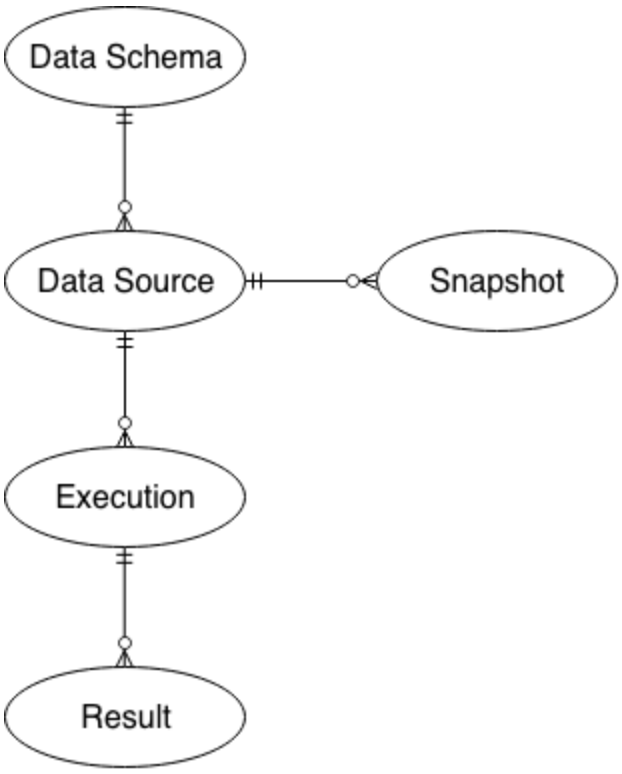
Responder Orchestration Server - Rest Services

Inventory

The following table lists all resources provided by the Responder Server to be consumed by the Secure Enclave.

All Services use XML format with vendor specific Media Type: *application/vnd.encrypedquery.responder+xml*

Resource	Description	URI
Root	List of root resources and their URIs. This can be used by the client to avoid hard coding the various resource URIs.	/responder/api/rest/
Snapshot	State of all resources (including children) starting at Data Schemas specific point in time. (Does not include Result's payloads)	/responder/api/rest/snapshots
Data Schema	Available Data Schemas.	/responder/api/rest/dataschemas
Data Source	Data Sources supporting the given Data Schema.	/responder/api/rest/dataschemas/{dataSchemaId}/datasources
Execution	Executions for a given Data Source. Queries can be executed on Data Sources. Each Data Source may have multiple query executions.	/responder/api/rest/dataschemas/{dataSchemaId}/datasources/{datasourceId}/executions
Result	Executions produce Results. Some executions, for example Streaming Data Sources can produce more than one Result.	/responder/api/rest/dataschemas/{dataSchemaId}/datasources/{datasourceId}/executions/{executionId}/results



Versioning

In order to support versioning, a vendor specific media type is used throughout this API.

Media Type: *application/vnd.encrypedquery.responder+xml*

This media type also defines a version parameter: *version*={XXX}. Clients can then specify the version of the resource they request in the standard HTTP header *Accept.*

Example:

```
GET /responder/api/rest/datasources/42 HTTP/1.1
Accept: application/vnd.encrypedquery.responder+xml; version=1
```

The server response will always contain the version of the resource:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.encrypedquery.responder+xml; version=1
```

If the version parameter is not specified by the client, the server will return the latest supported version of the resource.

Detailed Description of the Resources

Root Resources

Address: /responder/api/rest

Verbs: GET

Example:

```
GET /responder/api/rest HTTP/1.1
Accept: application/vnd.encrypedquery.responder+xml; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encrypedquery.responder+xml; version=1

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:resources xmlns="http://enquery.net/encryptedquery/resources">
    <resource>
        <id>datasource</id>
        <selfUri>/responder/api/rest/datasources</selfUri>
    </resource>
 <resource>
  <id>snapshot</id>
  <selfUri>/responder/api/rest/snapshots</selfUri>
 </resource>
</ns2:resources>
```

Snapshot Resource

Address: /responder/api/rest/snapshots/{snapshotId}

Verbs: GET, POST

Parameters:

Name	Type	Data Type	Description	Optional	Scope	Effect
snapshotId	path	int	Id of the Snapshot	true	GET	If present, only the snapshot with this Id is returned.

Data:

Member	Data Type	Description	Scope
id	int	Id of this Snapshot	GET
selfUri	uri	URI to this Snapshot Resource	GET

Example:

Create a new Snapshot

