list.
- set_high_availability_synchronization (self, micro_seconds)

  Sets the synchronization polling time.
- get_client_id (self)

  Gets the client identifier.
- get_cache_statistics_file (self)

  Gets the cache statistics log file.
- set_sparksee_config_file (self, path)

  Sets the config file path.
- save (self)

  Save the current configuration in the specified config file.
- set_pool_persistent_max_size (self, frames)

  Sets the maximum size for the persistent pool in number of frames.
- get_log_level (self)

  Gets the log level.
- set_license (self, key)

  Sets the license key.
- get_download_status (self)

  Gets a message with the license download result.
- set_license_id (self, license_id)

  Set the license identifier.
- get_license (self)

  Gets the license key.
- get_high_availability_i_p (self)

  Gets the IP address and port of the instance.
- set_a_e_s_encryption_enabled (self, key_size)

  Enables storage encryption using AES and GENARATES a key and an iv.
- get_in_mem_alloc_size (self)

  Gets the in-memory allocator size.
- get_license_pre_download_days (self)

  Get the number of days before expiration when a new license will be downloaded.
- set_call_stack_dump (self, status)

  Sets whether the signals will be captured to dump the call stack or not.
- download_license (self)

  Try to download a license key Can be used to explicitly download a license when the autodownload is disabled (LicensePreDownloadDays == -1).
- get_recovery_enabled (self)

  Gets whether the recovery is enabled or disabled.
- **get_license_request** (self)

Get information useful to manually request a license.

- set_cache_statistics_file (self, file_path)

  Sets the cache statistics log file.
- get_tmp_folder (self)

  Gets the temporary folder used for temporary staging.
- get_pool_partitions (self)

  Gets the number of partitions in each PartitionedPool.
- get_high_availability_coordinators (self)

  Gets the coordinators address and port list.
- get_recovery_checkpoint_time (self)

  Gets the delay time (in microseconds) between automatic checkpoints.
- set_log_file (self, file_path)

  Sets the log file.
- set_extent_pages (self, pages)

  Sets the number of pages per extent.
- set_client_id (self, client_id)

  Set the client identifier.
- set_log_level (self, level)

  Sets the log level.
- get_recovery_cache_max_size (self)

  Gets the maximum size for the recovery log cache in extents.
- get_checksum_enabled (self)

  Gets whether the storage checksum usage is enabled or disabled.
- get_high_availability_synchronization (self)

  Gets the synchronization polling time.
- get_high_availability_master_history (self)

  Gets the master's history log.
- get_license_id (self)

  Gets the license identifier.
- __init__ (self, path, client_id, license_id)

  Creates a new instance with a specific config file and IDs.
- get_a_e_s_i_v (self)

  Get the AES initialization vector in a hexadecimal encoded string.
- get_encryption_enabled (self)

  Gets whether the storage encryption is enabled or disabled.
- set_pool_partitions (self, pools)

  Sets the number of pools in each PartitionedPool.
- get_rollback_enabled (self)

  Gets whether the rollback is enabled or disabled.
- set_pool_temporary_max_size (self, frames)

  Sets the maximum size for the temporary pool in number of frames.
- get_cache_max_size (self)

  Gets the maximum size for the cache (all pools) in MB.
- set_rollback_enabled (self, status)

  Enables or disables the rollback.
- set_tmp_folder (self, tmp_folder)

  Sets the temporary folder used for temporary staging.
- set_cache_statistics_snapshot_time (self, micro_seconds)

  Sets the cache statistics snapshot time.
- set_cache_max_size (self, mega_bytes)

  Sets the maximum size for the cache (all pools) in MB.

- __init__ (self)

  Creates a new instance.
- set_pool_temporary_min_size (self, frames)

  Sets the minimum size for the temporary pool in number of frames.
- **download_expected** (self)

  Check if a new license will be automatically downloaded with the current settings.

5.70.1 Detailed Description

Sparksee configuration class.

If not specified, 0 means unlimited which is the maximum available. For the pools that's the total cache size. For the cache unlimited means nearly all the physical memory of the computer.

For each field, there is a default value. This value can be overrided with values from a properties file (see SparkseeProperties class). Also, this settings can be overrided calling a specific setter.

For each field, it is shown its default value and the property to override this value:

Extent size: 4KB ('sparksee.storage.extentsize' at SparkseeProperties).

Pages per extent: 1 page ('sparksee.storage.extentpages' at SparkseeProperties).

Checksums enabled: true ('sparksee.storage.checksum' at SparkseeProperties).

Pool frame size: 1 extent ('sparksee.io.pool.frame.size' at SparkseeProperties).

Minimum size for the persistent pool: 64 frames ('sparksee.io.pool.persistent.minsize' at SparkseeProperties).

Maximum size for the persistent pool: 0 frames ('sparksee.io.pool.persistent.maxsize' at SparkseeProperties).

Minimum size for the temporary pool: 16 frames ('sparksee.io.pool.temporal.minsize' at SparkseeProperties).

Maximum size for the temporary pool: 0 frames ('sparksee.io.pool.temporal.maxsize' at SparkseeProperties).

Number of pools in the pool cluster: 0 pools ('sparksee.io.pool.clustersize' at SparkseeProperties). 0 or 1 means the clustering is disabled.

Maximum size for the cache (all pools): 0 MB ('sparksee.io.cache.maxsize' at SparkseeProperties).

License code: "" ('sparksee.license' at SparkseeProperties). No license code means evaluation license.

Log level: Info ('sparksee.log.level' at SparkseeProperties).

Log file: "sparksee.log" ('sparksee.log.file' at SparkseeProperties).

Cache statistics: false (disabled) ('sparksee.cache.statistics' at SparkseeProperties).

Cache statistics log file: "statistics.log" ('sparksee.cache.statisticsFile' at SparkseeProperties).

Cache statistics snapshot time: 1000 msecs [TimeUnit] ('sparksee.cache.statisticsSnapshotTime' at SparkseeProperties).

Recovery enabled: false ('sparksee.io.recovery' at SparkseeProperties).

Recovery log file: "" ('sparksee.io.recovery.logfile' at SparkseeProperties).

Recovery cache max size: 1MB ('sparksee.io.recovery.cachesize' at SparkseeProperties).

Recovery checkpoint time: 60 seconds [TimeUnit] ('sparksee.io.recovery.checkpointTime' at SparkseeProperties).

High-availability: false (disabled) ('sparksee.ha' at SparkseeProperties).

High-availability coordinators: "" ('sparksee.ha.coordinators' at SparkseeProperties).

High-availability IP: "" ('sparksee.ha.ip' at SparkseeProperties).

High-availability sync polling: 0 (disabled) [TimeUnit] ('sparksee.ha.sync' at SparkseeProperties).

High-availability master history: 1D (1 day) [TimeUnit] ('sparksee.ha.master.history' at SparkseeProperties).

Use of TimeUnit:

Those variables using TimeUnit allow for:

<X>[D|H|M|S|s|m|u]

where <X> is a number followed by an optional character which represents the unit: D for days, H for hours, M for minutes, S or s for seconds, m for milliseconds and u for microseconds. If no unit character is given, seconds are assumed.

Capture abort signals to dump the call stack ('sparksee.callstackdump' at SparkseeProperties) is enabled by default on most platforms.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.70.2 Constructor & Destructor Documentation

__init__() [1/3]

```
sparksee.SparkseeConfig.__init__ (
              self,
              path)
```

Creates a new instance with a specific config file.

The config file will be loaded using the global properties class and the file may be automatically overwritten with the config changes. If the file doesn't exists, it will be created.

Parameters

path	[in] File path to the config file.

__init__() [2/3]

```
sparksee.SparkseeConfig.__init__ (
              self,
              path,
              client_id,
              license_id)
```

Creates a new instance with a specific config file and IDs.

The config file will be loaded using the global properties class and the file may be automatically overwritten with the config changes. If the config file already exists, the client and license ids should have the same values as the given arguments. If the file doesn't exists, it will be created.

Parameters

path	[in] File path to the config file.
client_id	[in] The client identifier.
license_id	[in] The license identifier.

__init__() [3/3]

```
sparksee.SparkseeConfig.__init__ (
              self)
```

Creates a new instance.

Values are set with default values.

5.70.3 Member Function Documentation

download_license()

```
sparksee.SparkseeConfig.download_license (
              self)
```

Try to download a license key Can be used to explicitly download a license when the autodownload is disabled (LicensePreDownloadDays == -1).

Returns

Returns -1 if a valid license key couldn't be downloaded, 0 if the same license was downloaded or 1 if a different license is downloaded.

get_a_e_s_i_v()

```
sparksee.SparkseeConfig.get_a_e_s_i_v (
              self)
```

Get the AES initialization vector in a hexadecimal encoded string.

Returns

The AES initialization vector as an hexadecimal string.

get_a_e_s_key()

```
sparksee.SparkseeConfig.get_a_e_s_key (
              self)
```

Get the AES encryption key in a hexadecimal encoded string.

Returns

The AES encryption Key as an hexadecimal string.

get_cache_max_size()

```
sparksee.SparkseeConfig.get_cache_max_size (
              self)
```

Gets the maximum size for the cache (all pools) in MB.

Returns

The maximum size for the cache (all pools) in MB.

get_cache_statistics_enabled()

```
sparksee.SparkseeConfig.get_cache_statistics_enabled (
              self)
```

Gets whether cache statistics are enabled or disabled.

Returns

TRUE if cache statistics are enabled, FALSE otherwise.

get_cache_statistics_file()

```
sparksee.SparkseeConfig.get_cache_statistics_file (
              self)
```

Gets the cache statistics log file.

Useless if cache statistics are disabled.

Returns

The cache statistics log file.

get_cache_statistics_snapshot_time()

```
sparksee.SparkseeConfig.get_cache_statistics_snapshot_time (
              self)
```

Gets the cache statistics snapshot time in microseconds.

Useless if cache statistics are disabled.

Returns

The cache statistics snapshot time in microseconds.

get_call_stack_dump()

```
sparksee.SparkseeConfig.get_call_stack_dump (
              self)
```

Gets whether the signals will be captured to dump the call stack or not.

Returns

TRUE if the signals must be captured, FALSE otherwise.

get_checksum_enabled()

```
sparksee.SparkseeConfig.get_checksum_enabled (
              self)
```

Gets whether the storage checksum usage is enabled or disabled.

Returns

TRUE if the checksum is enabled, FALSE otherwise.

get_client_id()

```
sparksee.SparkseeConfig.get_client_id (
              self)
```

Gets the client identifier.

Returns

The client identifier.

get_download_status()

```
sparksee.SparkseeConfig.get_download_status (
              self)
```

Gets a message with the license download result.

It can be used when the DownloadLicense call failed.

Returns

The log download result.

get_encryption_enabled()

```
sparksee.SparkseeConfig.get_encryption_enabled (
              self)
```

Gets whether the storage encryption is enabled or disabled.

Returns

TRUE if the encryption is enabled, FALSE otherwise.

get_extent_pages()

```
sparksee.SparkseeConfig.get_extent_pages (
              self)
```

Gets the number of pages per extent.

Returns

The number of pages per extent.

get_extent_size()

```
sparksee.SparkseeConfig.get_extent_size (
              self)
```

Gets the size of a extent.

Returns

The size of a extent in KB.

get_high_availability_coordinators()

```
sparksee.SparkseeConfig.get_high_availability_coordinators (
              self)
```

Gets the coordinators address and port list.

Returns

The coordinators address and port list.

get_high_availability_enabled()

```
sparksee.SparkseeConfig.get_high_availability_enabled (
              self)
```

Gets whether high availability mode is enabled or disabled.

Returns

TRUE if high availability mode is enabled, FALSE otherwise.

get_high_availability_i_p()

```
sparksee.SparkseeConfig.get_high_availability_i_p (
              self)
```

Gets the IP address and port of the instance.

Returns

The IP address and port of the instance.

get_high_availability_master_history()

```
sparksee.SparkseeConfig.get_high_availability_master_history (
              self)
```

Gets the master's history log.

Returns

The master's history log.

get_high_availability_synchronization()

```
sparksee.SparkseeConfig.get_high_availability_synchronization (
              self)
```

Gets the synchronization polling time.

Returns

The Synchronization polling time.

get_in_mem_alloc_size()

```
sparksee.SparkseeConfig.get_in_mem_alloc_size (
              self)
```

Gets the in-memory allocator size.

Returns

Returns the in-memory allocator size

get_license()

```
sparksee.SparkseeConfig.get_license (
              self)
```

Gets the license key.

Returns

The license key.

get_license_id()

```
sparksee.SparkseeConfig.get_license_id (
              self)
```

Gets the license identifier.

Returns

The license identifier.

get_license_pre_download_days()

```
sparksee.SparkseeConfig.get_license_pre_download_days (
              self)
```

Get the number of days before expiration when a new license will be downloaded.

Returns

Returns the number of days or -1 if it will never be automatically downloaded.

get_log_file()

```
sparksee.SparkseeConfig.get_log_file (
              self)
```

Gets the log file.

Returns

The log file.

get_log_level()

```
sparksee.SparkseeConfig.get_log_level (
              self)
```

Gets the log level.

Returns

The LogLevel.

get_plugins_dir()

```
sparksee.SparkseeConfig.get_plugins_dir (
              self)
```

Gets the Plugins directory.

Returns

The pulgins directory.

get_pool_frame_size()

```
sparksee.SparkseeConfig.get_pool_frame_size (
              self)
```

Gets the size of a pool frame in number of extents.

Returns

The size of a pool frame in number of extents.

get_pool_partitions()

```
sparksee.SparkseeConfig.get_pool_partitions (
              self)
```

Gets the number of partitions in each PartitionedPool.

Returns

The number of partitions in each PartitionedPool.

get_pool_persistent_max_size()

```
sparksee.SparkseeConfig.get_pool_persistent_max_size (
              self)
```

Gets the maximum size for the persistent pool in number of frames.

Returns

The maximum size for the persistent pool in number of frames.

get_pool_persistent_min_size()

```
sparksee.SparkseeConfig.get_pool_persistent_min_size (
              self)
```

Gets the minimum size for the persistent pool in number of frames.

Returns

The minimum size for the persistent pool in number of frames.

get_pool_temporary_max_size()

```
sparksee.SparkseeConfig.get_pool_temporary_max_size (
              self)
```

Gets the maximum size for the temporary pool in number of frames.

Returns

The maximum size for the temporary pool in number of frames.

get_pool_temporary_min_size()

```
sparksee.SparkseeConfig.get_pool_temporary_min_size (
              self)
```

Gets the minimum size for the temporary pool in number of frames.

Returns

The minimum size for the temporary pool in number of frames.

get_recovery_cache_max_size()

```
sparksee.SparkseeConfig.get_recovery_cache_max_size (
              self)
```

Gets the maximum size for the recovery log cache in extents.

Returns

The maximum size for the recovery log cache in extents.

get_recovery_checkpoint_time()

```
sparksee.SparkseeConfig.get_recovery_checkpoint_time (
              self)
```

Gets the delay time (in microseconds) between automatic checkpoints.

Returns

The delay time (in microseconds) between automatic checkpoints.

get_recovery_enabled()

```
sparksee.SparkseeConfig.get_recovery_enabled (
              self)
```

Gets whether the recovery is enabled or disabled.

Returns

TRUE if the recovery is enabled, FALSE otherwise.

get_recovery_log_file()

```
sparksee.SparkseeConfig.get_recovery_log_file (
              self)
```

Gets the recovery log file.

Returns

The recovery log file.

get_rollback_enabled()

```
sparksee.SparkseeConfig.get_rollback_enabled (
              self)
```

Gets whether the rollback is enabled or disabled.

Returns

TRUE if the rollback is enabled, FALSE otherwise.

get_sparksee_config_file()

```
sparksee.SparkseeConfig.get_sparksee_config_file (
              self)
```

Gets the config file path.

Returns

Returns the configured path or an empty string

get_tmp_enabled()

```
sparksee.SparkseeConfig.get_tmp_enabled (
              self)
```

Gets whether using temporary storage for computations is enabled.

Returns

True if enabled

get_tmp_folder()

```
sparksee.SparkseeConfig.get_tmp_folder (
              self)
```

Gets the temporary folder used for temporary staging.

Returns

The temporary folder path

save()

```
sparksee.SparkseeConfig.save (
              self)
```

Save the current configuration in the specified config file.

It will try to save the current configuration only if a config file was specified. This method will be automatically used when a Sparksee class downloads a new License.

Returns

Returns true if the config file is successfully written or false otherwise.

save_all()

```
sparksee.SparkseeConfig.save_all (
              self)
```

Save all the current configuration settings in the specified config file.

It will try to save the current configuration only if a config file was specified. The saved config file WILL CONATAIN all the modified settings including the secret ENCRYPTION KEYS. You should usually just use the Save method instead.

Returns

Returns true if the config file is successfully written or false otherwise.

set_a_e_s_encryption_enabled() [1/2]

```
sparksee.SparkseeConfig.set_a_e_s_encryption_enabled (
              self,
              key_in_hex,
              iv_in_hex)
```

Enables storage encryption using AES with the given key and iv.

The key and initialization vector will not be saved in the config file.

Parameters

key_in_hex	[In] The AES encryption Key as an hexadecimal string (8, 16 or 32 bytes).
iv_in_hex	[In] The AES Initialization Vector as an hexadecimal string (16 bytes).

Returns

Returns true if the encryption had been enabled with the given arguments or false otherwise.

set_a_e_s_encryption_enabled() [2/2]

```
sparksee.SparkseeConfig.set_a_e_s_encryption_enabled (
              self,
              key_size)
```

Enables storage encryption using AES and GENARATES a key and an iv.

YOU MUST GET AND KEEP SAFE THE GENERATED KEY AND IV! The key and initialization vector will not be saved in the config file.

Parameters

key_size	[In] The key size in bytes (8, 16 or 32).

Returns

Returns true if the encryption had been enabled.

set_cache_max_size()

```
sparksee.SparkseeConfig.set_cache_max_size (
              self,
              mega_bytes)
```

Sets the maximum size for the cache (all pools) in MB.

Parameters

mega_bytes	[in] The maximum size for the cache (all pools) in MB. It must be non-negative.

set_cache_statistics_enabled()

```
sparksee.SparkseeConfig.set_cache_statistics_enabled (
              self,
              status)
```

Enables or disables cache statistics.

Parameters

status	[in] If TRUE this enables cache statistics, if FALSE this disables cache statistics.

set_cache_statistics_file()

```
sparksee.SparkseeConfig.set_cache_statistics_file (
              self,
              file_path)
```

Sets the cache statistics log file.

Useless if cache statistics are disabled.

Parameters

file_path	[in] The cache statistics log file.

set_cache_statistics_snapshot_time()

```
sparksee.SparkseeConfig.set_cache_statistics_snapshot_time (
              self,
              micro_seconds)
```

Sets the cache statistics snapshot time.

Useless if cache statistics are disabled.

Parameters

micro_seconds	[in] The cache statistics snapshot time in microseconds.

set_call_stack_dump()

```
sparksee.SparkseeConfig.set_call_stack_dump (
              self,
              status)
```

Sets whether the signals will be captured to dump the call stack or not.

Parameters

status	[in] If TRUE signals must be captured.

set_checksum_enabled()

```
sparksee.SparkseeConfig.set_checksum_enabled (
              self,
              status)
```

Enables or disables the storage checksum usage.

Parameters

status	[in] If TRUE this enables the checksum, if FALSE then disables it.

set_client_id()

```
sparksee.SparkseeConfig.set_client_id (
              self,
              client_id)
```

Set the client identifier.

If you don't have a client identifier, you may want to register at http://sparsity-technologies.com/↩#sparksee

Parameters

client↩_id	[in] The client identifier.

set_extent_pages()

```
sparksee.SparkseeConfig.set_extent_pages (
              self,
              pages)
```

Sets the number of pages per extent.

Parameters

pages	[in] The number of pages. It must be at least 1 page and the page size must be greater than or equal to 4KB.

set_extent_size()

```
sparksee.SparkseeConfig.set_extent_size (
              self,
              k_bytes)
```

Sets the size of the extents in KB.

Parameters

k_bytes	[in] The size of an extent in KB. An extent can have a size between 4KB and 64KB, and it must be a power of 2.

set_high_availability_coordinators()

```
sparksee.SparkseeConfig.set_high_availability_coordinators (
              self,
              ip)
```

Sets the coordinators address and port list.

Parameters

ip	[in] The coordinators address and port list.

set_high_availability_enabled()

```
sparksee.SparkseeConfig.set_high_availability_enabled (
              self,
              status)
```

Enables or disables high availability mode.

Parameters

status	[in] If TRUE this enables high availability mode, if FALSE this disables high availability mode.

set_high_availability_i_p()

```
sparksee.SparkseeConfig.set_high_availability_i_p (
              self,
              ip)
```

Sets the IP address and port of the instance.

