```
<<Interface>>
MonitoringServiceInterface
(Implementation Model::juniper-sa-deployment-monitor::eu.juniper.sa.deployment.monitor)
--------------------------------------------------------------------------------
<<Property>> +monitoringServiceURL : String
<<Property>> +applicationId : String
--------------------------------------------------------------------------------
+getApplicationDetails() : String
+getMetricsNames() : String []
+getMetricValues(metricName : String) : String []
+getMetricValues(metricName : String, conditionName : String, conditionValue : String) : String []
+getMetricAggregated(metricName : String, fromTimestamp : long, toTimestamp : long) : AggregatedM...
```

```
MonitoringDbService
(Implementation Model::juniper-sa-deployment-monitor::eu.juniper.sa.deployment.monitor)
--------------------------------------------------------------------------------
<<Property>> -databaseConnection : Connection
<<Property>> -monitoringServiceURL : String
<<Property>> -applicationId : String
-JDBC_DRIVER_H2 : String = "org.h2.Driver"
-SQL_CREATE_TABLE_METRICS : String = "CREATE TABLE IF NOT EXISTS metrics (recordid int NOT NULL, name varchar(32) NOT NULL, numericvalue double, textvalue varchar(32), PRIMARY KEY (recordid, name), CONSTRAINT not_null_metric_value CHECK (numericvalue IS NOT NULL OR textvalue IS NOT ...
-SQL_CREATE_TABLE_RECORDS : String = "CREATE TABLE IF NOT EXISTS records (id int GENERATED BY DEFAULT AS IDENTITY, time timestamp NOT NULL, metrictype varchar(32), hostname varchar_ignorecase(32) NOT NULL, PRIMARY KEY (id));"
-SQL_CREATE_CONSTRAINT_RECORD_METRICS : String = "ALTER TABLE metrics ADD CONSTRAINT IF NOT EXISTS record_has_recorded_metrics FOREIGN KEY (recordid) REFERENCES records (id) ON UPDATE Cascade ON DELETE Cascade;"
-SQL_CREATE_ALIAS_SECONDS : String = "CREATE ALIAS IF NOT EXISTS seconds DETERMINISTIC AS $$ long seconds(Timestamp timestamp) { return timestamp.getTime() / 1000; } $$;"
-SQL_DROP_ALIAS_SECONDS : String = "DROP ALIAS IF EXISTS seconds;"
-SQL_DROP_CONSTRAINT_RECORD_METRICS : String = "ALTER TABLE metrics DROP CONSTRAINT IF EXISTS record_has_recorded_metrics;"
-SQL_DROP_TABLE_METRICS : String = "DROP TABLE IF EXISTS metrics;"
-SQL_DROP_TABLE_RECORDS : String = "DROP TABLE IF EXISTS records;"
-SQL_DROP_ALL_AND_DELETE : String = "DROP ALL OBJECTS DELETE FILES;"
-SQL_INSERT_RECORDS : String = "INSERT INTO records(time, metrictype, hostname) VALUES (?, ?, ?);"
-SQL_INSERT_METRICS_NUMERIC : String = "INSERT INTO metrics(recordid, name, numericvalue) VALUES (?, ?, ?);"
-SQL_INSERT_METRICS_TEXT : String = "INSERT INTO metrics(recordid, name, textvalue) VALUES (?, ?, ?);"
--------------------------------------------------------------------------------
+MonitoringDbService(monitoringServiceURL : String, applicationId : String, dbURL : String, dbDriverClassName : String)
+MonitoringDbService(monitoringServiceURL : String, applicationId : String, dbURL : String)
+MonitoringDbService(monitoringServiceURL : String, applicationId : String)
+getApplications(monitoringServiceURL : String) : String []
+close() : void
+closeDatabase() : void
+createDatabaseTables() : void
+dropDatabaseTables() : void
+exportDatabase(exportSqlScriptFilename : String) : void
+importDatabase(importSqlScriptFilename : String) : void
+importMetrics() : int
+getApplicationDetails() : String
+getMetricsNames() : String []
+getMetricValues(metricName : String) : String []
+getMetricValues(metricName : String, conditionName : String, conditionValue : String) : String []
+getMetricAggregated(metricName : String, fromTimestamp : long, toTimestamp : long) : AggregatedMetric
+main(args : String []) : void
+deleteDatabase() : void
```

```
MonitoringService
(Implementation Model::juniper-sa-deployment-monitor::eu.juniper.sa.deployment.monitor)
--------------------------------------------------------------------------------
-monitoringLib : MonitoringLib
<<Property>> -monitoringServiceURL : String
<<Property>> -applicationId : String
--------------------------------------------------------------------------------
+MonitoringService(monitoringServiceURL : String, applicationId : String)
#getMonitoringLib() : MonitoringLib
+getApplications(monitoringServiceURL : String) : String []
+getApplicationDetails() : String
+getMetricsNames() : String []
#getMetricAggregated(metricName : String, timeInterval : String) : AggregatedMetric
+getMetricAggregated(metricName : String, fromTimestamp : long, toTimestamp : long) : AggregatedM...
+getMetricValues(metricName : String) : String []
+getMetricValues(metricName : String, conditionName : String, conditionValue : String) : String []
+main(args : String []) : void
```

```
AggregatedMetric
(Implementation Model::juniper-sa-deployment-monitor::eu.juniper.sa.deployment.monitor)
--------------------------------------------------------------------------------
+count : Integer
+min : Double
+max : Double
+avg : Double
+sum : Double
+sumOfSquares : Double
+variance : Double
+stdDeviation : Double
--------------------------------------------------------------------------------
+AggregatedMetric(count : Integer, min : Double, max : Double, avg : Double, sum : Double, sumOfSquares : Double, variance : Double, stdDeviation : D...
+AggregatedMetric(jsonRepresentation : String)
+toString() : String
```

```
MonitoringDbServer
(Implementation Model::juniper-sa-deployment-monitor::eu.juniper.sa.deployment.monitor)
--------------------------------------------------------------------------------
+main(args : String []) : void
```