```
POST /responder/api/rest/snapshots HTTP/1.1
Accept: application/vnd.encrypedquery.responder+xml; version=1

HTTP/1.1 201 Created
Content-Type: application/vnd.encrypedquery.responder+xml; version=1

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<snapshotResource
xmlns="http://enquery.net/encryptedquery/snapshot/resource">
 <id>1010</id>
 <selfUri>/responder/api/rest/snapshots/1010</selfUri>
<snapshotResource>
```

List existing snapshots

```
GET /responder/api/rest/snapshots HTTP/1.1
Accept: application/vnd.encrypedquery.responder+xml; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encrypedquery.responder+xml; version=1

<ns2:snapshotsResources
 xmlns:ns2="http://enquery.net/encryptedquery/snapshot"
 xmlns="http://enquery.net/encryptedquery/snapshot/resource">

 <snapshotResource>
  <id>1010</id>
  <selfUri>/responder/api/rest/snapshots/1010</selfUri>
 </snapshotResource>

 <snapshotResource>
  <id>1011</id>
  <selfUri>/responder/api/rest/snapshots/1011</selfUri>
 </snapshotResource>

 <pagination>

<previousUri>/responder/api/rest/snapshots?before=MTAxNTZ&limit=10</prev
iousUri>
  <nextUri>/responder/api/rest/snapshots?after=NDMy&limit=10</nextUri>
 </pagination>

</ns2:snapshotsResources>
```

Retrieve a specific snapshot

```
GET /responder/api/rest/snapshots/1010  HTTP/1.1
Accept: application/vnd.encrypedquery.responder+xml; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encrypedquery.responder+xml; version=1

<snapshotResource>
 <id>1010</id>
 <selfUri>/responder/api/rest/snapshots/1010</selfUri>
 <dataSchemas>
  <dataSchema>
   <id>1010</id>
   <selfUri>/responder/api/rest/dataschemas/1010</selfUri>
   <name>Phone Record</name>
   <field>
    <name>action</name>
    <dataType>string</dataType>
    <isArray>false</isArray>
    <position>0</position>
   </field>
   <field>
    <name>callee</name>
    <dataType>string</dataType>
    <isArray>false</isArray>
    <position>5</position>
   </field>
   <dataSources>
    <dataSource>
     <id>1020</id>

<selfUri>/responder/api/rest/dataschemas/1010/datasources/1020</selfUri>

     <name>Flink-Kafka-Phone-Record</name>
     <description>Flink Engine running on a Kafka Phone Record
Stream</description>
     <type>streaming</batch>
     <executions>
      <execution>
       <id>1030</id>

<selfUri>/responder/api/rest/dataschemas/1010/datasources/1020/execution
s/1030</selfUri>
       <submittedOn>2018-04-23T18:25:43.511Z</submittedOn>
       <startedOn>2018-04-23T19:00:00.000Z</startedOn>
       <completedOn>2018-05-23T6:00:00.000Z</completedOn>
       <results>
```

```
         <result>
          <id>1040</id>

<selfUri>/responder/api/rest/dataschemas/1010/datasources/1020/execution
s/1030/results/1040</selfUri>
          <createdOn>2018-04-23T18:25:43.511Z</createdOn>
         </result>
         <result>
          <id>1041</id>

<selfUri>/responder/api/rest/dataschemas/1010/datasources/1020/execution
s/1030/results/1041</selfUri>
          <createdOn>2018-04-24T18:25:43.511Z</createdOn>
         </result>
        </results>
       </execution>
      </executions>
     </dataSource>
    </dataSources>
```