Parameters

ip	[in] The IP address and port of the instance.

set_high_availability_master_history()

```
sparksee.SparkseeConfig.set_high_availability_master_history (
              self,
              file_path)
```

Sets the master's history log.

Parameters

file_path	[in] The master's history log.

set_high_availability_synchronization()

```
sparksee.SparkseeConfig.set_high_availability_synchronization (
              self,
              micro_seconds)
```

Sets the synchronization polling time.

Parameters

micro_seconds	[in] The synchronization polling time.

set_in_mem_alloc_size()

```
sparksee.SparkseeConfig.set_in_mem_alloc_size (
              self,
              size)
```

Sets the in-memory allocator size.

Parameters

size	The size to set the in-memory allocator to

set_license()

```
sparksee.SparkseeConfig.set_license (
              self,
              key)
```

Sets the license key.

Parameters

key	[in] The license key. Returns -1 if the new key can't be used, 0 if it's the same license or 1 if the new license is applied.

set_license_id()

```
sparksee.SparkseeConfig.set_license_id (
              self,
              license_id)
```

Set the license identifier.

If you don't have a license identifier, you may want to register at http://sparsity-technologies.com/↩#sparksee

Parameters

license↩_id	[in] The license identifier.

set_license_pre_download_days()

```
sparksee.SparkseeConfig.set_license_pre_download_days (
            self,
            days)
```

Start trying to automatically download a new license at the specified number of days before expiration.

Parameters

days	[in] Number of days before expiration or -1 if the license should never be downloaded.

set_log_file()

```
sparksee.SparkseeConfig.set_log_file (
            self,
            file_path)
```

Sets the log file.

Parameters

file_path	[in] The log file.

set_log_level()

```
sparksee.SparkseeConfig.set_log_level (
            self,
            level)
```

Sets the log level.

Parameters

level	[in] The LogLevel.

set_plugins_dir()

```
sparksee.SparkseeConfig.set_plugins_dir (
            self,
            dir_path)
```

Sets the Plugins directory.

Parameters

dir_path	[in] The plugins directory path.

set_pool_frame_size()

```
sparksee.SparkseeConfig.set_pool_frame_size (
              self,
              extents)
```

Sets the size of a pool frame in number of extents.

Parameters

extents	[in] The size of a pool frame in number of extents. It must be non-negative.

set_pool_partitions()

```
sparksee.SparkseeConfig.set_pool_partitions (
              self,
              pools)
```

Sets the number of pools in each PartitionedPool.

Parameters

pools	[in] The number of pools in each PartitionedPool. It must be non-negative.

set_pool_persistent_max_size()

```
sparksee.SparkseeConfig.set_pool_persistent_max_size (
              self,
              frames)
```

Sets the maximum size for the persistent pool in number of frames.

Parameters

frames	[in] The maximum size for the persistent pool in number of frames. It must be non-negative.

set_pool_persistent_min_size()

```
sparksee.SparkseeConfig.set_pool_persistent_min_size (
              self,
              frames)
```

Sets the minimum size for the persistent pool in number of frames.

Parameters

frames	[in] The minimum size for the persistent pool in number of frames. It must be non-negative.

set_pool_temporary_max_size()

```
sparksee.SparkseeConfig.set_pool_temporary_max_size (
              self,
              frames)
```

Sets the maximum size for the temporary pool in number of frames.

Parameters

frames	[in] The maximum size for the temporary pool in number of frames. It must be non-negative.

set_pool_temporary_min_size()

```
sparksee.SparkseeConfig.set_pool_temporary_min_size (
              self,
              frames)
```

Sets the minimum size for the temporary pool in number of frames.

Parameters

frames	[in] The minimum size for the temporary pool in number of frames. It must be non-negative.

set_recovery_cache_max_size()

```
sparksee.SparkseeConfig.set_recovery_cache_max_size (
              self,
              extents)
```

Sets the maximum size for the recovery log cache in extents.

Parameters

extents	[in] The maximum size for the recovery log cache in extents. A 0 sets the default value (extents up to 1MB).

set_recovery_checkpoint_time()

```
sparksee.SparkseeConfig.set_recovery_checkpoint_time (
              self,
              micro_seconds)
```

Sets the delay time (in microseconds) between automatic checkpoints.

Parameters

micro_seconds	[in] The delay time (in microseconds) between automatic checkpoints. A 0 forces a checkpoint after each commited transaction.

set_recovery_enabled()

```
sparksee.SparkseeConfig.set_recovery_enabled (
              self,
              status)
```

Enables or disables the recovery.

Parameters

status	[in] If TRUE this enables the recovery, if FALSE then disables it.

set_recovery_log_file()

```
sparksee.SparkseeConfig.set_recovery_log_file (
              self,
              file_path)
```

Sets the recovery log file.

Parameters

file_path	[in] The recovery log file. Left it empty for the default log file (same as <database_file_name>.log)

set_rollback_enabled()

```
sparksee.SparkseeConfig.set_rollback_enabled (
              self,
              status)
```

Enables or disables the rollback.

Parameters

status	[in] If TRUE this enables the rollback, if FALSE then disables it.

set_sparksee_config_file()

```
sparksee.SparkseeConfig.set_sparksee_config_file (
              self,
              path)
```

Sets the config file path.

The file is not loaded in this operation, see the constructor options if you need to load the file. The config file may be automatically overwritten with the config changes. If the file doesn't exists, it will be created.

Parameters

path	[in] File path to the config file.

set_tmp_enabled()

```
sparksee.SparkseeConfig.set_tmp_enabled (
              self,
              enabled)
```

Sets whether to use temporary storage for computations or not.

Parameters

enabled	True to use temporary storage for computation

set_tmp_folder()

```
sparksee.SparkseeConfig.set_tmp_folder (
              self,
              tmp_folder)
```

Sets the temporary folder used for temporary staging.

Parameters

tmp_folder	The temporary folder to set

5.71 sparksee.SparkseeProperties Class Reference

Sparksee properties file.

Public Member Functions

- **clear** (self)

  Remove all the properties.
- get (self, key, def)

  Gets a property.
- get_integer (self, key, def)

  Gets a property as an integer.
- set_h_i (self, value)

  YOU SHOULD NOT USE THIS METHOD.
- load (self, path)

  Loads properties from the given file path.
- get_time_unit (self, key, def)

  Gets a property as a time unit.
- get_boolean (self, key, def)

  Gets a property as a boolean.

5.71.1 Detailed Description

Sparksee properties file.

This class is implemented as a singleton, so all public methods are static.

It allows for getting the property values stored in a properties file. A properties file is a file where there is one line per property. A property is defined by a key and a value as follows: key=value

By default, this loads properties from the file './sparksee.cfg'. The user may choose to load a different file by calling the method Load().

If the default properties file or the one loaded by the user do not exist, then this behaves as loading an empty properties file.

5.71.2 Member Function Documentation

get()

```
sparksee.SparkseeProperties.get (
              self,
              key,
              def)
```

Gets a property.

Parameters

key	[in] The name of the property to lookup.
def	[in] Default value to be returned in case there is no property with the name key.

Returns

The value of the property, or def if the key is not found.

get_boolean()

```
sparksee.SparkseeProperties.get_boolean (
              self,
              key,
              def)
```

Gets a property as a boolean.

Parameters

key	[in] The name of the property to lookup.
def	[in] Default value to be returned in case there is no property with the name key.

Returns

The property value, or def if the key is not found or in case of error.

get_integer()

```
sparksee.SparkseeProperties.get_integer (
              self,
              key,
              def)
```

Gets a property as an integer.

Parameters

key	[in] The name of the property to lookup.
def	[in] Default value to be returned in case there is no property with the name key.

Returns

The property value, or def if the key is not found or in case of error.

get_time_unit()

```
sparksee.SparkseeProperties.get_time_unit (
              self,
              key,
              def)
```

Gets a property as a time unit.

A time unit is a string representation of a time duration with a time unit such as '10s' or '3H'.

Valid format for the string representation: Blanks at the begining or at the end are ignored. No blanks are allowed between the time duration and the unit time.

Allowed time units: 'D' for days, 'H' for hours, 'M' for minutes, 'S' o 's' for seconds, 'm' for milliseconds and 'u' for microseconds.

There is a special case: If no time unit is given, seconds is the default. So, '10' means 10 seconds.

Parameters

key	[in] The name of the property to lookup.
def	[in] The default value (in microseconds) to be returned in case there is no property with the name key.

Returns

The time duration in microseconds, or def if the key is not found or in case of error.

load()

```
sparksee.SparkseeProperties.load (
              self,
              path)
```

Loads properties from the given file path.

Parameters

path	[in] File path to load properties from.

set_h_i()

```
sparksee.SparkseeProperties.set_h_i (
              self,
              value)
```

YOU SHOULD NOT USE THIS METHOD.

This method exists only for a specific service.

Parameters

value	null

5.72 sparksee.StringList Class Reference

String list.

Public Member Functions

- **clear** (self)

  Clears the list.
- __init__ (self)

  Constructor.
- __iter__ (self)

  Gets a new TypeListIterator.
- iterator (self)

  Gets a new StringListIterator.
- add (self, str)

  Adds a String at the end of the list.
- count (self)

  Number of elements in the list.

5.72.1 Detailed Description

String list.

It stores a String (unicode) list.

Use StringListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.72.2 Constructor & Destructor Documentation

__init__()

```
sparksee.StringList.__init__ (
              self)
```

Constructor.

This creates an empty list.

5.72.3 Member Function Documentation

__iter__()

```
sparksee.StringList.__iter__ (
              self)
```

Gets a new TypeListIterator.

Returns

TypeListIterator instance

add()

```
sparksee.StringList.add (
              self,
              str)
```

Adds a String at the end of the list.

Parameters

str	[in] String.

count()

```
sparksee.StringList.count (
              self)
```

Number of elements in the list.

Returns

Number of elements in the list.

iterator()

```
sparksee.StringList.iterator (
              self)
```

Gets a new StringListIterator.

Returns

StringListIterator instance.

5.73 sparksee.StringListIterator Class Reference

StringList iterator class.

Public Member Functions

- next (self)
  Moves to the next element.
- has_next (self)
  Gets if there are more elements.
- __next__ (self)
  Return the next item from the container.

5.73.1 Detailed Description

StringList iterator class.

Iterator to traverse all the strings into a StringList instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.73.2 Member Function Documentation

__next__()

```
sparksee.StringListIterator.__next__ (
              self)
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

has_next()

```
sparksee.StringListIterator.has_next (
              self)
```

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

next()

```
sparksee.StringListIterator.next (
              self)
```

Moves to the next element.

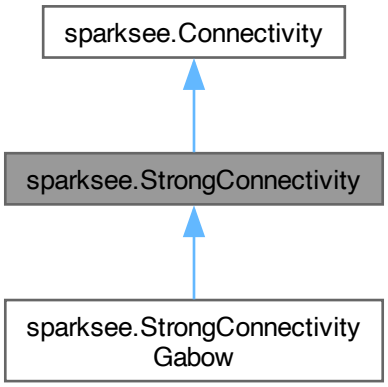
Returns

The next element.

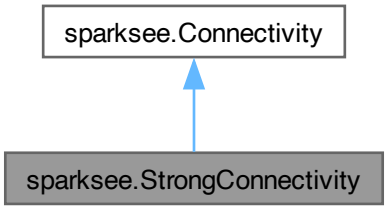
5.74 sparksee.StrongConnectivity Class Reference

StrongConnectivity class.

Inheritance diagram for sparksee.StrongConnectivity:



Collaboration diagram for sparksee.StrongConnectivity:



Public Member Functions

- add_edge_type (self, type, dir)

  Allows connectivity through edges of the given type.
- run (self)

  Runs the algorithm in order to find the connected components.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- get_connected_components (self)

  Returns the results generated by the execution of the algorithm.
- add_all_edge_types (self, dir)

Allows connectivity through all edge types of the graph.

- exclude_edges (self, edges)

  Set which edges can't be used.
- add_node_type (self, t)

  Allows connectivity through nodes of the given type.
- set_materialized_attribute (self, attribute_name)

  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.
- **add_all_node_types** (self)

  Allows connectivity through all node types of the graph.
- close (self)

  Closes the Connectivity instance.
- is_closed (self)

  Gets if Connectivity has been closed or not.

5.74.1 Detailed Description

StrongConnectivity class.

Any class implementing this abstract class can be used to solve the problem of finding strongly connected components in a directed graph.

It consists in finding components where every pair (u,v) of nodes contained in it has a path from u to v using the specified direction for each edge type.

It is possible to set some restrictions after constructing a new instance of this class and before running it in order to limit the results.

After the execution, we can retrieve the results stored in an instance of the ConnectedComponents class using the GetConnectedComponents method.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.74.2 Member Function Documentation

add_all_edge_types()

```
sparksee.StrongConnectivity.add_all_edge_types (
              self,
              dir)
```

Allows connectivity through all edge types of the graph.

Parameters

dir	[in] Edge direction.

Reimplemented in sparksee.StrongConnectivityGabow.

add_edge_type()

```
sparksee.StrongConnectivity.add_edge_type (
              self,
              type,
              dir)
```

Allows connectivity through edges of the given type.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Reimplemented in sparksee.StrongConnectivityGabow.

add_node_type()

```
sparksee.StrongConnectivity.add_node_type (
              self,
              t)
```

Allows connectivity through nodes of the given type.

Parameters

t	null

Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.StrongConnectivityGabow.

close()

```
sparksee.Connectivity.close (
              self)  [inherited]
```

Closes the Connectivity instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.StrongConnectivity.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.StrongConnectivityGabow.

exclude_nodes()

```
sparksee.StrongConnectivity.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.StrongConnectivityGabow.

get_connected_components()

```
sparksee.StrongConnectivity.get_connected_components (
              self)
```

Returns the results generated by the execution of the algorithm.

These results contain information related to the connected components found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class ConnectedComponents which contain information related to the connected components found.

Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.StrongConnectivityGabow.

is_closed()

```
sparksee.Connectivity.is_closed (
              self)  [inherited]
```

Gets if Connectivity has been closed or not.

See also

close()

Returns

TRUE if the Connectivity instance has been closed, FALSE otherwise.

run()

```
sparksee.StrongConnectivity.run (
              self)
```

Runs the algorithm in order to find the connected components.

This method can be called only once.

Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.StrongConnectivityGabow.

set_materialized_attribute()

```
sparksee.StrongConnectivity.set_materialized_attribute (
              self,
              attribute_name)
```

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class ConnectedComponents indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the connected components found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

attribute_name	[in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm.

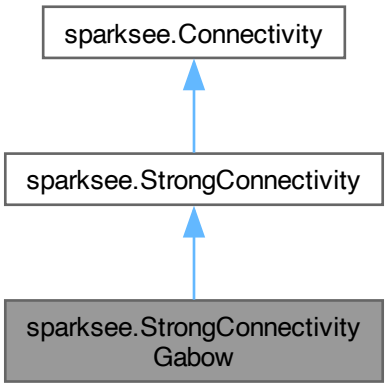
Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.StrongConnectivityGabow.

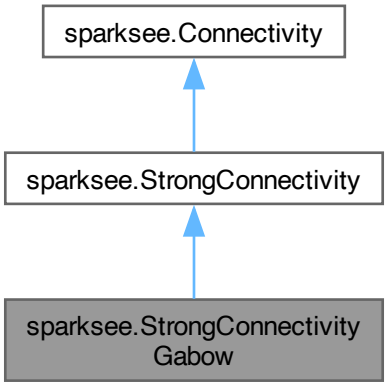
5.75 sparksee.StrongConnectivityGabow Class Reference

This class can be used to solve the problem of finding strongly connected components in a directed graph.

Inheritance diagram for sparksee.StrongConnectivityGabow:



Collaboration diagram for sparksee.StrongConnectivityGabow:



Public Member Functions

- add_edge_type (self, type, dir)

  Allows connectivity through edges of the given type.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.

- get_connected_components (self)

  Returns the results generated by the execution of the algorithm.
- **run** (self)

  Executes the algorithm.
- add_all_edge_types (self, dir)

  Allows connectivity through all edge types of the graph.
- exclude_edges (self, edges)

  Set which edges can't be used.
- add_node_type (self, t)

  Allows connectivity through nodes of the given type.
- set_materialized_attribute (self, attribute_name)

  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.
- __init__ (self, session)

  Creates a new instance of StrongConnectivityGabow.
- **add_all_node_types** (self)

  Allows connectivity through all node types of the graph.
- close (self)

  Closes the Connectivity instance.
- is_closed (self)

  Gets if Connectivity has been closed or not.

5.75.1 Detailed Description

This class can be used to solve the problem of finding strongly connected components in a directed graph.

It consists in finding components where every pair (u,v) of nodes contained in it has a path from u to v using the specified direction for each edge type. This implementation is based on the Gabow algorithm.

It is possible to set some restrictions after constructing a new instance of this class and before running it in order to limit the results.

After the execution, we can retrieve the results stored in an instance of the ConnectedComponents class using the GetConnectedComponents method.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.75.2 Constructor & Destructor Documentation

__init__()

```
sparksee.StrongConnectivityGabow.__init__ (
              self,
              session)
```

Creates a new instance of StrongConnectivityGabow.

After creating this instance is required to indicate the set of edge types and the set of node types which will be navigated through while traversing the graph in order to find the strong connected components.

Parameters

session	[in] Session to get the graph from and calculate the connectivity

5.75.3 Member Function Documentation

add_all_edge_types()

```
sparksee.StrongConnectivityGabow.add_all_edge_types (
              self,
              dir)
```

Allows connectivity through all edge types of the graph.

Parameters

dir	[in] Edge direction.

Reimplemented from sparksee.StrongConnectivity.

add_edge_type()

```
sparksee.StrongConnectivityGabow.add_edge_type (
              self,
              type,
              dir)
```

Allows connectivity through edges of the given type.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Reimplemented from sparksee.StrongConnectivity.

add_node_type()

```
sparksee.StrongConnectivityGabow.add_node_type (
              self,
              t)
```

Allows connectivity through nodes of the given type.