```
  </dataSchema>
</snapshotResource>
```

Data Schema Resource

Address: /responder/api/rest/dataschemas/{dataSchemaId}

Verbs: GET

Parameters:

Name	Type	Data Type	Description	Optional	Effect
dataSchemaId	path	int	Id of the Data Schema	true	If present, only the Data Schema with this Id is returned.

Data. See Data Schema XSD

Member	Data Type	Description	Scope
id	int	Id of this Data Schema Resource	GET
dataSchema.name	string	Name of this Data Schema	GET
dataSchema.field.name	string	Name of field	GET
dataSchema.field.dataType	string	Data type of field	GET
dataSchema.field.isArrray	boolean	Array or scalar?	GET
dataSchema.field.position	int	Query object to execute	GET
selfUri	uri	URI to this Data Schema Resource	GET
dataSourcesUri	uri	URI to child Data Sources	GET

Example:

Retrieve list of available Data Schemas

```
GET /responder/api/rest/dataschemas HTTP/1.1
Accept: application/vnd.encrypedquery.responder+xml; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encrypedquery.responder+xml; version=1

<dataSchemaResources
 xmlns="http://enquery.net/encryptedquery/dataschema"
 xmlns:res="http://enquery.net/encryptedquery/resource"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

 <dataSchemaResource>
  <res:id>1000</res:id>
  <res:selfUri>/responder/api/rest/dataschemas/1000</res:selfUri>
  <dataSchema>
   <name>name</name>
   <field>
    <name>name</name>
    <dataType>dataType</dataType>
    <position>0</position>
   </field>
  </dataSchema>

<dataSourcesUri>/responder/api/rest/dataschemas/1000/datasources</dataSo
urcesUri>
 </dataSchemaResource>

 <dataSchemaResource>
  <res:id>1010</res:id>
  <res:selfUri>/responder/api/rest/dataschemas/1010</res:selfUri>
  <dataSchema>
   <name>name</name>
   <field>
    <name>name</name>
    <dataType>dataType</dataType>
    <position>0</position>
   </field>
  </dataSchema>

<dataSourcesUri>/responder/api/rest/dataschemas/1010/datasources</dataSo
urcesUri>
 </dataSchemaResource>
</dataSchemaResources>
```

Data Source Resource

Address: /responder/api/rest/dataschemas/{dataSchemaId}/datasources/{dataSourceId}

Verbs: GET

Parameters:

Name	Type	Data Type	Description	Optional	Effect
dataSchemaId	path	int	Id of the parent Data Schema	true	If present, result set is restricted to this Data Schema. If omitted, the result set contains Data Sources from any Data Schema.
dataSourceId	path	int	Id of the Data Source to retrieve	true	If present, only the Data Source with this Id is returned.

Data. See Data Source XSD

Member	Data Type	Description	Scope
dataSourceResource.id	int	Id of this Data Source Resource	GET
dataSourceResource.dataSchemaId	int	Id of parent Data Schema	GET
dataSourceResource.datasource.name	string	Id of the parent Data Source	GET
dataSourceResource.datasource.description	string	Description of the Data Source	GET
dataSourceResource.datasource.type	string	batch or streaming	GET
dataSourceResource.selfUri	uri	URI to this Data Source Resource	GET
dataSourceResource.dataSchemaUri	uri	URI to parent Data Schema	GET
dataSourceResource.executionsUri	uri	URI to child executions	GET

Example:

Retrieve list of Data Sources available under Data Schema 1000

```
GET /responder/api/rest/dataschemas/1010/datasources HTTP/1.1
Accept: application/vnd.encrypedquery.responder+xml; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encrypedquery.responder+xml; version=1

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:dataSourceResources
 xmlns="http://enquery.net/encryptedquery/datasource"
 xmlns:ns2="http://enquery.net/encryptedquery/datasource/resource">

 <ns2:dataSourceResource>
  <id>2010</id>
  <ns2:dataSchemaId>1010</ns2:dataSchemaId>
     <dataSource>
   <name>Flink-Kafka-Phone-Record</name>
         <description>Flink Engine running on a Kafka Phone Record
Stream</description>
   <type>streaming</batch>
  </dataSource>