Parameters

t	null

Reimplemented from sparksee.StrongConnectivity.

close()

```
sparksee.Connectivity.close (
              self)  [inherited]
```

Closes the Connectivity instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.StrongConnectivityGabow.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.StrongConnectivity.

exclude_nodes()

```
sparksee.StrongConnectivityGabow.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.StrongConnectivity.

get_connected_components()

```
sparksee.StrongConnectivityGabow.get_connected_components (
              self)
```

Returns the results generated by the execution of the algorithm.

These results contain information related to the connected components found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class ConnectedComponents which contain information related to the connected components found.

Reimplemented from sparksee.StrongConnectivity.

is_closed()

```
sparksee.Connectivity.is_closed (
              self)  [inherited]
```

Gets if Connectivity has been closed or not.

See also

close()

Returns

TRUE if the Connectivity instance has been closed, FALSE otherwise.

set_materialized_attribute()

```
sparksee.StrongConnectivityGabow.set_materialized_attribute (
              self,
              attribute_name)
```

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class ConnectedComponents indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the connected components found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

attribute_name	[in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm.

Reimplemented from sparksee.StrongConnectivity.

5.76 sparksee.TextStream Class Reference

TextStream class.

Public Member Functions

- write (self, data_i_n, length)
  Write data to the stream.
- close (self)
  Closes the stream.
- read (self, length)
  Read data from the stream.
- is_null (self)
  Returns TRUE if the stream is not available.
- __init__ (self, append)
  Creates a new instance.

5.76.1 Detailed Description

TextStream class.

It allows for reading and writting Text attribute values.

It is very important to close the stream once no more reading or writting operations will be performed to ensure data is successfully stored.

Whereas string attributes are set and got using the Value class, text attributes are operated using a stream pattern.

Use of TextStream for writing: (i) Create a TextStream instance and (ii) set the stream for a text attribute of a node or edge instance with the graph SetAttributeText method. Once the set attribute text has been done, (iii) perform as many write operations as you need to the TextStream instance. Lastly, (iv) exeucte Close to flush and close the stream.

Use of TextStream for reading: (i) Get the stream of a text attribute of a node or edge instance with the GetAttributeText graph method. Once you have the TextStream instance, (ii) you can execute Read operations to read from the stream. (iii) The end of the stream is reached when Read returns 0. Finally, (iv) execute Close to close stream resources.

Check out the 'Attributes and values' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.76.2 Constructor & Destructor Documentation

__init__()

```
sparksee.TextStream.__init__ (
              self,
              append)
```

Creates a new instance.

A TextStream only can be created by the user to write data.

Parameters

append	[in] If TRUE, the it is created in append mode to write from the end of the stream, otherwise it is created to write from the begining of the stream.

5.76.3 Member Function Documentation

close()

```
sparksee.TextStream.close (
              self)
```

Closes the stream.

Once the Stream is closed, it cannot be used again.

Closing the stream is mandatory when the stream is not null and strongly recommended when it's null to avoid deallocation problems in some platforms.

is_null()

```
sparksee.TextStream.is_null (
              self)
```

Returns TRUE if the stream is not available.

It returns for reading or writing data.

Returns

FALSE if the stream is ready

read()

```
sparksee.TextStream.read (
              self,
              length)
```

Read data from the stream.

Parameters

length	[in] Length of the given data buffer. It must be $>$ 0.

Returns

The read data.

write()

```
sparksee.TextStream.write (
              self,
              data_i_n,
              length)
```

Write data to the stream.

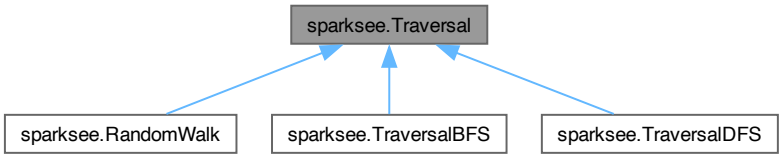
Parameters

data_i↩ _n	The string to write data from.
length	Number of characters to write. It must be $>$ 0.

5.77 sparksee.Traversal Class Reference

Traversal class.

Inheritance diagram for sparksee.Traversal:



Public Member Functions

- exclude_nodes (self, nodes)
  Set which nodes can't be used.
- has_next (self)
  Gets if there are more objects to be traversed.
- __iter__ (self)
  Gets a new TraversalIterator.
- add_all_edge_types (self, dir)
  Allows for traversing all edge types of the graph.
- exclude_edges (self, edges)
  Set which edges can't be used.
- __next__ (self)
  Return the next item from the container.
- add_edge_type (self, type, dir)
  Allows for traversing edges of the given type.
- get_current_depth (self)
  Returns the depth of the current node.
- add_node_type (self, type)
  Allows for traversing nodes of the given type.
- next (self)
  Gets the next object of the traversal.
- close (self)
  Closes the Traversal instance.
- **add_all_node_types** (self)
  Allows for traversing all node types of the graph.
- set_maximum_hops (self, maxhops)
  Sets the maximum hops restriction.
- is_closed (self)
  Gets if Traversal has been closed or not.

5.77.1 Detailed Description

Traversal class.

Any class implementing this abstract class can be used to traverse a graph.

Once the instance has been created and the allowed node and edge types has been set, it can be used as an iterator, retrieving the next object identifier of the traversal until there are no more.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.77.2 Member Function Documentation

__iter__()

```
sparksee.Traversal.__iter__ (
              self)
```

Gets a new TraversalIterator.

Returns

TraversalIterator instance

__next__()

```
sparksee.Traversal.__next__ (
              self)
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

add_all_edge_types()

```
sparksee.Traversal.add_all_edge_types (
              self,
              dir)
```

Allows for traversing all edge types of the graph.

Parameters

dir	[in] Edge direction.

Reimplemented in sparksee.RandomWalk, sparksee.TraversalBFS, and sparksee.TraversalDFS.

add_edge_type()

```
sparksee.Traversal.add_edge_type (
              self,
              type,
              dir)
```

Allows for traversing edges of the given type.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Reimplemented in sparksee.RandomWalk, sparksee.TraversalBFS, and sparksee.TraversalDFS.

add_node_type()

```
sparksee.Traversal.add_node_type (
              self,
              type)
```

Allows for traversing nodes of the given type.

Parameters

type	The node type to add

Reimplemented in sparksee.RandomWalk, sparksee.TraversalBFS, and sparksee.TraversalDFS.

close()

```
sparksee.Traversal.close (
              self)
```

Closes the Traversal instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.Traversal.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented in sparksee.RandomWalk, sparksee.TraversalBFS, and sparksee.TraversalDFS.

exclude_nodes()

```
sparksee.Traversal.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented in sparksee.RandomWalk, sparksee.TraversalBFS, and sparksee.TraversalDFS.

get_current_depth()

```
sparksee.Traversal.get_current_depth (
              self)
```

Returns the depth of the current node.

That is, it returns the depth of the node returned in the last call to Next().

Returns

The depth of the current node.

Reimplemented in sparksee.RandomWalk, sparksee.TraversalBFS, and sparksee.TraversalDFS.

has_next()

```
sparksee.Traversal.has_next (
              self)
```

Gets if there are more objects to be traversed.

Returns

TRUE if there are more objects, FALSE otherwise.

Reimplemented in sparksee.RandomWalk, sparksee.TraversalBFS, and sparksee.TraversalDFS.

is_closed()

```
sparksee.Traversal.is_closed (
              self)
```

Gets if Traversal has been closed or not.

See also

close()

Returns

TRUE if the Traversal instance has been closed, FALSE otherwise.

next()

```
sparksee.Traversal.next (
              self)
```

Gets the next object of the traversal.

Returns

A node or edge identifier.

Reimplemented in sparksee.RandomWalk, sparksee.TraversalBFS, and sparksee.TraversalDFS.

set_maximum_hops()

```
sparksee.Traversal.set_maximum_hops (
              self,
              maxhops)
```

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

Parameters

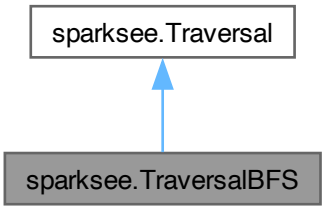
maxhops	[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.

Reimplemented in sparksee.RandomWalk, sparksee.TraversalBFS, and sparksee.TraversalDFS.

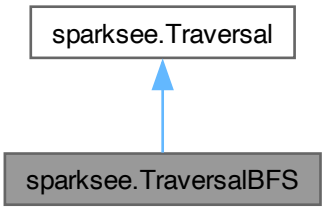
5.78 sparksee.TraversalBFS Class Reference

Breadth-First Search implementation of Traversal.

Inheritance diagram for sparksee.TraversalBFS:



Collaboration diagram for sparksee.TraversalBFS:



Public Member Functions

- add_edge_type (self, type, dir)

  Allows for traversing edges of the given type.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- add_all_edge_types (self, dir)

  Allows for traversing all edge types of the graph.
- add_node_type (self, type)

  Allows for traversing nodes of the given type.
- exclude_edges (self, edges)

  Set which edges can't be used.
- next (self)

  Gets the next object of the traversal.
- has_next (self)

  Gets if there are more objects to be traversed.
- get_current_depth (self)

  Returns the depth of the current node.
- __init__ (self, session, node)

  Creates a new instance.
- **add_all_node_types** (self)

  Allows for traversing all node types of the graph.
- set_maximum_hops (self, maxhops)

  Sets the maximum hops restriction.
- __iter__ (self)

  Gets a new TraversalIterator.
- __next__ (self)

  Return the next item from the container.
- close (self)

  Closes the Traversal instance.
- is_closed (self)

  Gets if Traversal has been closed or not.

5.78.1 Detailed Description

Breadth-First Search implementation of Traversal.

Starting from a source node, it visits all its neighbors at distance 1, then all its neighbors at distance 2, and so on.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.78.2 Constructor & Destructor Documentation

__init__()

```
sparksee.TraversalBFS.__init__ (
              self,
              session,
              node)
```

Creates a new instance.

Parameters

session	[in] Session to get the graph from and perform traversal.
node	[in] Node to start traversal from.

5.78.3 Member Function Documentation

__iter__()

```
sparksee.Traversal.__iter__ (
              self)  [inherited]
```

Gets a new TraversalIterator.

Returns

TraversalIterator instance

__next__()

```
sparksee.Traversal.__next__ (
              self)  [inherited]
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

add_all_edge_types()

```
sparksee.TraversalBFS.add_all_edge_types (
              self,
              dir)
```

Allows for traversing all edge types of the graph.

Parameters

dir	[in] Edge direction.

Reimplemented from sparksee.Traversal.

add_edge_type()

```
sparksee.TraversalBFS.add_edge_type (
              self,
              type,
              dir)
```

Allows for traversing edges of the given type.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Reimplemented from sparksee.Traversal.

add_node_type()

```
sparksee.TraversalBFS.add_node_type (
              self,
              type)
```

Allows for traversing nodes of the given type.

Parameters

type	The node type to add

Reimplemented from sparksee.Traversal.

close()

```
sparksee.Traversal.close (
              self)  [inherited]
```

Closes the Traversal instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.TraversalBFS.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.Traversal.

exclude_nodes()

```
sparksee.TraversalBFS.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.Traversal.

get_current_depth()

```
sparksee.TraversalBFS.get_current_depth (
              self)
```

Returns the depth of the current node.

That is, it returns the depth of the node returned in the last call to Next().

Returns

The depth of the current node.

Reimplemented from sparksee.Traversal.

has_next()

```
sparksee.TraversalBFS.has_next (
              self)
```

Gets if there are more objects to be traversed.

Returns

TRUE if there are more objects, FALSE otherwise.

Reimplemented from sparksee.Traversal.

is_closed()

```
sparksee.Traversal.is_closed (
              self)  [inherited]
```

Gets if Traversal has been closed or not.

See also

close()

Returns

TRUE if the Traversal instance has been closed, FALSE otherwise.

next()

```
sparksee.TraversalBFS.next (
              self)
```

Gets the next object of the traversal.

Returns

A node or edge identifier.

Reimplemented from sparksee.Traversal.

set_maximum_hops()

```
sparksee.TraversalBFS.set_maximum_hops (
              self,
              maxhops)
```

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

Parameters

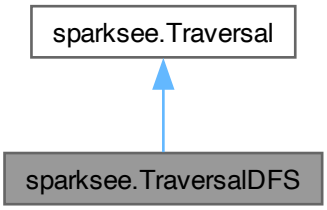
maxhops	[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.

Reimplemented from sparksee.Traversal.

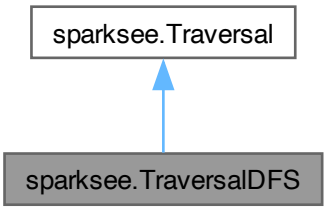
5.79 sparksee.TraversalDFS Class Reference

Depth-First Search (DFS) implementation of Traversal.

Inheritance diagram for sparksee.TraversalDFS:



Collaboration diagram for sparksee.TraversalDFS:



Public Member Functions

- add_edge_type (self, type, dir)

  Allows for traversing edges of the given type.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- add_all_edge_types (self, dir)

  Allows for traversing all edge types of the graph.
- add_node_type (self, type)

  Allows for traversing nodes of the given type.
- exclude_edges (self, edges)

  Set which edges can't be used.
- next (self)

  Gets the next object of the traversal.
- has_next (self)

  Gets if there are more objects to be traversed.
- get_current_depth (self)

  Returns the depth of the current node.
- __init__ (self, session, node)

  Creates a new instance.
- **add_all_node_types** (self)

  Allows for traversing all node types of the graph.
- set_maximum_hops (self, maxhops)

  Sets the maximum hops restriction.
- __iter__ (self)

  Gets a new TraversalIterator.
- __next__ (self)

  Return the next item from the container.
- close (self)

  Closes the Traversal instance.
- is_closed (self)

  Gets if Traversal has been closed or not.

5.79.1 Detailed Description

Depth-First Search (DFS) implementation of Traversal.

Starting from a source or root node, it visits as far as possible along each branch before backtracking.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.79.2 Constructor & Destructor Documentation

__init__()

```
sparksee.TraversalDFS.__init__ (
              self,
              session,
              node)
```

Creates a new instance.

Parameters

session	[in] Session to get the graph from and perform traversal.
node	[in] Node to start traversal from.

5.79.3 Member Function Documentation

__iter__()

```
sparksee.Traversal.__iter__ (
              self)  [inherited]
```

Gets a new TraversalIterator.

Returns

TraversalIterator instance

__next__()

```
sparksee.Traversal.__next__ (
              self)  [inherited]
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

add_all_edge_types()

```
sparksee.TraversalDFS.add_all_edge_types (
              self,
              dir)
```

Allows for traversing all edge types of the graph.

Parameters

dir	[in] Edge direction.

Reimplemented from sparksee.Traversal.

add_edge_type()

```
sparksee.TraversalDFS.add_edge_type (
              self,
              type,
              dir)
```

Allows for traversing edges of the given type.

Parameters

type	[in] Edge type.
dir	[in] Edge direction.

Reimplemented from sparksee.Traversal.

add_node_type()

```
sparksee.TraversalDFS.add_node_type (
              self,
              type)
```

Allows for traversing nodes of the given type.

Parameters

type	The node type to add

Reimplemented from sparksee.Traversal.

close()

```
sparksee.Traversal.close (
              self)  [inherited]
```

Closes the Traversal instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.TraversalDFS.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.Traversal.

exclude_nodes()

```
sparksee.TraversalDFS.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.Traversal.

get_current_depth()

```
sparksee.TraversalDFS.get_current_depth (
              self)
```

Returns the depth of the current node.

That is, it returns the depth of the node returned in the last call to Next().

Returns

The depth of the current node.

Reimplemented from sparksee.Traversal.

has_next()

```
sparksee.TraversalDFS.has_next (
              self)
```

Gets if there are more objects to be traversed.

Returns

TRUE if there are more objects, FALSE otherwise.

Reimplemented from sparksee.Traversal.

is_closed()

```
sparksee.Traversal.is_closed (
              self)  [inherited]
```

Gets if Traversal has been closed or not.

See also

close()

Returns

TRUE if the Traversal instance has been closed, FALSE otherwise.

next()

```
sparksee.TraversalDFS.next (
              self)
```

Gets the next object of the traversal.

Returns

A node or edge identifier.

Reimplemented from sparksee.Traversal.

set_maximum_hops()

```
sparksee.TraversalDFS.set_maximum_hops (
              self,
              maxhops)
```

Sets the maximum hops restriction.

All paths longer than the maximum hops restriction will be ignored.

Parameters

maxhops	[in] The maximum hops restriction. It must be positive or zero. Zero, the default value, means unlimited.

Reimplemented from sparksee.Traversal.

5.80 sparksee.Type Class Reference

Type data class.

Public Member Functions

- get_object_type (self)
  Gets the object type.
- get_id (self)
  Gets the Sparksee type identifier.
- get_restricted_from (self)
  Gets the tail or source type identifier for restricted edge types.
- get_num_objects (self)
  Gets the number of objects belonging to the type.
- get_are_neighbors_indexed (self)
  Gets if this is an edge type with neighbors index.
- get_restricted_to (self)
  Gets the head or target type identifier for restricted edge types.
- get_is_restricted (self)
  Gets if this is a restricted edge type.
- get_name (self)
  Gets the unique type name.
- get_is_directed (self)
  Gets if this is a directed edge type.

Static Public Attributes

- int **INVALID_TYPE** = 0
  Invalid type identifier.
- int **NODES_TYPE** = 2
  Identifier for all nodeType attributes.
- int **GLOBAL_TYPE** = 1
  Global type identifier.
- int **EDGES_TYPE** = 3
  Identifier for all edgeType attributes.

5.80.1 Detailed Description

Type data class.

It contains information about a node or edge type.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.80.2 Member Function Documentation

get_are_neighbors_indexed()

```
sparksee.Type.get_are_neighbors_indexed (
              self)
```

Gets if this is an edge type with neighbors index.

Returns

TRUE for edges types with neighbors index, FALSE otherwise.

get_id()

```
sparksee.Type.get_id (
              self)
```

Gets the Sparksee type identifier.

Returns

The Sparksee type identifier.

get_is_directed()

```
sparksee.Type.get_is_directed (
              self)
```

Gets if this is a directed edge type.

Returns

TRUE for directed edge types, FALSE otherwise.

get_is_restricted()

```
sparksee.Type.get_is_restricted (
              self)
```

Gets if this is a restricted edge type.

Returns

TRUE for restricted edge types, FALSE otherwise.

get_name()

```
sparksee.Type.get_name (
              self)
```

Gets the unique type name.

Returns

The unique type name.

get_num_objects()

```
sparksee.Type.get_num_objects (
              self)
```

Gets the number of objects belonging to the type.

Returns

The number of objects belonging to the type.

get_object_type()

```
sparksee.Type.get_object_type (
              self)
```

Gets the object type.

Returns

The object type.

get_restricted_from()

```
sparksee.Type.get_restricted_from (
              self)
```

Gets the tail or source type identifier for restricted edge types.

Returns

For restricted edge types, the tail or source type identifier, the Type InvalidType otherwise.

get_restricted_to()

```
sparksee.Type.get_restricted_to (
              self)
```

Gets the head or target type identifier for restricted edge types.

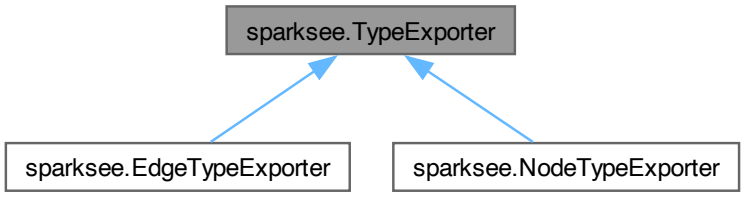
Returns

For restricted edge types, the head or target type identifier, the Type InvalidType otherwise.

5.81 sparksee.TypeExporter Class Reference

Base TypeExporter class.

Inheritance diagram for sparksee.TypeExporter:



Public Member Functions

- run (self)
  Runs export process.
- set_type (self, type)
  Sets the type to be exported.
- set_header (self, header)
  Sets the presence of a header row.
- set_row_writer (self, rw)
  Sets the output data destination.
- set_graph (self, graph)
  Sets the graph that will be exported.
- register (self, tel)
  Registers a new listener.
- set_attributes (self, attrs)
  Sets the list of Attributes.
- set_frequency (self, freq)
  Sets the frequency of listener notification.

5.81.1 Detailed Description

Base TypeExporter class.

Base class to export a node or edge type from a graph using a RowWriter.

TypeExporterListener can be registered to receive information about the progress of the export process by means of TypeExporterEvent. The default frequency of notification to listeners is 100000.

By default no header row is created.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.81.2 Member Function Documentation

register()

```
sparksee.TypeExporter.register (
              self,
              tel)
```

Registers a new listener.

Parameters

tel	[in] TypeExporterListener to be registered.

Reimplemented in sparksee.EdgeTypeExporter, and sparksee.NodeTypeExporter.

run()

```
sparksee.TypeExporter.run (
              self)
```

Runs export process.

WrongArgumentErrorIn case of bad settings ErrorFor other issues

Exceptions

RuntimeError	null
ValueError	null
IOError	If bad things happen writting to the RowWriter.

Reimplemented in sparksee.EdgeTypeExporter, and sparksee.NodeTypeExporter.

set_attributes()

```
sparksee.TypeExporter.set_attributes (
              self,
              attrs)
```

Sets the list of Attributes.

Parameters

attrs	[in] Attribute identifiers to be exported

Reimplemented in sparksee.EdgeTypeExporter, and sparksee.NodeTypeExporter.

set_frequency()

```
sparksee.TypeExporter.set_frequency (
              self,
              freq)
```

Sets the frequency of listener notification.

Parameters

freq	[in] Frequency in number of rows managed to notify progress to all listeners

Reimplemented in sparksee.EdgeTypeExporter, and sparksee.NodeTypeExporter.

set_graph()

```
sparksee.TypeExporter.set_graph (
              self,
              graph)
```

Sets the graph that will be exported.

Parameters

graph	[in] Graph.

Reimplemented in sparksee.EdgeTypeExporter, and sparksee.NodeTypeExporter.

set_header()

```
sparksee.TypeExporter.set_header (
              self,
              header)
```

Sets the presence of a header row.

Parameters

header	[in] If TRUE, a header row is dumped with the name of the attributes.

Reimplemented in sparksee.EdgeTypeExporter, and sparksee.NodeTypeExporter.

set_row_writer()

```
sparksee.TypeExporter.set_row_writer (
              self,
              rw)
```

Sets the output data destination.

Parameters

rw	[in] Input RowWriter.

Reimplemented in sparksee.EdgeTypeExporter, and sparksee.NodeTypeExporter.

set_type()

```
sparksee.TypeExporter.set_type (
              self,
              type)
```

Sets the type to be exported.

Parameters

type	[in] Type identifier.

Reimplemented in sparksee.EdgeTypeExporter, and sparksee.NodeTypeExporter.

5.82 sparksee.TypeExporterEvent Class Reference

Provides information about the progress of an TypeExproter instance.

Public Member Functions

- get_type_id (self)
  Gets the type identifier.
- get_total (self)
  Gets the total number of objects exported.
- get_count (self)
  Gets the current number of objects exported.
- is_last (self)
  Gets if this is the last event or not.

5.82.1 Detailed Description

Provides information about the progress of an TypeExproter instance.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.82.2 Member Function Documentation

get_count()

```
sparksee.TypeExporterEvent.get_count (
              self)
```

Gets the current number of objects exported.

Returns

The current number of objects exported.

get_total()

```
sparksee.TypeExporterEvent.get_total (
              self)
```

Gets the total number of objects exported.

Returns

The total number of objects exported.

get_type_id()

```
sparksee.TypeExporterEvent.get_type_id (
              self)
```

Gets the type identifier.

Returns

The type identifier.

is_last()

```
sparksee.TypeExporterEvent.is_last (
              self)
```

Gets if this is the last event or not.

Returns

TRUE if this is the last event, FALSE otherwise.

5.83 sparksee.TypeExporterListener Class Reference

Interface to be implemented to receive TypeExporterEvent events from a TypeExporter.

Public Member Functions

- notify_event (self, tee)
  Method to be notified from a TypeExporter.

5.83.1 Detailed Description

Interface to be implemented to receive TypeExporterEvent events from a TypeExporter.

Check out the 'Data export' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.83.2 Member Function Documentation

notify_event()

```
sparksee.TypeExporterListener.notify_event (
              self,
              tee)
```

Method to be notified from a TypeExporter.

Parameters

tee	[in] Notified event.

5.84 sparksee.TypeList Class Reference

Sparksee type identifier list.

Public Member Functions

- **clear** (self)

  Clears the list.
- __iter__ (self)

  Gets a new TypeListIterator.
- iterator (self)

  Gets a new TypeListIterator.
- __init__ (self)

  Constructor.
- count (self)

  Number of elements in the list.
- add (self, type)

  Adds a Sparksee type identifier at the end of the list.

5.84.1 Detailed Description

Sparksee type identifier list.

It stores a Sparksee node or edge type identifier list.

Use TypeListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.84.2 Constructor & Destructor Documentation

__init__()

```
sparksee.TypeList.__init__ (
              self)
```

Constructor.

This creates an empty list.

5.84.3 Member Function Documentation

__iter__()

```
sparksee.TypeList.__iter__ (
              self)
```

Gets a new TypeListIterator.

Returns

TypeListIterator instance

add()

```
sparksee.TypeList.add (
              self,
              type)
```

Adds a Sparksee type identifier at the end of the list.

Parameters

type	[in] Sparksee type identifier.

count()

```
sparksee.TypeList.count (
              self)
```

Number of elements in the list.

Returns

Number of elements in the list.

iterator()

```
sparksee.TypeList.iterator (
              self)
```

Gets a new TypeListIterator.

Returns

TypeListIterator instance.

5.85 sparksee.TypeListIterator Class Reference

TypeList iterator class.

Public Member Functions

- next (self)
  Moves to the next element.
- has_next (self)
  Gets if there are more elements.
- __next__ (self)
  Return the next item from the container.

5.85.1 Detailed Description

TypeList iterator class.

Iterator to traverse all the Sparksee node or edge type identifiers into a TypeList instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.85.2 Member Function Documentation

__next__()

```
sparksee.TypeListIterator.__next__ (
              self)
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

has_next()

```
sparksee.TypeListIterator.has_next (
              self)
```

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

next()

```
sparksee.TypeListIterator.next (
              self)
```

Moves to the next element.