<ns2:selfUri>/responder/api/rest/dataschemas/1010/datasources/2010</ns2:
selfUri>
```

<ns2:dataSchemaUri>/responder/api/rest/dataschemas/1010</ns2:dataSchemaUri>

<ns2:executionsUri>/responder/api/rest/dataschemas/1010/datasources/2010/executions</ns2:executionsUri>
    </ns2:dataSourceResource>

 <ns2:dataSourceResource>
        <ns2:id>2011</ns2:id>
  <ns2:dataSchemaId>1010</ns2:dataSchemaId>
        <dataSource>
            <name>MapReduce-Json-TextFile-Phone-Record</name>
            <description>Phone Record Text Json File with Map Reduce</description>
   <type>batch</batch>
        </dataSource>

<ns2:selfUri>/responder/api/rest/dataschemas/1010/datasources/2011</ns2:selfUri>

<ns2:dataSchemaUri>/responder/api/rest/dataschemas/1010</ns2:dataSchemaUri>

<ns2:executionsUri>/responder/api/rest/dataschemas/1010/datasources/2011/executions</ns2:executionsUri>

```
    </ns2:dataSourceResource>

</ns2:dataSourceResources>
```

Execution Resource

Address: /responder/api/rest/dataschemas/{dataSchemaId}/datasources/{dataSourceId}/executions/{executionId}

Verbs: GET, POST

Parameters:

Name	Type	Data Type	Description	Optional	Effect
dataSchemaId	path	int	Id of the parent Data Schema	false	
dataSourceId	path	int	Id of the Data Source to retrieve	false	
executionId	path	int	Id of the Execution to retrieve	true	If present, only the Execution with this Id is returned, otherwise all executions restricted by *dataSourceId,* and *dataSchemaId* parameters (if any) are returned.
limit	query	int	Maximum number of items to be returned	true	If omitted the server default limit is used.
before	query	string	Cursor of the last item to return (exclusive)	true	If specified result set will include items up to (but not including) the element designated by this cursor.
after	query	string	Cursor of the first item to return (exclusive)	true	If specified result set will include items starting from (but not including) the element designated by this cursor.

Data. See Execution XSD.

Member	Data Type	Description	Scope
id	int	Id of the execution (system assigned)	GET
execution.submittedOn	timestamp	Time stamp when this execution was received by Responder	GET
execution.startedOn	timestamp	Time stamp when this execution started. Only if the Query was executed.	GET
execution.completedOn	timestamp	Time stamp when this execution was finished. Only if the Query was executed and completed.	GET
execution.scheduledFor	timestamp	Time stamp when this query is to be run.	GET, POST
execution.query	Query	Query to execute. See Query	POST
execution.configuration	key value map	Configuration for this execution	GET, POST
execution.errorMessage	string	Optional error message if execution failed.	GET
execution.cancelled	boolean	Optional flag indicating execution was cancelled.	GET
selfUri	uri	URI to this Execution resource.	GET
dataSourceUri	uri	URI to parent Data Source	GET
resultsUri	uri	URI to the results of this execution	GET
pagination.previousUri	uri	URI to the previous page of Execution resources	GET
pagination.nextUri	uri	URI to the next page of Execution resources	GET

Example:

Retrieve list of Executions

```
GET /responder/api/rest/dataschemas/1000/datasources/1000/executions
HTTP/1.1
Accept: application/vnd.encrypedquery.responder+xml; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encrypedquery.responder+xml; version=1

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:executionResources
     xmlns="http://enquery.net/encryptedquery/execution"
     xmlns:ns2="http://enquery.net/encryptedquery/execution/resource">

 <ns2:executionResource>
  <ns2:id>2010</ns2:id>
  <ns2:dataSourceId>1000</ns2:dataSourceId>
  <execution>
   <submittedOn>2018-04-23T18:25:43.511Z</submittedOn>
   <startedOn>2018-04-23T19:00:00.000Z</startedOn>
   <completedOn>2018-05-23T6:00:00.000Z</completedOn>
  </execution>

<ns2:selfUri>/responder/api/rest/dataschemas/1000/datasources/1000/execu
tions/2010</ns2:selfUri>

<ns2:dataSourceUri>/responder/api/rest/dataschemas/1000/datasources/1000
</ns2:dataSourceUri>