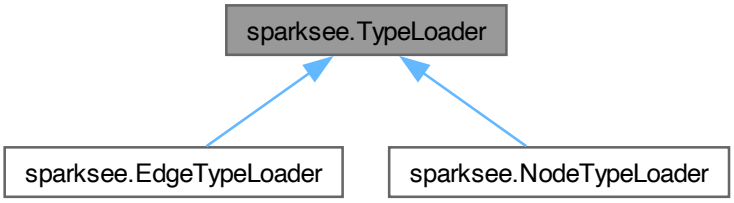
Returns

The next element.

5.86 sparksee.TypeLoader Class Reference

Base TypeLoader class.

Inheritance diagram for sparksee.TypeLoader:



Public Member Functions

- run (self)
  Run the loader.
- run_two_phases (self)
  Run the loader for two phases loading.
- set_log_error (self, path)
  Sets a log error file.
- set_type (self, type)
  Sets the type to be loaded.
- set_locale (self, locale_str)
  Sets the locale that will be used to read the data.
- set_timestamp_format (self, timestamp_format)
  Sets a specific timestamp format.
- set_attributes (self, attrs)
  Sets the list of Attributes.
- register (self, tel)
  Registers a new listener.
- set_frequency (self, freq)
  Sets the frequency of listener notification.
- set_row_reader (self, rr)
  Sets the input data source.
- set_attribute_positions (self, attrs_pos)
  Sets the list of attribute positions.
- run_n_phases (self, partitions)
  Run the loader for N phases loading.
- set_graph (self, graph)
  Sets the graph where the data will be loaded.
- set_log_off (self)
  Truns off all the error reporting.

5.86.1 Detailed Description

Base TypeLoader class.

Base class to load a node or edge type from a graph using a RowReader.

TypeLoaderListener can be registered to receive information about the progress of the load process by means of TypeLoaderEvent. The default frequency of notification to listeners is 100000.

Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.86.2 Member Function Documentation

register()

```
sparksee.TypeLoader.register (
              self,
              tel)
```

Registers a new listener.

Parameters

tel	TypeLoaderListener to be registered.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

run()

```
sparksee.TypeLoader.run (
              self)
```

Run the loader.

ErrorFor other issues

Exceptions

RuntimeError	null
IOError	If bad things happen using to the RowReader.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

run_n_phases()

```
sparksee.TypeLoader.run_n_phases (
              self,
              partitions)
```

Run the loader for N phases loading.

Firstly load all objects (and create them if necessary) and secondly loads all the attributes. But in this case, attributes are loaded one by one. This way, if there are three attributes, then 4 traverses are necessary.

Working on this mode it is necessary to build a temporary file. ErrorFor other issues

Parameters

partitions	[in] Number of horizontal partitions to perform the load.

Exceptions

RuntimeError	null
IOError	If bad things happen using to the RowReader.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

run_two_phases()

```
sparksee.TypeLoader.run_two_phases (
              self)
```

Run the loader for two phases loading.

Firstly load all objects (and create them if necessary) and secondly loads all the attributes.

Working on this mode it is necessary to build a temporary file. ErrorFor other issues

Exceptions

RuntimeError	null
IOError	If bad things happen using to the RowReader.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

set_attribute_positions()

```
sparksee.TypeLoader.set_attribute_positions (
              self,
              attrs_pos)
```

Sets the list of attribute positions.

Parameters

attrs_pos	[in] Attribute positions (column index >=0).

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

set_attributes()

```
sparksee.TypeLoader.set_attributes (
              self,
              attrs)
```

Sets the list of Attributes.

Parameters

attrs	[in] Attribute identifiers to be loaded

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

set_frequency()

```
sparksee.TypeLoader.set_frequency (
              self,
              freq)
```

Sets the frequency of listener notification.

Parameters

freq	[in] Frequency in number of rows managed to notify progress to all listeners

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

set_graph()

```
sparksee.TypeLoader.set_graph (
              self,
              graph)
```

Sets the graph where the data will be loaded.

Parameters

graph	[in] Graph.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

set_locale()

```
sparksee.TypeLoader.set_locale (
              self,
              locale_str)
```

Sets the locale that will be used to read the data.

It should match the locale used in the rowreader.

Parameters

locale_str	[in] The locale string for the read data. See CSVReader.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

set_log_error()

```
sparksee.TypeLoader.set_log_error (
              self,
              path)
```

Sets a log error file.

By default errors are thrown as a exception and the load process ends. If a log file is set, errors are logged there and the load process does not stop.

Parameters

path	[in] The path to the error log file.

Exceptions

IOError	If bad things happen opening the file.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

set_log_off()

```
sparksee.TypeLoader.set_log_off (
              self)
```

Truns off all the error reporting.

The log file will not be created and no exceptions for invalid data will be thrown. If you just want to turn off the logs, but abort at the first error what you should do is not call this method and not set a logError file.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

set_row_reader()

```
sparksee.TypeLoader.set_row_reader (
              self,
              rr)
```

Sets the input data source.

Parameters

rr	[in] Input RowReader.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

set_timestamp_format()

```
sparksee.TypeLoader.set_timestamp_format (
              self,
              timestamp_format)
```

Sets a specific timestamp format.

Parameters

timestamp_format	[in] A string with the timestamp format definition.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

set_type()

```
sparksee.TypeLoader.set_type (
              self,
              type)
```

Sets the type to be loaded.

Parameters

type	[in] Type identifier.

Reimplemented in sparksee.EdgeTypeLoader, and sparksee.NodeTypeLoader.

5.87 sparksee.TypeLoaderEvent Class Reference

Provides information about the progress of a TypeLoader instance.

5.87.1 Detailed Description

Provides information about the progress of a TypeLoader instance.

Check out the 'Data import' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

Edge data class.

It stores the tail and the head of an edge instance.

In case of undirected eges, the tail and the head are just the two ends of the edge.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.88 sparksee.TypeLoaderListener Class Reference

Public Member Functions

- notify_event (self, ev)

  Method to receive events from a Loader.

5.88.1 Member Function Documentation

notify_event()

```
sparksee.TypeLoaderListener.notify_event (
              self,
              ev)
```

Method to receive events from a Loader.

Parameters

ev	Loader.LoaderEvent with information from a running Loader.

5.89 sparksee.UnrecoverableError Class Reference

Enum representing unrecoverable errors.

Static Public Attributes

- int **SYSTEM_CALL_ERROR** = 0

  Error triggered due to an unrecoverable error detected during a system call in the Sparksee core.
- int **INVALID_CHECKSUM** = 1

  Error triggered after detecting an invalid checksum during reading data from an I/O device.

5.89.1 Detailed Description

Enum representing unrecoverable errors.

5.90 sparksee.Value Class Reference

Value class.

Public Member Functions

- set_null (self)
  Sets the Value to NULL.
- get_timestamp (self)
  Gets Timestamp Value.
- get_long (self)
  Gets Long Value.
- operator (self, value)
  Assignment operator.
- set_oid (self, value)
  Sets the Value.
- set_double_void (self, value)
  Sets the Value.
- set_boolean (self, value)
  Sets the Value.
- __init__ (self)
  Creates a new instance.
- get_integer (self)
  Gets Integer Value.
- set_timestamp_void (self, year, month, day, hour, minutes, seconds, millisecs)
  Sets the Value.
- equals (self, value)
  Compares with the given Value.
- set_boolean_void (self, value)
  Sets the Value.
- set_oid_void (self, value)
  Sets the OID Value.
- set_long_void (self, value)
  Sets the Value.
- set_string (self, value)
  Sets the Value.
- compare (self, value)
  Compares with the given Value.
- get_oid (self)
  Gets OID Value.
- get_boolean (self)
  Gets Boolean Value.
- set_timestamp (self, value)
  Sets the Value.
- get_data_type (self)

Gets the DataType.

- set_long (self, value)

  Sets the Value.
- set_integer (self, value)

  Sets the Value.
- set_double (self, value)

  Sets the Value.
- get_double (self)

  Gets Double Value.
- __init__ (self, value)

  Copy constructor.
- set_timestamp (self, year, month, day, hour, minutes, seconds, millisecs)

  Sets the Value.
- set_string_void (self, value)

  Sets the Value.
- set_void (self, value)

  Sets the Value.
- set_timestamp_void (self, value)

  Sets the Value.
- set_integer_void (self, value)

  Sets the Value.
- is_null (self)

  Gets if this is a NULL Value.
- get_string (self)

  Gets String Value.
- **set_null_void** (self)

  Sets the Value to NULL.
- **to_string** (self)

  *Returns a String representation of the Value, used in **unicode** and **str**.*

Static Public Attributes

- int **MAX_LENGTH_STRING** = 2047

  Maximum number of characters allowed for a String.

5.90.1 Detailed Description

Value class.

It is a container which stores a value and its data type (domain). A Value can be NULL.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.90.2 Constructor & Destructor Documentation

__init__() [1/2]

```
sparksee.Value.__init__ (
              self)
```

Creates a new instance.

It creates a NULL Value.

__init__() [2/2]

```
sparksee.Value.__init__ (
              self,
              value)
```

Copy constructor.

Parameters

value	[in] Value to be copied.

5.90.3 Member Function Documentation

compare()

```
sparksee.Value.compare (
              self,
              value)
```

Compares with the given Value.

It does not work if the given Value objects does not have the same DataType.

Parameters

value	Given value to compare to.

Returns

0 if this Value is equal to the given one; a value less than 0 if this Value is less than the given one; and a value greater than 0 if this Value is greater than the given one.

equals()

```
sparksee.Value.equals (
              self,
              value)
```

Compares with the given Value.

It does not work if the given Value objects does not have the same DataType.

Parameters

value	Given value to compare to.

Returns

TRUE if this Value is equal to the given one; FALSE otherwise.

get_boolean()

```
sparksee.Value.get_boolean (
              self)
```

Gets Boolean Value.

This must be a non-NULL Boolean Value.

Returns

The Boolean Value.

get_data_type()

```
sparksee.Value.get_data_type (
              self)
```

Gets the DataType.

Value cannot be NULL.

Returns

The DataType.

get_double()

```
sparksee.Value.get_double (
              self)
```

Gets Double Value.

This must be a non-NULL Double Value.

Returns

The Double Value.

get_integer()

```
sparksee.Value.get_integer (
              self)
```

Gets Integer Value.

This must be a non-NULL Integer Value.

Returns

The Integer Value.

get_long()

```
sparksee.Value.get_long (
              self)
```

Gets Long Value.

This must be a non-NULL Long Value.

Returns

The Long Value.

get_oid()

```
sparksee.Value.get_oid (
              self)
```

Gets OID Value.

This must be an non-NULL OID Value.

Returns

The OID Value.

get_string()

```
sparksee.Value.get_string (
              self)
```

Gets String Value.

This must be a non-NULL String Value.

Returns

The String Value.

get_timestamp()

```
sparksee.Value.get_timestamp (
              self)
```

Gets Timestamp Value.

This must be a non-NULL Timestamp Value.

Returns

The Timestamp Value.

is_null()

```
sparksee.Value.is_null (
              self)
```

Gets if this is a NULL Value.

Returns

TRUE if this is a NULL Value, FALSE otherwise.

operator()

```
sparksee.Value.operator (
              self,
              value)
```

Assignment operator.

Parameters

value	[in] Value to be copied.

Returns

Returns the Value reference.

set_boolean()

```
sparksee.Value.set_boolean (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] Nex Boolean value.

Returns

The calling instance.

set_boolean_void()

```
sparksee.Value.set_boolean_void (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New Boolean value.

set_double()

```
sparksee.Value.set_double (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New Double value.

Returns

The calling instance.

set_double_void()

```
sparksee.Value.set_double_void (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New Double value.

set_integer()

```
sparksee.Value.set_integer (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New Integer value.

Returns

The calling instance.

set_integer_void()

```
sparksee.Value.set_integer_void (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New Integer value.

set_long()

```
sparksee.Value.set_long (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New Long value.

Returns

The calling instance.

set_long_void()

```
sparksee.Value.set_long_void (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New Long value.

set_null()

```
sparksee.Value.set_null (
              self)
```

Sets the Value to NULL.

Returns

The calling instance.

set_oid()

```
sparksee.Value.set_oid (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New OID Value.

Returns

The calling instance.

set_oid_void()

```
sparksee.Value.set_oid_void (
              self,
              value)
```

Sets the OID Value.

Parameters

value	[in] New OID value.

set_string()

```
sparksee.Value.set_string (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New String value.

Returns

The calling instance.

set_string_void()

```
sparksee.Value.set_string_void (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New String value.

set_timestamp() [1/2]

```
sparksee.Value.set_timestamp (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New Timestamp value.

Returns

The calling instance.

set_timestamp() [2/2]

```
sparksee.Value.set_timestamp (
              self,
              year,
              month,
              day,
              hour,
              minutes,
              seconds,
              millisecs)
```

Sets the Value.

Parameters

year	[in] The year (>=1970).
month	[in] The month ([1..12]).
day	[in] The of the month ([1..31]).

hour	[in] The hour ([0..23]).
minutes	[in] The minutes ([0..59]).
seconds	[in] The seconds ([0..59]).
millisecs	[in] The milliseconds ([0..999]).

Returns

The calling instance.

set_timestamp_void() [1/2]

```
sparksee.Value.set_timestamp_void (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New Timestamp value.

set_timestamp_void() [2/2]

```
sparksee.Value.set_timestamp_void (
              self,
              year,
              month,
              day,
              hour,
              minutes,
              seconds,
              millisecs)
```

Sets the Value.

Parameters

year	[in] The year (>=1970).
month	[in] The month ([1..12]).
day	[in] The of the month ([1..31]).

hour	[in] The hour ([0..23]).
minutes	[in] The minutes ([0..59]).
seconds	[in] The seconds ([0..59]).
millisecs	[in] The milliseconds ([0..999]).

set_void()

```
sparksee.Value.set_void (
              self,
              value)
```

Sets the Value.

Parameters

value	[in] New value.

5.91 sparksee.ValueArray Class Reference

ValueArray class.

Public Member Functions

- set (self, index, value)
  Set a Value to a specific array position.
- get_integer (self)
  Get all the values from this int array.
- get_long (self)
  Get all the values from this long array.
- set_timestamp (self, values)
  Set all the values of this timestamp array.
- set_oid (self, values)
  Set all the values of this oid array.
- get_oid (self)
  Get all the values from this oid array.
- get (self, index, value)
  Get a Value from the array.
- get_timestamp (self)
  Get all the values from this timestamp array.
- get_double_range (self, index, size)
  Get a subset of the values from this double array.
- get_integer_range (self, index, size)
  Get a subset of the values from this integer array.
- get_boolean (self)

Get all the values from this bool array.

- set_boolean (self, values)

  Set all the values of this bool array.
- set_double_range (self, index, values)

  Set a subset of the values from this double array.
- size (self)

  Returns the array size.
- get_long_range (self, index, size)

  Get a subset of the values from this long array.
- set_timestamp_range (self, index, values)

  Set a subset of the values from this timestamp array.
- make_null (self)

  Sets the attribute array to Null.
- set_integer_range (self, index, values)

  Set a subset of the values from this integer array.
- get_oid_range (self, index, size)

  Get a subset of the values from this oid array.
- set_boolean_range (self, index, values)

  Set a subset of the values from this boolean array.
- set_oid_range (self, index, values)

  Set a subset of the values from this oid array.
- get_double (self)

  Get all the values from this double array.
- set_long (self, values)

  Set all the values of this long array.
- set_double (self, values)

  Set all the values of this double array.
- set_integer (self, values)

  Set all the values of this int array.
- set_long_range (self, index, values)

  Set a subset of the values from this long array.
- set (self, value)

  Set a Value to the whole array.
- get_boolean_range (self, index, size)

  Get a subset of the values from this boolean array.
- get_timestamp_range (self, index, size)

  Get a subset of the values from this timestamp array.

5.91.1 Detailed Description

ValueArray class.

It allows for getting and setting ValueArray attribute values.

It is very important to close the ValueArray once no more get or set operations will be performed.

Creation of a new ValueArray: (i) Set all the ValueArray elements using Graph::SetAttributeArray (ii) perform as many get/set operations as you need to the ValueArray instance. Lastly, (iii) Close the ValueArray

Use of an existing ValueArray: (i) Get a ValueArray instance using Graph::GetAttributeArray (ii) perform as many get/set operations as you need to the ValueArray instance. Lastly, (iii) Close the ValueArray

Check out the 'Attributes and values' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.91.2 Member Function Documentation

get()

```
sparksee.ValueArray.get (
              self,
              index,
              value)
```

Get a Value from the array.

AppErrorIf the ValueArray is not available

Parameters

index	[in] Position of the element to get [0..N-1]
value	[out] Value to get the array element

Exceptions

ValueError	If the index is out of range or the Value not valid
RuntimeError	null

get_boolean()

```
sparksee.ValueArray.get_boolean (
              self)
```

Get all the values from this bool array.

UnsupportedOperationErrorIf array DataType is not bool

Returns

All the array values

Exceptions

RuntimeError	If the ValueArray is not available
RuntimeError	null

get_boolean_range()

```
sparksee.ValueArray.get_boolean_range (
              self,
              index,
              size)
```

Get a subset of the values from this boolean array.

Returns all the requested values

Parameters

index	[in] Position of the first element to get [0..N-1]
size	[in] Number of elements to get [1..N]

get_double()

```
sparksee.ValueArray.get_double (
              self)
```

Get all the values from this double array.

UnsupportedOperationErrorIf array DataType is not Double

Returns

All the array values

Exceptions

RuntimeError	If the ValueArray is not available
RuntimeError	null

get_double_range()

```
sparksee.ValueArray.get_double_range (
              self,
              index,
              size)
```

Get a subset of the values from this double array.

Returns all the requested values

Parameters

index	[in] Position of the first element to get [0..N-1]
size	[in] Number of elements to get [1..N]

get_integer()

```
sparksee.ValueArray.get_integer (
              self)
```

Get all the values from this int array.

UnsupportedOperationErrorIf array DataType is not int

Returns

All the array values

Exceptions

RuntimeError	If the ValueArray is not available
RuntimeError	null

get_integer_range()

```
sparksee.ValueArray.get_integer_range (
              self,
              index,
              size)
```

Get a subset of the values from this integer array.

Returns all the requested values

Parameters

index	[in] Position of the first element to get [0..N-1]
size	[in] Number of elements to get [1..N]

get_long()

```
sparksee.ValueArray.get_long (
              self)
```

Get all the values from this long array.

UnsupportedOperationErrorIf array DataType is not long

Returns

All the array values

Exceptions

RuntimeError	If the ValueArray is not available
RuntimeError	null

get_long_range()

```
sparksee.ValueArray.get_long_range (
              self,
              index,
              size)
```

Get a subset of the values from this long array.

Returns all the requested values

Parameters

index	[in] Position of the first element to get [0..N-1]
size	[in] Number of elements to get [1..N]

get_oid()

```
sparksee.ValueArray.get_oid (
              self)
```

Get all the values from this oid array.

UnsupportedOperationErrorIf array DataType is not oid

Returns

All the array values

Exceptions

RuntimeError	If the ValueArray is not available
RuntimeError	null

get_oid_range()

```
sparksee.ValueArray.get_oid_range (
                self,
                index,
                size)
```

Get a subset of the values from this oid array.

Returns all the requested values

Parameters

index	[in] Position of the first element to get [0..N-1]
size	[in] Number of elements to get [1..N]

get_timestamp()

```
sparksee.ValueArray.get_timestamp (
                self)
```

Get all the values from this timestamp array.

UnsupportedOperationErrorIf array DataType is not timestamp

Returns

All the array values

Exceptions

RuntimeError	If the ValueArray is not available
RuntimeError	null

get_timestamp_range()

```
sparksee.ValueArray.get_timestamp_range (
              self,
              index,
              size)
```

Get a subset of the values from this timestamp array.

Returns all the requested values

Parameters

index	[in] Position of the first element to get [0..N-1]
size	[in] Number of elements to get [1..N]

make_null()

```
sparksee.ValueArray.make_null (
              self)
```

Sets the attribute array to Null.

The ValueArray can not be used after this call.

Exceptions

RuntimeError	null

set() [1/2]

```
sparksee.ValueArray.set (
              self,
              index,
              value)
```

Set a Value to a specific array position.

AppErrorIf the ValueArray is not available

Parameters

index	[in] Position of the element to set [0..N-1]
value	[in] Value to set in the array element

Exceptions

ValueError	If the index is out of range or the Value not valid
RuntimeError	null

set() [2/2]

```
sparksee.ValueArray.set (
            self,
            value)
```

Set a Value to the whole array.

AppErrorIf the ValueArray is not available

Parameters

value	[in] Value to set in all the array elements

Exceptions

ValueError	If the Value is not valid
RuntimeError	null

set_boolean()

```
sparksee.ValueArray.set_boolean (
            self,
            values)
```

Set all the values of this bool array.

AppErrorIf the ValueArray is not available

Parameters

values	[in] All the values to set

Exceptions

RuntimeError	null
RuntimeError	If array DataType is not bool

set_boolean_range()

```
sparksee.ValueArray.set_boolean_range (
              self,
              index,
              values)
```

Set a subset of the values from this boolean array.

Parameters

index	[in] Position of the first element to get [0..N-1]
values	[in] All the values to set

set_double()

```
sparksee.ValueArray.set_double (
              self,
              values)
```

Set all the values of this double array.

AppErrorIf the ValueArray is not available

Parameters

values	[in] All the values to set

Exceptions

RuntimeError	null
RuntimeError	If array DataType is not Double

set_double_range()

```
sparksee.ValueArray.set_double_range (
              self,
              index,
              values)
```

Set a subset of the values from this double array.

Parameters

index	[in] Position of the first element to get [0..N-1]
values	[in] All the values to set

set_integer()

```
sparksee.ValueArray.set_integer (
              self,
              values)
```

Set all the values of this int array.

AppErrorIf the ValueArray is not available

Parameters

values	[in] All the values to set

Exceptions

RuntimeError	null
RuntimeError	If array DataType is not int

set_integer_range()

```
sparksee.ValueArray.set_integer_range (
              self,
              index,
              values)
```

Set a subset of the values from this integer array.

Parameters

index	[in] Position of the first element to get [0..N-1]
values	[in] All the values to set

set_long()

```
sparksee.ValueArray.set_long (
              self,
              values)
```

Set all the values of this long array.

AppErrorIf the ValueArray is not available

Parameters

values	[in] All the values to set

Exceptions

RuntimeError	null
RuntimeError	If array DataType is not long

set_long_range()

```
sparksee.ValueArray.set_long_range (
              self,
              index,
              values)
```

Set a subset of the values from this long array.

Parameters

index	[in] Position of the first element to get [0..N-1]
values	[in] All the values to set

set_oid()

```
sparksee.ValueArray.set_oid (
              self,
              values)
```

Set all the values of this oid array.

AppErrorIf the ValueArray is not available

Parameters

values	[in] All the values to set

Exceptions

RuntimeError	null
RuntimeError	If array DataType is not oid

set_oid_range()

```
sparksee.ValueArray.set_oid_range (
              self,
              index,
              values)
```

Set a subset of the values from this oid array.

Parameters

index	[in] Position of the first element to get [0..N-1]
values	[in] All the values to set

set_timestamp()

```
sparksee.ValueArray.set_timestamp (
              self,
              values)
```

Set all the values of this timestamp array.

AppErrorIf the ValueArray is not available

Parameters

values	[in] All the values to set

Exceptions

RuntimeError	null
RuntimeError	If array DataType is not timestamp

set_timestamp_range()

```
sparksee.ValueArray.set_timestamp_range (
              self,
              index,
              values)
```

Set a subset of the values from this timestamp array.

Parameters

index	[in] Position of the first element to get [0..N-1]
values	[in] All the values to set

size()

```
sparksee.ValueArray.size (
              self)
```

Returns the array size.

Returns

The number of elements in the array.

5.92 sparksee.ValueList Class Reference

Value list.

Public Member Functions

- **clear** (self)

  Clears the list.
- get (self, index)

  Returns the Value at the specified position in the list.
- __iter__ (self)

  Gets a new ValueListIterator.
- iterator (self)

  Gets a new ValueListIterator.
- __init__ (self)

  Constructor.
- add (self, value)

  Adds a value to the end of the list.
- count (self)

  Number of elements in the list.

5.92.1 Detailed Description

Value list.

It stores a Value list.

Use ValueListIterator to access all elements into this collection.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.92.2 Constructor & Destructor Documentation

__init__()

```
sparksee.ValueList.__init__ (
              self)
```

Constructor.

This creates an empty list.

5.92.3 Member Function Documentation

__iter__()

```
sparksee.ValueList.__iter__ (
              self)
```

Gets a new ValueListIterator.

Returns

ValueListIterator instance

add()

```
sparksee.ValueList.add (
              self,
              value)
```

Adds a value to the end of the list.

Parameters

value	[in] The value to add

count()

```
sparksee.ValueList.count (
              self)
```

Number of elements in the list.

Returns

Number of elements in the list.

get()

```
sparksee.ValueList.get (
              self,
              index)
```

Returns the Value at the specified position in the list.

Parameters

index	[in] Index of the element to return, starting at 0.

iterator()

```
sparksee.ValueList.iterator (
              self)
```

Gets a new ValueListIterator.

Returns

ValueListIterator instance.

5.93 sparksee.ValueListIterator Class Reference

Public Member Functions

- next (self)
  Moves to the next element.
- has_next (self)
  Gets if there are more elements.
- __next__ (self)
  Used in next().

5.93.1 Member Function Documentation

__next__()

```
sparksee.ValueListIterator.__next__ (
              self)
```

Used in next().

Returns

The next element

has_next()

```
sparksee.ValueListIterator.has_next (
              self)
```

Gets if there are more elements.

Returns

TRUE if there are more elements, FALSE otherwise.

next()

```
sparksee.ValueListIterator.next (
              self)
```

Moves to the next element.

Returns

The next element.

5.94 sparksee.Values Class Reference

Value set class.

Public Member Functions

- count (self)
  Gets the number of elements into the collection.
- __iter__ (self)
  Gets a new ValuesIterator.
- iterator (self, order)
  Gets a ValuesIterator.
- close (self)
  Closes the Values instance.
- is_closed (self)
  Gets if Values instance has been closed or not.

5.94.1 Detailed Description

Value set class.

This is a set of Value instances, that is there is no duplicated elements.

Use a ValuesIterator to traverse all the elements into the set.

When the Values instance is closed, it closes all existing and non-closed ValuesIterator instances too.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.94.2 Member Function Documentation

__iter__()

```
sparksee.Values.__iter__ (
              self)
```

Gets a new ValuesIterator.

Returns

ValuesIterator instance

close()

```
sparksee.Values.close (
              self)
```

Closes the Values instance.

It must be called to ensure the integrity of all data.

count()

```
sparksee.Values.count (
              self)
```

Gets the number of elements into the collection.

Returns

The number of elements into the collection.

is_closed()

```
sparksee.Values.is_closed (
              self)
```

Gets if Values instance has been closed or not.

See also

close()

Returns

TRUE if the Values instance has been closed, FALSE otherwise.

iterator()

```
sparksee.Values.iterator (
              self,
              order)
```

Gets a ValuesIterator.

Parameters

order	[in] Ascending or descending order.

Returns

ValuesIterator instance.

5.95 sparksee.ValuesIterator Class Reference

Values iterator class.

Public Member Functions

- next (self)
  Gets the next element to traverse.
- has_next (self)
  Gets if there are more elements to traverse.
- close (self)
  Closes the ValuesIterator instance.
- __next__ (self)
  Return the next item from the container.
- is_closed (self)
  Gets if ValuesIterator instance has been closed or not.

5.95.1 Detailed Description

Values iterator class.

It allows for traversing all the elements into a Values instance.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.95.2 Member Function Documentation

__next__()

```
sparksee.ValuesIterator.__next__ (
              self)
```

Return the next item from the container.

If there are no further items, raise the StopIteration exception.

Returns

The next element

close()

```
sparksee.ValuesIterator.close (
              self)
```

Closes the ValuesIterator instance.

It must be called to ensure the integrity of all data.

has_next()

```
sparksee.ValuesIterator.has_next (
              self)
```

Gets if there are more elements to traverse.

Returns

TRUE if there are more elements to traverse, FALSE otherwise.

is_closed()

```
sparksee.ValuesIterator.is_closed (
              self)
```

Gets if ValuesIterator instance has been closed or not.

See also

close()

Returns

TRUE if the ValuesIterator instance has been closed, FALSE otherwise.

next()

```
sparksee.ValuesIterator.next (
              self)
```

Gets the next element to traverse.

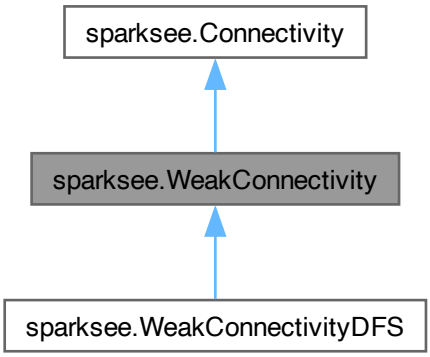
Returns

The next element.

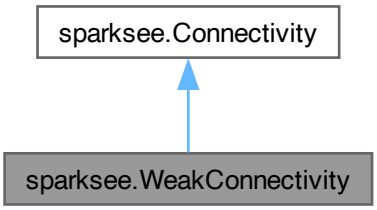
5.96 sparksee.WeakConnectivity Class Reference

WeakConnectivity class.

Inheritance diagram for sparksee.WeakConnectivity:



Collaboration diagram for sparksee.WeakConnectivity:



Public Member Functions

- run (self)

  Runs the algorithm in order to find the connected components.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- add_all_edge_types (self)

  Allows connectivity through all edge types of the graph.
- get_connected_components (self)

  Returns the results generated by the execution of the algorithm.
- add_edge_type (self, type)

  Allows connectivity through edges of the given type.

- exclude_edges (self, edges)

  Set which edges can't be used.
- add_node_type (self, t)

  Allows connectivity through nodes of the given type.
- set_materialized_attribute (self, attribute_name)

  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.
- **add_all_node_types** (self)

  Allows connectivity through all node types of the graph.
- close (self)

  Closes the Connectivity instance.
- is_closed (self)

  Gets if Connectivity has been closed or not.

5.96.1 Detailed Description

WeakConnectivity class.

Any class implementing this abstract class can be used to solve the problem of finding weakly connected components in an undirected graph or in a directed graph which will be considered as an undirected one.

It consists in finding components where every pair (u,v) of nodes contained in it has a path from u to v and from v to u.

It is possible to set some restrictions after constructing a new instance of this class and before running it in order to limit the results.

After the execution, we can retrieve the results stored in an instance of the ConnectedComponents class using the getConnectedComponents method.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.96.2 Member Function Documentation

add_all_edge_types()

```
sparksee.WeakConnectivity.add_all_edge_types (
              self)
```

Allows connectivity through all edge types of the graph.

In a weak connectivity the edges can be used in Any direction.

Reimplemented in sparksee.WeakConnectivityDFS.

add_edge_type()

```
sparksee.WeakConnectivity.add_edge_type (
              self,
              type)
```

Allows connectivity through edges of the given type.

In a weak connectivity the edges can be used in Any direction.

Parameters

type	[in] Edge type.

Reimplemented in sparksee.WeakConnectivityDFS.

add_node_type()

```
sparksee.WeakConnectivity.add_node_type (
              self,
              t)
```

Allows connectivity through nodes of the given type.

Parameters

t	null

Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.WeakConnectivityDFS.

close()

```
sparksee.Connectivity.close (
              self)  [inherited]
```

Closes the Connectivity instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.WeakConnectivity.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.WeakConnectivityDFS.

exclude_nodes()

```
sparksee.WeakConnectivity.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.WeakConnectivityDFS.

get_connected_components()

```
sparksee.WeakConnectivity.get_connected_components (
              self)
```

Returns the results generated by the execution of the algorithm.

These results contain information related to the connected components found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class ConnectedComponents which contain information related to the connected components found.

Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.WeakConnectivityDFS.

is_closed()

```
sparksee.Connectivity.is_closed (
              self)  [inherited]
```

Gets if Connectivity has been closed or not.

See also

close()

Returns

TRUE if the Connectivity instance has been closed, FALSE otherwise.

run()

```
sparksee.WeakConnectivity.run (
              self)
```

Runs the algorithm in order to find the connected components.

This method can be called only once.

Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.WeakConnectivityDFS.

set_materialized_attribute()

```
sparksee.WeakConnectivity.set_materialized_attribute (
              self,
              attribute_name)
```

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class ConnectedComponents indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the connected components found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

attribute_name	[in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm.

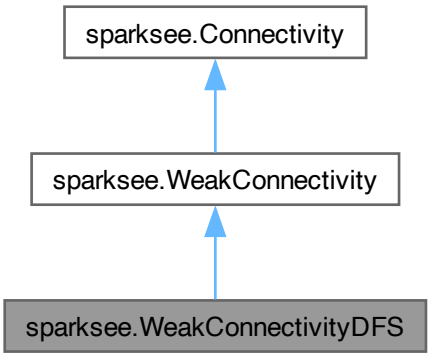
Reimplemented from sparksee.Connectivity.

Reimplemented in sparksee.WeakConnectivityDFS.

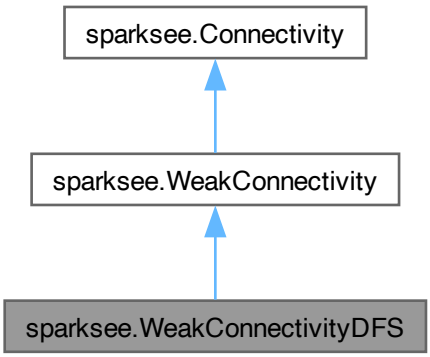
5.97 sparksee.WeakConnectivityDFS Class Reference

WeakConnectivityDFS class.

Inheritance diagram for sparksee.WeakConnectivityDFS:



Collaboration diagram for sparksee.WeakConnectivityDFS:



Public Member Functions

- __init__ (self, session)

  Creates a new instance of WeakConnectivityDFS.
- exclude_nodes (self, nodes)

  Set which nodes can't be used.
- add_all_edge_types (self)

  Allows connectivity through all edge types of the graph.

- get_connected_components (self)

  Returns the results generated by the execution of the algorithm.
- **run** (self)

  Executes the algorithm.
- add_edge_type (self, type)

  Allows connectivity through edges of the given type.
- exclude_edges (self, edges)

  Set which edges can't be used.
- add_node_type (self, t)

  Allows connectivity through nodes of the given type.
- set_materialized_attribute (self, attribute_name)

  Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.
- **add_all_node_types** (self)

  Allows connectivity through all node types of the graph.
- close (self)

  Closes the Connectivity instance.
- is_closed (self)

  Gets if Connectivity has been closed or not.

5.97.1 Detailed Description

WeakConnectivityDFS class.

This class can be used to solve the problem of finding weakly connected components in an undirected graph or in a directed graph which will be considered as an undirected one.

It consists in finding components where every pair (u,v) of nodes contained in it has a path from u to v and from v to u. This implementation is based on the Depth-First Search (DFS) algorithm.

It is possible to set some restrictions after constructing a new instance of this class and before running it in order to limit the results.

After the execution, we can retrieve the results stored in an instance of the ConnectedComponents class using the getConnectedComponents method.

Check out the 'Algorithms' section in the SPARKSEE User Manual for more details on this.

Author

Sparsity Technologies http://www.sparsity-technologies.com

5.97.2 Constructor & Destructor Documentation

__init__()

```
sparksee.WeakConnectivityDFS.__init__ (
              self,
              session)
```

Creates a new instance of WeakConnectivityDFS.

After creating this instance is required to indicate the set of edge types and the set of node types which will be navigated through while traversing the graph in order to find the weak connected components.

Parameters

session	[in] Session to get the graph from and calculate the connectivity

5.97.3 Member Function Documentation

add_all_edge_types()

```
sparksee.WeakConnectivityDFS.add_all_edge_types (
              self)
```

Allows connectivity through all edge types of the graph.

In a weak connectivity the edges can be used in Any direction.

Reimplemented from sparksee.WeakConnectivity.

add_edge_type()

```
sparksee.WeakConnectivityDFS.add_edge_type (
              self,
              type)
```

Allows connectivity through edges of the given type.

In a weak connectivity the edges can be used in Any direction.

Parameters

type	[in] Edge type.

Reimplemented from sparksee.WeakConnectivity.

add_node_type()

```
sparksee.WeakConnectivityDFS.add_node_type (
              self,
              t)
```

Allows connectivity through nodes of the given type.

Parameters

t	null

Reimplemented from sparksee.WeakConnectivity.

close()

```
sparksee.Connectivity.close (
              self)  [inherited]
```

Closes the Connectivity instance.

It must be called to ensure the integrity of all data.

exclude_edges()

```
sparksee.WeakConnectivityDFS.exclude_edges (
              self,
              edges)
```

Set which edges can't be used.

This will replace any previously specified set of excluded edges. Should only be used to exclude the usage of specific edges from allowed edge types because it's less efficient than not allowing an edge type.

Parameters

edges	[in] A set of edge identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.WeakConnectivity.

exclude_nodes()

```
sparksee.WeakConnectivityDFS.exclude_nodes (
              self,
              nodes)
```

Set which nodes can't be used.

This will replace any previously specified set of excluded nodes. Should only be used to exclude the usage of specific nodes from allowed node types because it's less efficient than not allowing a node type.

Parameters

nodes	[in] A set of node identifiers that must be kept intact until the destruction of the class.

Reimplemented from sparksee.WeakConnectivity.

get_connected_components()

```
sparksee.WeakConnectivityDFS.get_connected_components (
              self)
```

Returns the results generated by the execution of the algorithm.

These results contain information related to the connected components found as the number of different components, the set of nodes contained in each component or many other data.

Returns

Returns an instance of the class ConnectedComponents which contain information related to the connected components found.

Reimplemented from sparksee.WeakConnectivity.

is_closed()

```
sparksee.Connectivity.is_closed (
              self)  [inherited]
```

Gets if Connectivity has been closed or not.

See also

close()

Returns

TRUE if the Connectivity instance has been closed, FALSE otherwise.

set_materialized_attribute()

```
sparksee.WeakConnectivityDFS.set_materialized_attribute (
              self,
              attribute_name)
```

Creates a new common attribute type for all node types in the graph in order to store, persistently, the results related to the connected components found while executing this algorithm.