<ns2:resultsUri>/responder/api/rest/dataschemas/1000/datasources/1000/ex
ecutions/2010/results</ns2:resultsUri>
 </ns2:executionResource>

 <pagination>

<previousUri>/responder/api/rest/dataschemas/1000/datasources/1000/execu
tions?before=MTAxNTZ&amp;limit=100</previousUri>

<nextUri>/responder/api/rest/dataschemas/1000/datasources/1000/execution
s?after=NDMy&limit=100</nextUri>
 </pagination>
</ns2:executionResources>
```

The following interaction shows a POST to create a new Execution. Only elements *scheduledFor* and *query* are required in this case. The response includes a full Execution resource, with all its URIs and Id.

Submit a Query for execution

```
POST /responder/api/rest/dataschemas/1000/datasources/1000/executions
HTTP/1.1 Accept: application/vnd.encrypedquery.responder+xml; version=1

<execution xmlns:ns2="http://enquery.net/encryptedquery/execution" >
    <scheduledFor>2018-08-29T10:35:55.577-04:00</scheduledFor>
    <query>cXVlcnkgYnl0ZXM=</query>
</execution>

HTTP/1.1 201 Created
Content-Type: application/vnd.encrypedquery.responder+xml; version=1
Location: /querier/api/rest/dataschemas/123/queryschemas/9001

<ns2:executionResource
xmlns="http://enquery.net/encryptedquery/resource"
 xmlns:ns2="http://enquery.net/encryptedquery/execution" >
    <id>2147483602</id>

<selfUri>/responder/api/rest/dataschemas/2147483599/datasources/21474836
01/executions/2147483602</selfUri>
    <ns2:execution>

<ns2:scheduledFor>2018-08-29T10:35:55.577-04:00</ns2:scheduledFor>
        <ns2:submittedOn>2018-08-28T16:44:26.770-04:00</ns2:submittedOn>
    </ns2:execution>

<ns2:dataSourceUri>/responder/api/rest/dataschemas/2147483599/datasource
s/2147483601</ns2:dataSourceUri>

<ns2:resultsUri>/responder/api/rest/dataschemas/2147483599/datasources/2
147483601/executions/2147483602/results</ns2:resultsUri>
</ns2:executionResource>
```

Result Resource

Address: /responder/api/rest/dataschemas/{dataSchemaId}/datasources/{dataSourceId}/executions/{executionId}/results/{resultId}

Verbs: GET

Parameters:

Name	Type	Data Type	Description	Optional	Effect
dataSchemaId	path	int	Id of the parent Data Schema	false	
dataSourceId	path	int	Id of the parent Data Source	false	
executionId	path	int	Id of the parent Execution	false	
resultId	path	int	Id of the Result	true	If present, only the Result with this Id is returned.
limit	query	int	Maximum number of items to be returned	true	If omitted the server default limit is used.

before	query	string	Cursor of the last item to return (exclusive)	true	If specified result set will include items up to (but not including) the element designated by this cursor.
after	query	string	Cursor of the first item to return (exclusive)	true	If specified result set will include items starting from (but not including) the element designated by this cursor.

Data. See Result XSD

Member	Data Type	Description	Scope
id	int	Id of the result	GET
createdOn	timestamp	Time stamp when this result was created	GET
result.payload	blob	Encrypted Result payload	GET (*resultId* != null)
selfUri	uri	URI to this Result	GET
execution.selfUri	uri	URI to parent Execution	GET (*resultId* != null)
execution.id	int	Id of parent Execution	GET (*resultId* != null)
pagination.previousUri	uri	URI to the previous page of Result resources	GET
pagination.nextUri	uri	URI to the next page of Result resources	GET

Example:

Retrieve list of available Results

```
GET
/responder/api/rest/dataschemas/1000/datasources/1000/executions/2010/results?limit=10 HTTP/1.1
Accept: application/vnd.encrypedquery.responder+xml; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encrypedquery.responder+xml; version=1