Whenever the user wants to retrieve the results, even when the graph has been closed and opened again, it is only necessary to create a new instance of the class ConnectedComponents indicating the graph and the name of the common attribute type which stores the results. This instance will have all the information related to the connected components found in the moment of the execution of the algorithm that stored this data.

It is possible to run the algorithm without specifying this parameter in order to avoid materializing the results of the execution.

Parameters

attribute_name	[in] The name of the common attribute type for all node types in the graph which will store persistently the results generated by the execution of the algorithm.

Reimplemented from sparksee.WeakConnectivity.

Index

__init__
- sparksee.AttributeList, 20
- sparksee.BooleanList, 28
- sparksee.CommunitiesSCD, 33
- sparksee.ConnectedComponents, 45
- sparksee.Context, 52
- sparksee.DisjointCommunities, 83
- sparksee.EdgeTypeExporter, 99
- sparksee.EdgeTypeLoader, 106
- sparksee.Int32List, 158
- sparksee.NodeTypeExporter, 175
- sparksee.NodeTypeLoader, 180
- sparksee.OIDList, 197
- sparksee.PageRank, 201
- sparksee.RandomWalk, 217
- sparksee.ResultSetList, 227
- sparksee.SinglePairShortestPathBFS, 251
- sparksee.SinglePairShortestPathDijkstra, 257
- sparksee.Sparksee, 265
- sparksee.SparkseeConfig, 279, 280
- sparksee.StringList, 305
- sparksee.StrongConnectivityGabow, 315
- sparksee.TextStream, 319
- sparksee.TraversalBFS, 328
- sparksee.TraversalDFS, 334
- sparksee.TypeList, 348
- sparksee.Value, 360
- sparksee.ValueList, 383
- sparksee.WeakConnectivityDFS, 397

__iter__
- sparksee.AttributeList, 20
- sparksee.BooleanList, 28
- sparksee.Int32List, 158
- sparksee.Objects, 186
- sparksee.OIDList, 197
- sparksee.RandomWalk, 217
- sparksee.ResultSetList, 227
- sparksee.StringList, 305
- sparksee.Traversal, 322
- sparksee.TraversalBFS, 328
- sparksee.TraversalDFS, 334
- sparksee.TypeList, 348
- sparksee.ValueList, 384
- sparksee.Values, 386

__next__
- sparksee.AttributeListIterator, 22
- sparksee.BooleanListIterator, 30
- sparksee.Int32ListIterator, 159
- sparksee.KeyValues, 162
- sparksee.ObjectsIterator, 195
- sparksee.OIDListIterator, 199
- sparksee.RandomWalk, 217
- sparksee.ResultSetListIterator, 229
- sparksee.StringListIterator, 307
- sparksee.Traversal, 322
- sparksee.TraversalBFS, 328
- sparksee.TraversalDFS, 334
- sparksee.TypeListIterator, 349
- sparksee.ValueListIterator, 385
- sparksee.ValuesIterator, 388

add
- sparksee.AttributeList, 20
- sparksee.BooleanList, 28
- sparksee.Int32List, 158
- sparksee.Objects, 186
- sparksee.OIDList, 197
- sparksee.StringList, 305
- sparksee.TypeList, 348
- sparksee.ValueList, 384

add_all_edge_types
- sparksee.CommunitiesSCD, 33
- sparksee.Context, 52
- sparksee.DisjointCommunityDetection, 87
- sparksee.PageRank, 202
- sparksee.RandomWalk, 217
- sparksee.ShortestPath, 240
- sparksee.SinglePairShortestPath, 245
- sparksee.SinglePairShortestPathBFS, 251
- sparksee.SinglePairShortestPathDijkstra, 258
- sparksee.StrongConnectivity, 309
- sparksee.StrongConnectivityGabow, 315
- sparksee.Traversal, 322
- sparksee.TraversalBFS, 328
- sparksee.TraversalDFS, 334
- sparksee.WeakConnectivity, 391
- sparksee.WeakConnectivityDFS, 397

add_checksums
- sparksee.Sparksee, 265

add_edge_type
- sparksee.CommunitiesSCD, 33
- sparksee.Context, 52
- sparksee.DisjointCommunityDetection, 87
- sparksee.PageRank, 202
- sparksee.RandomWalk, 218
- sparksee.ShortestPath, 241
- sparksee.SinglePairShortestPath, 245
- sparksee.SinglePairShortestPathBFS, 252
- sparksee.SinglePairShortestPathDijkstra, 258
- sparksee.StrongConnectivity, 309
- sparksee.StrongConnectivityGabow, 315
- sparksee.Traversal, 323
- sparksee.TraversalBFS, 329
- sparksee.TraversalDFS, 335
- sparksee.WeakConnectivity, 391
- sparksee.WeakConnectivityDFS, 397

add_node_type
- sparksee.CommunitiesSCD, 33
- sparksee.CommunityDetection, 38
- sparksee.Connectivity, 48

sparksee.Context, 52
sparksee.DisjointCommunityDetection, 88
sparksee.PageRank, 202
sparksee.RandomWalk, 218
sparksee.ShortestPath, 241
sparksee.SinglePairShortestPath, 245
sparksee.SinglePairShortestPathBFS, 252
sparksee.SinglePairShortestPathDijkstra, 258
sparksee.StrongConnectivity, 310
sparksee.StrongConnectivityGabow, 315
sparksee.Traversal, 323
sparksee.TraversalBFS, 329
sparksee.TraversalDFS, 335
sparksee.WeakConnectivity, 392
sparksee.WeakConnectivityDFS, 397
add_query_stream_manager
sparksee.Database, 67
add_weighted_edge_type
sparksee.SinglePairShortestPathDijkstra, 258
any
sparksee.Objects, 187
as_directed
sparksee.EdgeExport, 94

backup
sparksee.Graph, 123
BETWEEN
sparksee.Condition, 42

calculate_edge_cost
sparksee.SinglePairShortestPathDijkstraDynamicCost, 263
check_only_existence
sparksee.SinglePairShortestPathBFS, 252
close
sparksee.CommunitiesSCD, 34
sparksee.CommunityDetection, 38
sparksee.ConnectedComponents, 45
sparksee.Connectivity, 48
sparksee.Context, 53
sparksee.CSVReader, 57
sparksee.CSVWriter, 63
sparksee.Database, 67
sparksee.DisjointCommunities, 84
sparksee.DisjointCommunityDetection, 88
sparksee.KeyValues, 162
sparksee.Objects, 187
sparksee.ObjectsIterator, 195
sparksee.Query, 209
sparksee.RandomWalk, 218
sparksee.RowReader, 231
sparksee.RowWriter, 233
sparksee.Session, 237
sparksee.ShortestPath, 241
sparksee.SinglePairShortestPath, 246
sparksee.SinglePairShortestPathBFS, 252
sparksee.SinglePairShortestPathDijkstra, 259
sparksee.Sparksee, 265
sparksee.StrongConnectivity, 310
sparksee.StrongConnectivityGabow, 316
sparksee.TextStream, 319
sparksee.Traversal, 323
sparksee.TraversalBFS, 329
sparksee.TraversalDFS, 335
sparksee.Values, 386
sparksee.ValuesIterator, 388
sparksee.WeakConnectivity, 392
sparksee.WeakConnectivityDFS, 398
combine_difference
sparksee.Objects, 187
combine_intersection
sparksee.Objects, 188
combine_union
sparksee.Objects, 188
compare
sparksee.Value, 360
compute
sparksee.Context, 53
CONFIG
sparksee.LogLevel, 164
contains
sparksee.Objects, 189
copy
sparksee.Objects, 189
copy_defaults
sparksee.EdgeExport, 94
sparksee.NodeExport, 168
count
sparksee.AttributeList, 21
sparksee.BooleanList, 28
sparksee.Int32List, 158
sparksee.Objects, 190
sparksee.OIDList, 198
sparksee.ResultSetList, 227
sparksee.StringList, 306
sparksee.TypeList, 348
sparksee.ValueList, 384
sparksee.Values, 387
count_edges
sparksee.Graph, 123
count_nodes
sparksee.Graph, 124
create
sparksee.Sparksee, 265, 266

DEBUG
sparksee.LogLevel, 164
decrypt
sparksee.Sparksee, 267
degree
sparksee.Graph, 124
difference
sparksee.Objects, 190
download_license
sparksee.SparkseeConfig, 280
drop
sparksee.Graph, 124, 125
dump_data

sparksee.Graph, 125
dump_schema
sparksee.Database, 67
dump_storage
sparksee.Graph, 125

edges
sparksee.Graph, 126
enable_type
sparksee.DefaultExport, 75
sparksee.ExportManager, 114
encrypt
sparksee.Sparksee, 267
encrypted_backup
sparksee.Graph, 126
EQUAL
sparksee.Condition, 42
equals
sparksee.Objects, 190
sparksee.Value, 360
exclude_edges
sparksee.CommunitiesSCD, 34
sparksee.CommunityDetection, 39
sparksee.Connectivity, 48
sparksee.Context, 54
sparksee.DisjointCommunityDetection, 88
sparksee.RandomWalk, 219
sparksee.ShortestPath, 241
sparksee.SinglePairShortestPath, 246
sparksee.SinglePairShortestPathBFS, 253
sparksee.SinglePairShortestPathDijkstra, 259
sparksee.StrongConnectivity, 310
sparksee.StrongConnectivityGabow, 316
sparksee.Traversal, 323
sparksee.TraversalBFS, 329
sparksee.TraversalDFS, 335
sparksee.WeakConnectivity, 392
sparksee.WeakConnectivityDFS, 398
exclude_nodes
sparksee.CommunitiesSCD, 34
sparksee.CommunityDetection, 39
sparksee.Connectivity, 49
sparksee.Context, 54
sparksee.DisjointCommunityDetection, 89
sparksee.RandomWalk, 219
sparksee.ShortestPath, 242
sparksee.SinglePairShortestPath, 246
sparksee.SinglePairShortestPathBFS, 253
sparksee.SinglePairShortestPathDijkstra, 259
sparksee.StrongConnectivity, 311
sparksee.StrongConnectivityGabow, 316
sparksee.Traversal, 324
sparksee.TraversalBFS, 330
sparksee.TraversalDFS, 336
sparksee.WeakConnectivity, 393
sparksee.WeakConnectivityDFS, 398
execute
sparksee.Query, 209
exists
sparksee.Objects, 191
explode
sparksee.Graph, 127
export
sparksee.Graph, 128
export_all
sparksee.DefaultExport, 75
sparksee.ExportManager, 114

fetch
sparksee.QueryStream, 212
find_attribute
sparksee.Graph, 128
find_attributes
sparksee.Graph, 128
find_edge
sparksee.Graph, 129
find_edge_types
sparksee.Graph, 129
find_node_types
sparksee.Graph, 129
find_object
sparksee.Graph, 130
find_or_create_edge
sparksee.Graph, 130
find_or_create_object
sparksee.Graph, 131
find_type
sparksee.Graph, 131
find_types
sparksee.Graph, 131
FINE
sparksee.LogLevel, 164

generate_schema_script
sparksee.ScriptParser, 235
get
sparksee.ResultSetList, 228
sparksee.SparkseeProperties, 302
sparksee.ValueArray, 372
sparksee.ValueList, 384
get_a_e_s_i_v
sparksee.SparkseeConfig, 280
get_a_e_s_key
sparksee.SparkseeConfig, 281
get_alias
sparksee.Database, 67
get_are_neighbors_indexed
sparksee.Type, 339
get_array_attribute
sparksee.Graph, 132
get_array_size
sparksee.Attribute, 16
get_attribute
sparksee.Graph, 132, 133
get_attribute_interval_count
sparksee.Graph, 133
get_attribute_statistics
sparksee.Graph, 134

get_attribute_text
sparksee.Graph, 134
get_attributes
sparksee.Graph, 135
get_available_mem
sparksee.PlatformStatistics, 207
get_avg_length_string
sparksee.AttributeStatistics, 24
get_boolean
sparksee.SparkseeProperties, 302
sparksee.Value, 361
sparksee.ValueArray, 372
get_boolean_range
sparksee.ValueArray, 372
get_cache
sparksee.DatabaseStatistics, 70
get_cache_max_size
sparksee.Database, 67
sparksee.SparkseeConfig, 281
get_cache_statistics_enabled
sparksee.SparkseeConfig, 281
get_cache_statistics_file
sparksee.SparkseeConfig, 281
get_cache_statistics_snapshot_time
sparksee.SparkseeConfig, 282
get_call_stack_dump
sparksee.SparkseeConfig, 282
get_checksum_enabled
sparksee.SparkseeConfig, 282
get_client_id
sparksee.SparkseeConfig, 282
get_color_rgb
sparksee.EdgeExport, 94
sparksee.GraphExport, 156
sparksee.NodeExport, 168
get_column
sparksee.ResultSet, 224
get_column_data_type
sparksee.ResultSet, 224
get_column_index
sparksee.ResultSet, 224
get_column_name
sparksee.ResultSet, 225
get_communities
sparksee.CommunitiesSCD, 35
sparksee.DisjointCommunityDetection, 89
get_community
sparksee.DisjointCommunities, 84
get_connected_component
sparksee.ConnectedComponents, 45
get_connected_components
sparksee.Connectivity, 49
sparksee.StrongConnectivity, 311
sparksee.StrongConnectivityGabow, 317
sparksee.WeakConnectivity, 393
sparksee.WeakConnectivityDFS, 399
get_cost
sparksee.SinglePairShortestPath, 247
sparksee.SinglePairShortestPathBFS, 253
sparksee.SinglePairShortestPathDijkstra, 260
get_count
sparksee.Attribute, 16
sparksee.ConnectedComponents, 45
sparksee.DisjointCommunities, 84
sparksee.TypeExporterEvent, 345
get_current_depth
sparksee.RandomWalk, 219
sparksee.Traversal, 324
sparksee.TraversalBFS, 330
sparksee.TraversalDFS, 336
get_current_value
sparksee.AttributeListIterator, 22
get_data
sparksee.DatabaseStatistics, 70
get_data_type
sparksee.Attribute, 16
sparksee.Value, 361
get_default_value
sparksee.Attribute, 16
get_distinct
sparksee.AttributeStatistics, 24
get_double
sparksee.Value, 361
sparksee.ValueArray, 373
get_double_range
sparksee.ValueArray, 373
get_download_status
sparksee.SparkseeConfig, 283
get_edge
sparksee.DefaultExport, 75
sparksee.EdgeData, 92
sparksee.ExportManager, 115
get_edge_data
sparksee.Graph, 135
get_edge_peer
sparksee.Graph, 136
get_edge_type
sparksee.DefaultExport, 76
sparksee.ExportManager, 115
get_encryption_enabled
sparksee.SparkseeConfig, 283
get_extent_pages
sparksee.SparkseeConfig, 283
get_extent_size
sparksee.SparkseeConfig, 283
get_font_size
sparksee.EdgeExport, 94
sparksee.NodeExport, 168
get_graph
sparksee.DefaultExport, 76
sparksee.ExportManager, 116
sparksee.Session, 237
get_head
sparksee.EdgeData, 92
get_height
sparksee.NodeExport, 168

get_high_availability_coordinators
sparksee.SparkseeConfig, 284
get_high_availability_enabled
sparksee.SparkseeConfig, 284
get_high_availability_i_p
sparksee.SparkseeConfig, 284
get_high_availability_master_history
sparksee.SparkseeConfig, 284
get_high_availability_synchronization
sparksee.SparkseeConfig, 285
get_id
sparksee.Attribute, 17
sparksee.Type, 339
get_image
sparksee.NodeExport, 168
get_in_mem_alloc_size
sparksee.SparkseeConfig, 285
get_in_memory_pool_capacity
sparksee.Session, 238
sparksee.Sparksee, 268
get_integer
sparksee.SparkseeProperties, 303
sparksee.Value, 361
sparksee.ValueArray, 374
get_integer_range
sparksee.ValueArray, 374
get_is_directed
sparksee.Type, 339
get_is_restricted
sparksee.Type, 339
get_j_s_o_n
sparksee.ResultSet, 225
get_key
sparksee.KeyValue, 160
get_kind
sparksee.Attribute, 17
get_label
sparksee.EdgeExport, 94
sparksee.GraphExport, 156
sparksee.NodeExport, 169
get_labelcolor_rgb
sparksee.EdgeExport, 95
sparksee.NodeExport, 169
get_license
sparksee.SparkseeConfig, 285
get_license_id
sparksee.SparkseeConfig, 285
get_license_pre_download_days
sparksee.SparkseeConfig, 286
get_log_file
sparksee.SparkseeConfig, 286
get_log_level
sparksee.SparkseeConfig, 286
get_long
sparksee.Value, 362
sparksee.ValueArray, 374
get_long_range
sparksee.ValueArray, 375
get_max
sparksee.AttributeStatistics, 24
get_max_length_string
sparksee.AttributeStatistics, 24
get_mean
sparksee.AttributeStatistics, 25
get_median
sparksee.AttributeStatistics, 25
get_min
sparksee.AttributeStatistics, 25
get_min_length_string
sparksee.AttributeStatistics, 26
get_mode
sparksee.AttributeStatistics, 26
get_mode_count
sparksee.AttributeStatistics, 26
get_name
sparksee.Attribute, 17
sparksee.Type, 340
get_node
sparksee.DefaultExport, 77
sparksee.ExportManager, 116
get_node_type
sparksee.DefaultExport, 77
sparksee.ExportManager, 116
get_nodes
sparksee.ConnectedComponents, 46
sparksee.DisjointCommunities, 84
get_null
sparksee.AttributeStatistics, 26
get_num_c_p_us
sparksee.PlatformStatistics, 207
get_num_columns
sparksee.ResultSet, 225
get_num_objects
sparksee.Type, 340
get_object_type
sparksee.Graph, 136
sparksee.Type, 340
get_oid
sparksee.Value, 362
sparksee.ValueArray, 375
get_oid_range
sparksee.ValueArray, 376
get_path
sparksee.Database, 68
get_path_as_edges
sparksee.SinglePairShortestPath, 247
sparksee.SinglePairShortestPathBFS, 253
sparksee.SinglePairShortestPathDijkstra, 260
get_path_as_nodes
sparksee.SinglePairShortestPath, 247
sparksee.SinglePairShortestPathBFS, 254
sparksee.SinglePairShortestPathDijkstra, 260
get_plugins_dir
sparksee.SparkseeConfig, 286
get_pool_frame_size
sparksee.SparkseeConfig, 287

get_pool_partitions
sparksee.SparkseeConfig, 287
get_pool_persistent_max_size
sparksee.SparkseeConfig, 287
get_pool_persistent_min_size
sparksee.SparkseeConfig, 287
get_pool_temporary_max_size
sparksee.SparkseeConfig, 288
get_pool_temporary_min_size
sparksee.SparkseeConfig, 288
get_query_stream
sparksee.QueryStreamManager, 214
get_read
sparksee.DatabaseStatistics, 70
get_real_time
sparksee.PlatformStatistics, 207
get_recovery_cache_max_size
sparksee.SparkseeConfig, 288
get_recovery_checkpoint_time
sparksee.SparkseeConfig, 288
get_recovery_enabled
sparksee.SparkseeConfig, 289
get_recovery_log_file
sparksee.SparkseeConfig, 289
get_restricted_from
sparksee.Type, 340
get_restricted_to
sparksee.Type, 341
get_rollback_enabled
sparksee.SparkseeConfig, 289
get_row
sparksee.CSVReader, 57
sparksee.RowReader, 231
get_session
sparksee.Graph, 136
get_sessions
sparksee.DatabaseStatistics, 71
get_shape
sparksee.NodeExport, 169
get_size
sparksee.Attribute, 17
sparksee.ConnectedComponents, 46
sparksee.DisjointCommunities, 85
get_sparksee_config_file
sparksee.SparkseeConfig, 289
get_statistics
sparksee.Database, 68
sparksee.Platform, 206
get_string
sparksee.Value, 362
get_system_time
sparksee.PlatformStatistics, 208
get_tail
sparksee.EdgeData, 92
get_temp
sparksee.DatabaseStatistics, 71
get_time_unit
sparksee.SparkseeProperties, 303
get_timestamp
sparksee.Value, 362
sparksee.ValueArray, 376
get_timestamp_range
sparksee.ValueArray, 376
get_title
sparksee.EdgeExport, 95
sparksee.NodeExport, 169
get_tmp_enabled
sparksee.SparkseeConfig, 290
get_tmp_folder
sparksee.SparkseeConfig, 290
get_total
sparksee.AttributeStatistics, 27
sparksee.TypeExporterEvent, 345
get_total_mem
sparksee.PlatformStatistics, 208
get_type
sparksee.Graph, 137
get_type_id
sparksee.Attribute, 18
sparksee.TypeExporterEvent, 346
get_user_time
sparksee.PlatformStatistics, 208
get_value
sparksee.KeyValue, 160
get_values
sparksee.Graph, 137
get_variance
sparksee.AttributeStatistics, 27
get_width
sparksee.EdgeExport, 95
sparksee.NodeExport, 170
get_write
sparksee.DatabaseStatistics, 71
GRAPHML
sparksee.ExportType, 119
GRAPHVIZ
sparksee.ExportType, 119
GREATER_EQUAL
sparksee.Condition, 42
GREATER_THAN
sparksee.Condition, 42

has_next
sparksee.AttributeListIterator, 22
sparksee.BooleanListIterator, 30
sparksee.Int32ListIterator, 159
sparksee.KeyValues, 162
sparksee.ObjectsIterator, 195
sparksee.OIDListIterator, 199
sparksee.RandomWalk, 219
sparksee.ResultSetListIterator, 229
sparksee.StringListIterator, 307
sparksee.Traversal, 324
sparksee.TraversalBFS, 330
sparksee.TraversalDFS, 336
sparksee.TypeListIterator, 349
sparksee.ValueListIterator, 385

sparksee.ValuesIterator, 388
heads
sparksee.Graph, 137
HTML
sparksee.ExportType, 119

include_edges
sparksee.CommunitiesSCD, 35
sparksee.CommunityDetection, 39
sparksee.DisjointCommunityDetection, 89
include_nodes
sparksee.CommunitiesSCD, 35
sparksee.CommunityDetection, 40
sparksee.DisjointCommunityDetection, 90
index_attribute
sparksee.Graph, 138
index_neighbors
sparksee.Graph, 138
INFO
sparksee.LogLevel, 165
intersection
sparksee.Objects, 191
is_array_attribute
sparksee.Attribute, 18
is_closed
sparksee.CommunitiesSCD, 36
sparksee.CommunityDetection, 40
sparksee.ConnectedComponents, 46
sparksee.Connectivity, 49
sparksee.Context, 54
sparksee.Database, 68
sparksee.DisjointCommunities, 85
sparksee.DisjointCommunityDetection, 90
sparksee.KeyValues, 162
sparksee.Objects, 191
sparksee.ObjectsIterator, 195
sparksee.Query, 209
sparksee.RandomWalk, 220
sparksee.ResultSet, 226
sparksee.Session, 238
sparksee.ShortestPath, 242
sparksee.SinglePairShortestPath, 248
sparksee.SinglePairShortestPathBFS, 254
sparksee.SinglePairShortestPathDijkstra, 261
sparksee.StrongConnectivity, 311
sparksee.StrongConnectivityGabow, 317
sparksee.Traversal, 325
sparksee.TraversalBFS, 331
sparksee.TraversalDFS, 337
sparksee.Values, 387
sparksee.ValuesIterator, 389
sparksee.WeakConnectivity, 393
sparksee.WeakConnectivityDFS, 399
is_fit
sparksee.NodeExport, 170
is_last
sparksee.TypeExporterEvent, 346
is_null
sparksee.TextStream, 319
sparksee.Value, 363
is_session_attribute
sparksee.Attribute, 18
iterator
sparksee.AttributeList, 21
sparksee.BooleanList, 29
sparksee.Int32List, 158
sparksee.Objects, 192
sparksee.OIDList, 198
sparksee.ResultSetList, 228
sparksee.StringList, 306
sparksee.TypeList, 348
sparksee.ValueList, 384
sparksee.Values, 387
iterator_from_element
sparksee.Objects, 192
iterator_from_index
sparksee.Objects, 192

LESS_EQUAL
sparksee.Condition, 42
LESS_THAN
sparksee.Condition, 42
LIKE
sparksee.Condition, 42
LIKE_NO_CASE
sparksee.Condition, 43
load
sparksee.SparkseeProperties, 304
load_edges_csv
sparksee, 13
load_nodes_csv
sparksee, 14

make_null
sparksee.ValueArray, 377

neighbors
sparksee.Graph, 138, 139
new_array_attribute
sparksee.Graph, 139
new_attribute
sparksee.Graph, 140
new_edge
sparksee.Graph, 141
new_edge_type
sparksee.Graph, 142
new_node
sparksee.Graph, 142
new_node_type
sparksee.Graph, 142
new_objects
sparksee.Session, 238
new_query
sparksee.Session, 238
new_restricted_edge_type
sparksee.Graph, 143
new_session
sparksee.Database, 68

new_session_array_attribute
sparksee.Graph, 143
new_session_attribute
sparksee.Graph, 144
next
sparksee.AttributeListIterator, 22
sparksee.BooleanListIterator, 30
sparksee.Int32ListIterator, 160
sparksee.KeyValues, 162, 163
sparksee.ObjectsIterator, 195
sparksee.OIDListIterator, 199
sparksee.RandomWalk, 220
sparksee.ResultSet, 226
sparksee.ResultSetListIterator, 229
sparksee.StringListIterator, 307
sparksee.Traversal, 325
sparksee.TraversalBFS, 331
sparksee.TraversalDFS, 337
sparksee.TypeListIterator, 350
sparksee.ValueListIterator, 385
sparksee.ValuesIterator, 389
next_edge
sparksee.DefaultExport, 78
sparksee.ExportManager, 117
next_node
sparksee.DefaultExport, 78
sparksee.ExportManager, 117
NOT_EQUAL
sparksee.Condition, 43
notify_event
sparksee.TypeExporterListener, 347
sparksee.TypeLoaderListener, 357

open
sparksee.CSVReader, 58
sparksee.CSVWriter, 63
sparksee.Sparksee, 268
operator
sparksee.Value, 363

parse
sparksee.ScriptParser, 235
pre_commit
sparksee.Session, 239
prepare
sparksee.DefaultExport, 78
sparksee.ExportManager, 117
sparksee.QueryStream, 212

read
sparksee.CSVReader, 58
sparksee.RowReader, 231
sparksee.TextStream, 320
rebuild_database
sparksee.Sparksee, 269
redo_precommitted
sparksee.Database, 69
REG_EXP
sparksee.Condition, 43
register
sparksee.EdgeTypeExporter, 100
sparksee.EdgeTypeLoader, 106
sparksee.NodeTypeExporter, 176
sparksee.NodeTypeLoader, 180
sparksee.TypeExporter, 342
sparksee.TypeLoader, 352
release
sparksee.ExportManager, 118
release_query_stream
sparksee.QueryStreamManager, 214
remove
sparksee.Objects, 193
remove_attribute
sparksee.Graph, 145
remove_checksums
sparksee.Sparksee, 270
remove_type
sparksee.Graph, 145
rename_attribute
sparksee.Graph, 145
rename_type
sparksee.Graph, 146
reset
sparksee.CSVReader, 59
sparksee.RandomWalk, 220
sparksee.RowReader, 232
resize_in_memory_pool
sparksee.Sparksee, 270
restore
sparksee.Sparksee, 271
restore_encrypted_backup
sparksee.Sparksee, 272
run
sparksee.CommunityDetection, 40
sparksee.Connectivity, 50
sparksee.DisjointCommunityDetection, 90
sparksee.EdgeTypeExporter, 100
sparksee.EdgeTypeLoader, 106
sparksee.NodeTypeExporter, 176
sparksee.NodeTypeLoader, 180
sparksee.PageRank, 203
sparksee.ShortestPath, 242
sparksee.SinglePairShortestPath, 248
sparksee.StrongConnectivity, 312
sparksee.TypeExporter, 342
sparksee.TypeLoader, 352
sparksee.WeakConnectivity, 394
run_n_phases
sparksee.EdgeTypeLoader, 107
sparksee.NodeTypeLoader, 181
sparksee.TypeLoader, 352
run_two_phases
sparksee.EdgeTypeLoader, 107
sparksee.NodeTypeLoader, 181
sparksee.TypeLoader, 353

sample
sparksee.Objects, 193

save
sparksee.SparkseeConfig, 290
save_all
sparksee.SparkseeConfig, 290
select
sparksee.Graph, 146–148
set
sparksee.OIDList, 198
sparksee.ValueArray, 377, 378
set_a_e_s_encryption_enabled
sparksee.SparkseeConfig, 291
set_array_attribute
sparksee.Graph, 149
set_array_attribute_void
sparksee.Graph, 149
set_as_directed
sparksee.EdgeExport, 95
set_attribute
sparksee.Graph, 150
set_attribute_default_value
sparksee.Graph, 151
set_attribute_positions
sparksee.EdgeTypeLoader, 108
sparksee.NodeTypeLoader, 182
sparksee.TypeLoader, 353
set_attribute_text
sparksee.Graph, 151
set_attributes
sparksee.EdgeTypeExporter, 100
sparksee.EdgeTypeLoader, 108
sparksee.NodeTypeExporter, 176
sparksee.NodeTypeLoader, 182
sparksee.TypeExporter, 343
sparksee.TypeLoader, 353
set_auto_quotes
sparksee.CSVWriter, 63
set_boolean
sparksee.Value, 363
sparksee.ValueArray, 378
set_boolean_range
sparksee.ValueArray, 378
set_boolean_void
sparksee.Value, 364
set_cache_max_size
sparksee.Database, 69
sparksee.SparkseeConfig, 291
set_cache_statistics_enabled
sparksee.SparkseeConfig, 292
set_cache_statistics_file
sparksee.SparkseeConfig, 292
set_cache_statistics_snapshot_time
sparksee.SparkseeConfig, 292
set_call_stack_dump
sparksee.SparkseeConfig, 293
set_checksum_enabled
sparksee.SparkseeConfig, 293
set_client_id
sparksee.SparkseeConfig, 293
set_color_rgb
sparksee.EdgeExport, 96
sparksee.GraphExport, 156
sparksee.NodeExport, 170
set_damping
sparksee.PageRank, 203
set_default_weight
sparksee.PageRank, 203
sparksee.RandomWalk, 221
set_double
sparksee.Value, 364
sparksee.ValueArray, 379
set_double_range
sparksee.ValueArray, 379
set_double_void
sparksee.Value, 364
set_dynamic
sparksee.Query, 210
set_dynamic_edge_cost_callback
sparksee.SinglePairShortestPathDijkstra, 261
set_edge_weight_attribute_type
sparksee.PageRank, 203
sparksee.RandomWalk, 221
set_error_log
sparksee.ScriptParser, 235
set_extent_pages
sparksee.SparkseeConfig, 293
set_extent_size
sparksee.SparkseeConfig, 294
set_fit
sparksee.NodeExport, 170
set_font_size
sparksee.EdgeExport, 96
sparksee.NodeExport, 171
set_forced_quotes
sparksee.CSVWriter, 64
set_frequency
sparksee.EdgeTypeExporter, 101
sparksee.EdgeTypeLoader, 108
sparksee.NodeTypeExporter, 176
sparksee.NodeTypeLoader, 182
sparksee.TypeExporter, 343
sparksee.TypeLoader, 354
set_graph
sparksee.EdgeTypeExporter, 101
sparksee.EdgeTypeLoader, 108
sparksee.NodeTypeExporter, 177
sparksee.NodeTypeLoader, 182
sparksee.TypeExporter, 343
sparksee.TypeLoader, 354
set_h_i
sparksee.SparkseeProperties, 304
set_head_attribute
sparksee.EdgeTypeExporter, 101
sparksee.EdgeTypeLoader, 109
set_head_mep
sparksee.EdgeTypeLoader, 109
set_head_position

sparksee.EdgeTypeExporter, 102
sparksee.EdgeTypeLoader, 109
set_header
sparksee.EdgeTypeExporter, 102
sparksee.NodeTypeExporter, 177
sparksee.TypeExporter, 344
set_height
sparksee.NodeExport, 171
set_high_availability_coordinators
sparksee.SparkseeConfig, 294
set_high_availability_enabled
sparksee.SparkseeConfig, 294
set_high_availability_i_p
sparksee.SparkseeConfig, 295
set_high_availability_master_history
sparksee.SparkseeConfig, 295
set_high_availability_synchronization
sparksee.SparkseeConfig, 295
set_image
sparksee.NodeExport, 171
set_in_mem_alloc_size
sparksee.SparkseeConfig, 295
set_in_out_parameter
sparksee.RandomWalk, 221
set_initial_page_rank_value
sparksee.PageRank, 204
set_integer
sparksee.Value, 365
sparksee.ValueArray, 379
set_integer_range
sparksee.ValueArray, 380
set_integer_void
sparksee.Value, 365
set_label
sparksee.EdgeExport, 96
sparksee.GraphExport, 157
sparksee.NodeExport, 172
set_labelcolor_rgb
sparksee.EdgeExport, 96
sparksee.NodeExport, 172
set_license
sparksee.SparkseeConfig, 296
set_license_id
sparksee.SparkseeConfig, 296
set_license_pre_download_days
sparksee.SparkseeConfig, 296
set_locale
sparksee.CSVReader, 59
sparksee.CSVWriter, 64
sparksee.EdgeTypeLoader, 110
sparksee.NodeTypeLoader, 183
sparksee.TypeLoader, 354
set_log_error
sparksee.EdgeTypeLoader, 110
sparksee.NodeTypeLoader, 183
sparksee.TypeLoader, 355
set_log_file
sparksee.SparkseeConfig, 297
set_log_level
sparksee.SparkseeConfig, 297
set_log_off
sparksee.EdgeTypeLoader, 110
sparksee.NodeTypeLoader, 183
sparksee.TypeLoader, 355
set_long
sparksee.Value, 365
sparksee.ValueArray, 380
set_long_range
sparksee.ValueArray, 381
set_long_void
sparksee.Value, 366
set_look_ahead
sparksee.CommunitiesSCD, 36
set_materialized_attribute
sparksee.CommunitiesSCD, 36
sparksee.Connectivity, 50
sparksee.DisjointCommunityDetection, 91
sparksee.StrongConnectivity, 312
sparksee.StrongConnectivityGabow, 317
sparksee.WeakConnectivity, 394
sparksee.WeakConnectivityDFS, 399
set_max_array_elements
sparksee.DefaultExport, 79
set_max_text_characters
sparksee.DefaultExport, 79
set_maximum_hops
sparksee.Context, 55
sparksee.RandomWalk, 222
sparksee.ShortestPath, 243
sparksee.SinglePairShortestPath, 248
sparksee.SinglePairShortestPathBFS, 254
sparksee.SinglePairShortestPathDijkstra, 261
sparksee.Traversal, 325
sparksee.TraversalBFS, 331
sparksee.TraversalDFS, 337
set_multilines
sparksee.CSVReader, 59
set_node_type_image
sparksee.DefaultExport, 79
set_node_type_image_attribute
sparksee.DefaultExport, 80
set_node_type_shape
sparksee.DefaultExport, 80
set_null
sparksee.Value, 366
set_num_iterations
sparksee.PageRank, 204
set_num_lines
sparksee.CSVReader, 60
set_oid
sparksee.Value, 366
sparksee.ValueArray, 381
set_oid_range
sparksee.ValueArray, 381
set_oid_void
sparksee.Value, 367

set_output_attribute_type
sparksee.PageRank, 204
set_output_log
sparksee.ScriptParser, 236
set_plugins_dir
sparksee.SparkseeConfig, 297
set_pool_frame_size
sparksee.SparkseeConfig, 297
set_pool_partitions
sparksee.SparkseeConfig, 298
set_pool_persistent_max_size
sparksee.SparkseeConfig, 298
set_pool_persistent_min_size
sparksee.SparkseeConfig, 298
set_pool_temporary_max_size
sparksee.SparkseeConfig, 299
set_pool_temporary_min_size
sparksee.SparkseeConfig, 299
set_quotes
sparksee.CSVReader, 60
sparksee.CSVWriter, 64
set_recovery_cache_max_size
sparksee.SparkseeConfig, 299
set_recovery_checkpoint_time
sparksee.SparkseeConfig, 299
set_recovery_enabled
sparksee.SparkseeConfig, 300
set_recovery_log_file
sparksee.SparkseeConfig, 300
set_return_parameter
sparksee.RandomWalk, 222
set_rollback_enabled
sparksee.SparkseeConfig, 300
set_row_reader
sparksee.EdgeTypeLoader, 111
sparksee.NodeTypeLoader, 184
sparksee.TypeLoader, 355
set_row_writer
sparksee.EdgeTypeExporter, 102
sparksee.NodeTypeExporter, 177
sparksee.TypeExporter, 344
set_seed
sparksee.RandomWalk, 222
set_separator
sparksee.CSVReader, 60
sparksee.CSVWriter, 64
set_shape
sparksee.NodeExport, 172
set_sparksee_config_file
sparksee.SparkseeConfig, 301
set_start_line
sparksee.CSVReader, 61
set_starting_node
sparksee.PageRank, 205
set_stream
sparksee.Query, 210
set_string
sparksee.Value, 367
set_string_void
sparksee.Value, 367
set_tail_attribute
sparksee.EdgeTypeExporter, 103
sparksee.EdgeTypeLoader, 111
set_tail_mep
sparksee.EdgeTypeLoader, 111
set_tail_position
sparksee.EdgeTypeExporter, 103
sparksee.EdgeTypeLoader, 112
set_timestamp
sparksee.Value, 368
sparksee.ValueArray, 382
set_timestamp_format
sparksee.EdgeTypeLoader, 112
sparksee.NodeTypeLoader, 184
sparksee.TypeLoader, 356
set_timestamp_range
sparksee.ValueArray, 382
set_timestamp_void
sparksee.Value, 369
set_title
sparksee.EdgeExport, 97
sparksee.NodeExport, 172
set_title_with_attributes
sparksee.DefaultExport, 80
set_tmp_enabled
sparksee.SparkseeConfig, 301
set_tmp_folder
sparksee.SparkseeConfig, 301
set_tolerance
sparksee.PageRank, 205
set_type
sparksee.EdgeTypeExporter, 103
sparksee.EdgeTypeLoader, 112
sparksee.NodeTypeExporter, 178
sparksee.NodeTypeLoader, 184
sparksee.TypeExporter, 344
sparksee.TypeLoader, 356
set_type_color_rgb
sparksee.DefaultExport, 81
set_type_label_attribute
sparksee.DefaultExport, 81
set_unrecoverable_error_callback
sparksee.Sparksee, 273
set_unweighted_edge_cost
sparksee.SinglePairShortestPathDijkstra, 262
set_void
sparksee.Value, 370
set_width
sparksee.EdgeExport, 97
sparksee.NodeExport, 173
SEVERE
sparksee.LogLevel, 165
size
sparksee.ValueArray, 382
sparksee, 9
load_edges_csv, 13

load_nodes_csv, 14
sparksee.Attribute, 15
get_array_size, 16
get_count, 16
get_data_type, 16
get_default_value, 16
get_id, 17
get_kind, 17
get_name, 17
get_size, 17
get_type_id, 18
is_array_attribute, 18
is_session_attribute, 18
sparksee.AttributeKind, 18
UNIQUE, 19
sparksee.AttributeList, 19
__init__, 20
__iter__, 20
add, 20
count, 21
iterator, 21
sparksee.AttributeListIterator, 21
__next__, 22
get_current_value, 22
has_next, 22
next, 22
sparksee.AttributeStatistics, 23
get_avg_length_string, 24
get_distinct, 24
get_max, 24
get_max_length_string, 24
get_mean, 25
get_median, 25
get_min, 25
get_min_length_string, 26
get_mode, 26
get_mode_count, 26
get_null, 26
get_total, 27
get_variance, 27
sparksee.BooleanList, 27
__init__, 28
__iter__, 28
add, 28
count, 28
iterator, 29
sparksee.BooleanListIterator, 29
__next__, 30
has_next, 30
next, 30
sparksee.CommunitiesSCD, 31
__init__, 33
add_all_edge_types, 33
add_edge_type, 33
add_node_type, 33
close, 34
exclude_edges, 34
exclude_nodes, 34
get_communities, 35
include_edges, 35
include_nodes, 35
is_closed, 36
set_look_ahead, 36
set_materialized_attribute, 36
sparksee.CommunityDetection, 37
add_node_type, 38
close, 38
exclude_edges, 39
exclude_nodes, 39
include_edges, 39
include_nodes, 40
is_closed, 40
run, 40
sparksee.Condition, 41
BETWEEN, 42
EQUAL, 42
GREATER_EQUAL, 42
GREATER_THAN, 42
LESS_EQUAL, 42
LESS_THAN, 42
LIKE, 42
LIKE_NO_CASE, 43
NOT_EQUAL, 43
REG_EXP, 43
sparksee.ConnectedComponents, 44
__init__, 45
close, 45
get_connected_component, 45
get_count, 45
get_nodes, 46
get_size, 46
is_closed, 46
sparksee.Connectivity, 47
add_node_type, 48
close, 48
exclude_edges, 48
exclude_nodes, 49
get_connected_components, 49
is_closed, 49
run, 50
set_materialized_attribute, 50
sparksee.Context, 51
__init__, 52
add_all_edge_types, 52
add_edge_type, 52
add_node_type, 52
close, 53
compute, 53
exclude_edges, 54
exclude_nodes, 54
is_closed, 54
set_maximum_hops, 55
sparksee.CSVReader, 55
close, 57
get_row, 57
open, 58