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:resultResources xmlns="http://enquery.net/encryptedquery/resource"
xmlns:ns2="http://enquery.net/encryptedquery/result"
xmlns:ns3="http://enquery.net/encryptedquery/query"
xmlns:ns4="http://enquery.net/encryptedquery/queryschema"
xmlns:ns5="http://enquery.net/encryptedquery/dataschema"
xmlns:ns6="http://enquery.net/encryptedquery/response"
xmlns:ns7="http://enquery.net/encryptedquery/pagination"
xmlns:ns8="http://enquery.net/encryptedquery/execution"
xmlns:ns9="http://enquery.net/encryptedquery/datasource">
    <ns2:resultResource>
        <id>3010</id>

<selfUri>/responder/api/rest/dataschemas/1000/datasources/1000/executions/2010/results/3010</selfUri>
        <ns2:createdOn>2018-09-20T09:35:03.614-04:00</ns2:createdOn>
        <ns2:execution>
            <id>2010</id>

<selfUri>/responder/api/rest/dataschemas/1000/datasources/1000/executions/2010</selfUri>
        </ns2:execution>
    </ns2:resultResource>
 <ns2:resultResource>
  <id>3011</id>

<selfUri>/responder/api/rest/dataschemas/1000/datasources/1000/executions/2010/results/3011</selfUri>
  <ns2:createdOn>2018-09-20T10:35:03.614-04:00</ns2:createdOn>
  <ns2:execution>
   <id>2010</id>

<selfUri>/responder/api/rest/dataschemas/1000/datasources/1000/executions/2010</selfUri>
  </ns2:execution>
 </ns2:resultResource>
</ns2:resultResources>
```

Retrieve a specific Result, including its payload

```
GET
/responder/api/rest/dataschemas/1000/datasources/1000/executions/2010/re
sults/3010 HTTP/1.1
Accept: application/vnd.encrypedquery.responder+xml; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encrypedquery.responder+xml; version=1

<ns2:resultResource xmlns="http://enquery.net/encryptedquery/resource"
xmlns:ns2="http://enquery.net/encryptedquery/result"
xmlns:ns3="http://enquery.net/encryptedquery/query"
xmlns:ns4="http://enquery.net/encryptedquery/queryschema"
xmlns:ns5="http://enquery.net/encryptedquery/dataschema"
xmlns:ns6="http://enquery.net/encryptedquery/response"
xmlns:ns7="http://enquery.net/encryptedquery/execution"
xmlns:ns8="http://enquery.net/encryptedquery/datasource">
    <id>2147483603</id>

<selfUri>/responder/api/rest/dataschemas/1000/datasources/1000/execution
s/2010/results/3010</selfUri>
    <ns2:createdOn>2018-09-20T09:35:03.614-04:00</ns2:createdOn>
    <ns2:payload>
        <ns6:queryInfo>

<ns3:identifier>d77bc121-638c-4bba-b527-469aa4a75f2c</ns3:identifier>
            <ns3:numSelectors>1</ns3:numSelectors>
            <ns3:queryType>Books</ns3:queryType>
            <ns3:hashBitSize>9</ns3:hashBitSize>
            <ns3:hashKey>f4162eadb4bd97c8a302</ns3:hashKey>
            <ns3:numBitsPerDataElement>0</ns3:numBitsPerDataElement>
            <ns3:dataPartitionBitSize>8</ns3:dataPartitionBitSize>

<ns3:numPartitionsPerDataElement>0</ns3:numPartitionsPerDataElement>
            <ns3:useExpLookupTable>false</ns3:useExpLookupTable>
            <ns3:useHDFSLookupTable>false</ns3:useHDFSLookupTable>
            <ns3:embedSelector>true</ns3:embedSelector>
            <ns4:querySchema>
                <ns4:name>Books</ns4:name>
                <ns4:selectorField>title</ns4:selectorField>
                <ns4:field>
                    <ns4:name>price</ns4:name>
                    <ns4:lengthType>fixed</ns4:lengthType>
                    <ns4:size>0</ns4:size>
                    <ns4:maxArrayElements>1</ns4:maxArrayElements>
                </ns4:field>
                <ns5:dataSchema>
                    <ns5:name>Books</ns5:name>
```