read, 58
reset, 59
set_locale, 59
set_multilines, 59
set_num_lines, 60
set_quotes, 60
set_separator, 60
set_start_line, 61
sparksee.CSVWriter, 61
close, 63
open, 63
set_auto_quotes, 63
set_forced_quotes, 64
set_locale, 64
set_quotes, 64
set_separator, 64
write, 65
sparksee.Database, 65
add_query_stream_manager, 67
close, 67
dump_schema, 67
get_alias, 67
get_cache_max_size, 67
get_path, 68
get_statistics, 68
is_closed, 68
new_session, 68
redo_precommitted, 69
set_cache_max_size, 69
sparksee.DatabaseStatistics, 69
get_cache, 70
get_data, 70
get_read, 70
get_sessions, 71
get_temp, 71
get_write, 71
sparksee.DataType, 71
STRING, 72
TEXT, 72
TIMESTAMP, 72
sparksee.DefaultExport, 73
enable_type, 75
export_all, 75
get_edge, 75
get_edge_type, 76
get_graph, 76
get_node, 77
get_node_type, 77
next_edge, 78
next_node, 78
prepare, 78
set_max_array_elements, 79
set_max_text_characters, 79
set_node_type_image, 79
set_node_type_image_attribute, 80
set_node_type_shape, 80
set_title_with_attributes, 80
set_type_color_rgb, 81
set_type_label_attribute, 81
subgraph_edges, 81
subgraph_nodes, 82
sparksee.DisjointCommunities, 82
__init__, 83
close, 84
get_community, 84
get_count, 84
get_nodes, 84
get_size, 85
is_closed, 85
sparksee.DisjointCommunityDetection, 86
add_all_edge_types, 87
add_edge_type, 87
add_node_type, 88
close, 88
exclude_edges, 88
exclude_nodes, 89
get_communities, 89
include_edges, 89
include_nodes, 90
is_closed, 90
run, 90
set_materialized_attribute, 91
sparksee.EdgeData, 91
get_edge, 92
get_head, 92
get_tail, 92
sparksee.EdgeExport, 92
as_directed, 94
copy_defaults, 94
get_color_rgb, 94
get_font_size, 94
get_label, 94
get_labelcolor_rgb, 95
get_title, 95
get_width, 95
set_as_directed, 95
set_color_rgb, 96
set_font_size, 96
set_label, 96
set_labelcolor_rgb, 96
set_title, 97
set_width, 97
sparksee.EdgesDirection, 97
sparksee.EdgeTypeExporter, 98
__init__, 99
register, 100
run, 100
set_attributes, 100
set_frequency, 101
set_graph, 101
set_head_attribute, 101
set_head_position, 102
set_header, 102
set_row_writer, 102
set_tail_attribute, 103
set_tail_position, 103

set_type, 103
sparksee.EdgeTypeLoader, 104
__init__, 106
register, 106
run, 106
run_n_phases, 107
run_two_phases, 107
set_attribute_positions, 108
set_attributes, 108
set_frequency, 108
set_graph, 108
set_head_attribute, 109
set_head_mep, 109
set_head_position, 109
set_locale, 110
set_log_error, 110
set_log_off, 110
set_row_reader, 111
set_tail_attribute, 111
set_tail_mep, 111
set_tail_position, 112
set_timestamp_format, 112
set_type, 112
sparksee.ExportManager, 113
enable_type, 114
export_all, 114
get_edge, 115
get_edge_type, 115
get_graph, 116
get_node, 116
get_node_type, 116
next_edge, 117
next_node, 117
prepare, 117
release, 118
sparksee.ExportType, 118
GRAPHML, 119
GRAPHVIZ, 119
HTML, 119
YGRAPHML, 119
sparksee.Graph, 119
backup, 123
count_edges, 123
count_nodes, 124
degree, 124
drop, 124, 125
dump_data, 125
dump_storage, 125
edges, 126
encrypted_backup, 126
explode, 127
export, 128
find_attribute, 128
find_attributes, 128
find_edge, 129
find_edge_types, 129
find_node_types, 129
find_object, 130
find_or_create_edge, 130
find_or_create_object, 131
find_type, 131
find_types, 131
get_array_attribute, 132
get_attribute, 132, 133
get_attribute_interval_count, 133
get_attribute_statistics, 134
get_attribute_text, 134
get_attributes, 135
get_edge_data, 135
get_edge_peer, 136
get_object_type, 136
get_session, 136
get_type, 137
get_values, 137
heads, 137
index_attribute, 138
index_neighbors, 138
neighbors, 138, 139
new_array_attribute, 139
new_attribute, 140
new_edge, 141
new_edge_type, 142
new_node, 142
new_node_type, 142
new_restricted_edge_type, 143
new_session_array_attribute, 143
new_session_attribute, 144
remove_attribute, 145
remove_type, 145
rename_attribute, 145
rename_type, 146
select, 146–148
set_array_attribute, 149
set_array_attribute_void, 149
set_attribute, 150
set_attribute_default_value, 151
set_attribute_text, 151
tails, 151
tails_and_heads, 152
top_k, 152–155
sparksee.GraphExport, 155
get_color_rgb, 156
get_label, 156
set_color_rgb, 156
set_label, 157
sparksee.Int32List, 157
__init__, 158
__iter__, 158
add, 158
count, 158
iterator, 158
sparksee.Int32ListIterator, 159
__next__, 159
has_next, 159
next, 160
sparksee.KeyValue, 160

get_key, 160
get_value, 160
sparksee.KeyValues, 161
__next__, 162
close, 162
has_next, 162
is_closed, 162
next, 162, 163
sparksee.KOpt, 163
sparksee.LogLevel, 163
CONFIG, 164
DEBUG, 164
FINE, 164
INFO, 165
SEVERE, 165
WARNING, 165
sparksee.MissingEndpoint, 165
sparksee.NodeExport, 166
copy_defaults, 168
get_color_rgb, 168
get_font_size, 168
get_height, 168
get_image, 168
get_label, 169
get_labelcolor_rgb, 169
get_shape, 169
get_title, 169
get_width, 170
is_fit, 170
set_color_rgb, 170
set_fit, 170
set_font_size, 171
set_height, 171
set_image, 171
set_label, 172
set_labelcolor_rgb, 172
set_shape, 172
set_title, 172
set_width, 173
sparksee.NodeShape, 173
sparksee.NodeTypeExporter, 174
__init__, 175
register, 176
run, 176
set_attributes, 176
set_frequency, 176
set_graph, 177
set_header, 177
set_row_writer, 177
set_type, 178
sparksee.NodeTypeLoader, 178
__init__, 180
register, 180
run, 180
run_n_phases, 181
run_two_phases, 181
set_attribute_positions, 182
set_attributes, 182
set_frequency, 182
set_graph, 182
set_locale, 183
set_log_error, 183
set_log_off, 183
set_row_reader, 184
set_timestamp_format, 184
set_type, 184
sparksee.Objects, 185
__iter__, 186
add, 186
any, 187
close, 187
combine_difference, 187
combine_intersection, 188
combine_union, 188
contains, 189
copy, 189
count, 190
difference, 190
equals, 190
exists, 191
intersection, 191
is_closed, 191
iterator, 192
iterator_from_element, 192
iterator_from_index, 192
remove, 193
sample, 193
union, 193
sparksee.ObjectsIterator, 194
__next__, 195
close, 195
has_next, 195
is_closed, 195
next, 195
sparksee.ObjectType, 196
sparksee.OIDList, 196
__init__, 197
__iter__, 197
add, 197
count, 198
iterator, 198
set, 198
sparksee.OIDListIterator, 199
__next__, 199
has_next, 199
next, 199
sparksee.Order, 200
sparksee.PageRank, 200
__init__, 201
add_all_edge_types, 202
add_edge_type, 202
add_node_type, 202
run, 203
set_damping, 203
set_default_weight, 203
set_edge_weight_attribute_type, 203

set_initial_page_rank_value, 204
set_num_iterations, 204
set_output_attribute_type, 204
set_starting_node, 205
set_tolerance, 205
sparksee.Platform, 206
get_statistics, 206
sparksee.PlatformStatistics, 206
get_available_mem, 207
get_num_c_p_us, 207
get_real_time, 207
get_system_time, 208
get_total_mem, 208
get_user_time, 208
sparksee.Query, 208
close, 209
execute, 209
is_closed, 209
set_dynamic, 210
set_stream, 210
sparksee.QueryContext, 211
sparksee.QueryLanguage, 211
sparksee.QueryStream, 211
fetch, 212
prepare, 212
start, 213
sparksee.QueryStreamManager, 213
get_query_stream, 214
release_query_stream, 214
sparksee.RandomWalk, 215
__init__, 217
__iter__, 217
__next__, 217
add_all_edge_types, 217
add_edge_type, 218
add_node_type, 218
close, 218
exclude_edges, 219
exclude_nodes, 219
get_current_depth, 219
has_next, 219
is_closed, 220
next, 220
reset, 220
set_default_weight, 221
set_edge_weight_attribute_type, 221
set_in_out_parameter, 221
set_maximum_hops, 222
set_return_parameter, 222
set_seed, 222
sparksee.ResultSet, 223
get_column, 224
get_column_data_type, 224
get_column_index, 224
get_column_name, 225
get_j_s_o_n, 225
get_num_columns, 225
is_closed, 226
next, 226
sparksee.ResultSetList, 226
__init__, 227
__iter__, 227
count, 227
get, 228
iterator, 228
sparksee.ResultSetListIterator, 228
__next__, 229
has_next, 229
next, 229
sparksee.RowReader, 230
close, 231
get_row, 231
read, 231
reset, 232
sparksee.RowWriter, 232
close, 233
write, 233
sparksee.ScriptParser, 234
generate_schema_script, 235
parse, 235
set_error_log, 235
set_output_log, 236
sparksee.Session, 236
close, 237
get_graph, 237
get_in_memory_pool_capacity, 238
is_closed, 238
new_objects, 238
new_query, 238
pre_commit, 239
sparksee.ShortestPath, 239
add_all_edge_types, 240
add_edge_type, 241
add_node_type, 241
close, 241
exclude_edges, 241
exclude_nodes, 242
is_closed, 242
run, 242
set_maximum_hops, 243
sparksee.SinglePairShortestPath, 243
add_all_edge_types, 245
add_edge_type, 245
add_node_type, 245
close, 246
exclude_edges, 246
exclude_nodes, 246
get_cost, 247
get_path_as_edges, 247
get_path_as_nodes, 247
is_closed, 248
run, 248
set_maximum_hops, 248
sparksee.SinglePairShortestPathBFS, 249
__init__, 251
add_all_edge_types, 251

add_edge_type, 252
add_node_type, 252
check_only_existence, 252
close, 252
exclude_edges, 253
exclude_nodes, 253
get_cost, 253
get_path_as_edges, 253
get_path_as_nodes, 254
is_closed, 254
set_maximum_hops, 254
sparksee.SinglePairShortestPathDijkstra, 255
__init__, 257
add_all_edge_types, 258
add_edge_type, 258
add_node_type, 258
add_weighted_edge_type, 258
close, 259
exclude_edges, 259
exclude_nodes, 259
get_cost, 260
get_path_as_edges, 260
get_path_as_nodes, 260
is_closed, 261
set_dynamic_edge_cost_callback, 261
set_maximum_hops, 261
set_unweighted_edge_cost, 262
sparksee.SinglePairShortestPathDijkstraDynamicCost, 262
calculate_edge_cost, 263
sparksee.Sparksee, 263
__init__, 265
add_checksums, 265
close, 265
create, 265, 266
decrypt, 267
encrypt, 267
get_in_memory_pool_capacity, 268
open, 268
rebuild_database, 269
remove_checksums, 270
resize_in_memory_pool, 270
restore, 271
restore_encrypted_backup, 272
set_unrecoverable_error_callback, 273
verify_checksums, 273
sparksee.SparkseeConfig, 274
__init__, 279, 280
download_license, 280
get_a_e_s_i_v, 280
get_a_e_s_key, 281
get_cache_max_size, 281
get_cache_statistics_enabled, 281
get_cache_statistics_file, 281
get_cache_statistics_snapshot_time, 282
get_call_stack_dump, 282
get_checksum_enabled, 282
get_client_id, 282
get_download_status, 283
get_encryption_enabled, 283
get_extent_pages, 283
get_extent_size, 283
get_high_availability_coordinators, 284
get_high_availability_enabled, 284
get_high_availability_i_p, 284
get_high_availability_master_history, 284
get_high_availability_synchronization, 285
get_in_mem_alloc_size, 285
get_license, 285
get_license_id, 285
get_license_pre_download_days, 286
get_log_file, 286
get_log_level, 286
get_plugins_dir, 286
get_pool_frame_size, 287
get_pool_partitions, 287
get_pool_persistent_max_size, 287
get_pool_persistent_min_size, 287
get_pool_temporary_max_size, 288
get_pool_temporary_min_size, 288
get_recovery_cache_max_size, 288
get_recovery_checkpoint_time, 288
get_recovery_enabled, 289
get_recovery_log_file, 289
get_rollback_enabled, 289
get_sparksee_config_file, 289
get_tmp_enabled, 290
get_tmp_folder, 290
save, 290
save_all, 290
set_a_e_s_encryption_enabled, 291
set_cache_max_size, 291
set_cache_statistics_enabled, 292
set_cache_statistics_file, 292
set_cache_statistics_snapshot_time, 292
set_call_stack_dump, 293
set_checksum_enabled, 293
set_client_id, 293
set_extent_pages, 293
set_extent_size, 294
set_high_availability_coordinators, 294
set_high_availability_enabled, 294
set_high_availability_i_p, 295
set_high_availability_master_history, 295
set_high_availability_synchronization, 295
set_in_mem_alloc_size, 295
set_license, 296
set_license_id, 296
set_license_pre_download_days, 296
set_log_file, 297
set_log_level, 297
set_plugins_dir, 297
set_pool_frame_size, 297
set_pool_partitions, 298
set_pool_persistent_max_size, 298
set_pool_persistent_min_size, 298

set_pool_temporary_max_size, 299
set_pool_temporary_min_size, 299
set_recovery_cache_max_size, 299
set_recovery_checkpoint_time, 299
set_recovery_enabled, 300
set_recovery_log_file, 300
set_rollback_enabled, 300
set_sparksee_config_file, 301
set_tmp_enabled, 301
set_tmp_folder, 301
sparksee.SparkseeProperties, 301
get, 302
get_boolean, 302
get_integer, 303
get_time_unit, 303
load, 304
set_h_i, 304
sparksee.StringList, 304
__init__, 305
__iter__, 305
add, 305
count, 306
iterator, 306
sparksee.StringListIterator, 306
__next__, 307
has_next, 307
next, 307
sparksee.StrongConnectivity, 308
add_all_edge_types, 309
add_edge_type, 309
add_node_type, 310
close, 310
exclude_edges, 310
exclude_nodes, 311
get_connected_components, 311
is_closed, 311
run, 312
set_materialized_attribute, 312
sparksee.StrongConnectivityGabow, 313
__init__, 315
add_all_edge_types, 315
add_edge_type, 315
add_node_type, 315
close, 316
exclude_edges, 316
exclude_nodes, 316
get_connected_components, 317
is_closed, 317
set_materialized_attribute, 317
sparksee.TextStream, 318
__init__, 319
close, 319
is_null, 319
read, 320
write, 320
sparksee.Traversal, 321
__iter__, 322
__next__, 322
add_all_edge_types, 322
add_edge_type, 323
add_node_type, 323
close, 323
exclude_edges, 323
exclude_nodes, 324
get_current_depth, 324
has_next, 324
is_closed, 325
next, 325
set_maximum_hops, 325
sparksee.TraversalBFS, 326
__init__, 328
__iter__, 328
__next__, 328
add_all_edge_types, 328
add_edge_type, 329
add_node_type, 329
close, 329
exclude_edges, 329
exclude_nodes, 330
get_current_depth, 330
has_next, 330
is_closed, 331
next, 331
set_maximum_hops, 331
sparksee.TraversalDFS, 332
__init__, 334
__iter__, 334
__next__, 334
add_all_edge_types, 334
add_edge_type, 335
add_node_type, 335
close, 335
exclude_edges, 335
exclude_nodes, 336
get_current_depth, 336
has_next, 336
is_closed, 337
next, 337
set_maximum_hops, 337
sparksee.Type, 338
get_are_neighbors_indexed, 339
get_id, 339
get_is_directed, 339
get_is_restricted, 339
get_name, 340
get_num_objects, 340
get_object_type, 340
get_restricted_from, 340
get_restricted_to, 341
sparksee.TypeExporter, 341
register, 342
run, 342
set_attributes, 343
set_frequency, 343
set_graph, 343
set_header, 344

set_row_writer, 344
set_type, 344
sparksee.TypeExporterEvent, 345
get_count, 345
get_total, 345
get_type_id, 346
is_last, 346
sparksee.TypeExporterListener, 346
notify_event, 347
sparksee.TypeList, 347
__init__, 348
__iter__, 348
add, 348
count, 348
iterator, 348
sparksee.TypeListIterator, 349
__next__, 349
has_next, 349
next, 350
sparksee.TypeLoader, 350
register, 352
run, 352
run_n_phases, 352
run_two_phases, 353
set_attribute_positions, 353
set_attributes, 353
set_frequency, 354
set_graph, 354
set_locale, 354
set_log_error, 355
set_log_off, 355
set_row_reader, 355
set_timestamp_format, 356
set_type, 356
sparksee.TypeLoaderEvent, 356
sparksee.TypeLoaderListener, 357
notify_event, 357
sparksee.UnrecoverableError, 357
sparksee.Value, 358
__init__, 360
compare, 360
equals, 360
get_boolean, 361
get_data_type, 361
get_double, 361
get_integer, 361
get_long, 362
get_oid, 362
get_string, 362
get_timestamp, 362
is_null, 363
operator, 363
set_boolean, 363
set_boolean_void, 364
set_double, 364
set_double_void, 364
set_integer, 365
set_integer_void, 365
set_long, 365
set_long_void, 366
set_null, 366
set_oid, 366
set_oid_void, 367
set_string, 367
set_string_void, 367
set_timestamp, 368
set_timestamp_void, 369
set_void, 370
sparksee.ValueArray, 370
get, 372
get_boolean, 372
get_boolean_range, 372
get_double, 373
get_double_range, 373
get_integer, 374
get_integer_range, 374
get_long, 374
get_long_range, 375
get_oid, 375
get_oid_range, 376
get_timestamp, 376
get_timestamp_range, 376
make_null, 377
set, 377, 378
set_boolean, 378
set_boolean_range, 378
set_double, 379
set_double_range, 379
set_integer, 379
set_integer_range, 380
set_long, 380
set_long_range, 381
set_oid, 381
set_oid_range, 381
set_timestamp, 382
set_timestamp_range, 382
size, 382
sparksee.ValueList, 383
__init__, 383
__iter__, 384
add, 384
count, 384
get, 384
iterator, 384
sparksee.ValueListIterator, 385
__next__, 385
has_next, 385
next, 385
sparksee.Values, 386
__iter__, 386
close, 386
count, 387
is_closed, 387
iterator, 387
sparksee.ValuesIterator, 388
__next__, 388

close, 388
has_next, 388
is_closed, 389
next, 389
sparksee.WeakConnectivity, 390
add_all_edge_types, 391
add_edge_type, 391
add_node_type, 392
close, 392
exclude_edges, 392
exclude_nodes, 393
get_connected_components, 393
is_closed, 393
run, 394
set_materialized_attribute, 394
sparksee.WeakConnectivityDFS, 395
__init__, 397
add_all_edge_types, 397
add_edge_type, 397
add_node_type, 397
close, 398
exclude_edges, 398
exclude_nodes, 398
get_connected_components, 399
is_closed, 399
set_materialized_attribute, 399
start
sparksee.QueryStream, 213
STRING
sparksee.DataType, 72
subgraph_edges
sparksee.DefaultExport, 81
subgraph_nodes
sparksee.DefaultExport, 82

tails
sparksee.Graph, 151
tails_and_heads
sparksee.Graph, 152
TEXT
sparksee.DataType, 72
TIMESTAMP
sparksee.DataType, 72
top_k
sparksee.Graph, 152–155

union
sparksee.Objects, 193
UNIQUE
sparksee.AttributeKind, 19

verify_checksums
sparksee.Sparksee, 273

WARNING
sparksee.LogLevel, 165
write
sparksee.CSVWriter, 65
sparksee.RowWriter, 233
sparksee.TextStream, 320

YGRAPHML
sparksee.ExportType, 119