```
                    <ns5:field>
                        <ns5:name>id</ns5:name>
                        <ns5:dataType>int</ns5:dataType>
                        <ns5:isArray>false</ns5:isArray>
                        <ns5:position>0</ns5:position>
                    </ns5:field>
                    <ns5:field>
                        <ns5:name>title</ns5:name>
                        <ns5:dataType>string</ns5:dataType>
                        <ns5:isArray>false</ns5:isArray>
                        <ns5:position>1</ns5:position>
                    </ns5:field>
                    <ns5:field>
                        <ns5:name>author</ns5:name>
                        <ns5:dataType>string</ns5:dataType>
                        <ns5:isArray>false</ns5:isArray>
                        <ns5:position>2</ns5:position>
                    </ns5:field>
                    <ns5:field>
                        <ns5:name>price</ns5:name>
                        <ns5:dataType>double</ns5:dataType>
                        <ns5:isArray>false</ns5:isArray>
                        <ns5:position>3</ns5:position>
                    </ns5:field>
                    <ns5:field>
                        <ns5:name>qty</ns5:name>
                        <ns5:dataType>int</ns5:dataType>
                        <ns5:isArray>false</ns5:isArray>
                        <ns5:position>4</ns5:position>
                    </ns5:field>
                </ns5:dataSchema>
            </ns4:querySchema>
        </ns6:queryInfo>
    </ns2:payload>
    <ns2:execution>
        <id>2010</id>

<selfUri>/responder/api/rest/dataschemas/1000/datasources/1000/execution
s/2010</selfUri>
```

```
    </ns2:execution>
</ns2:resultResource>
```

XSD Schemas

Query XSD

```
<xs:schema targetNamespace="http://enquery.net/encryptedquery/query"
 xmlns="http://enquery.net/encryptedquery/query"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:qschema="http://enquery.net/encryptedquery/queryschema"
 elementFormDefault="qualified"
 version="2.0">

 <xs:import schemaLocation="query-schema.xsd"
namespace="http://enquery.net/encryptedquery/queryschema" />

 <xs:element name="query" type="Query"/>

 <xs:complexType name="QueryInfo">
  <xs:sequence>
   <xs:element name="queryId" type="xs:string" />
   <xs:element name="queryName" type="xs:string" minOccurs="0" />
   <xs:element name="cryptoSchemeId" type="xs:string"/>
   <xs:element name="publicKey" type="xs:base64Binary" />
   <xs:element name="numSelectors" type="xs:int" minOccurs="0" />
   <xs:element name="hashBitSize" type="xs:int" />
   <xs:element name="hashKey" type="xs:string" minOccurs="0" />
   <xs:element name="numBitsPerDataElement" type="xs:int" minOccurs="0"
/>
   <xs:element name="dataChunkSize" type="xs:int" />
   <xs:element name="numPartitionsPerDataElement" type="xs:int"
minOccurs="0" />
   <xs:element name="embedSelector" type="xs:boolean" />
   <xs:element ref="qschema:querySchema" />
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="Query">
  <xs:sequence>
   <xs:element name="queryInfo" type="QueryInfo" />
   <xs:element name="queryElements">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="entry" minOccurs="0" maxOccurs="unbounded">
       <xs:complexType>
```

```
        <xs:attribute name="key" type="xs:int" use="required"/>
        <xs:attribute name="value" type="xs:base64Binary"
use="required"/>
       </xs:complexType>
      </xs:element>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
```

```
  <xs:attribute name="schemaVersion" type="xs:decimal" use="required"/>
 </xs:complexType>
</xs:schema>
```

Data Source XSD

```
<xs:schema
 targetNamespace="http://enquery.net/encryptedquery/datasource"
 xmlns="http://enquery.net/encryptedquery/datasource"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified">

 <xs:element name="dataSource" type="dataSource">
 </xs:element>

 <xs:complexType name="dataSource">
  <xs:sequence>
   <xs:element name="name" type="xs:string">
    <xs:annotation>
     <xs:documentation>
      The name of the data source.
      The name omits
      leading and trailing whitespace, and is case sensitive.
     </xs:documentation>
    </xs:annotation>
   </xs:element>

   <xs:element name="description" type="xs:string">
    <xs:annotation>
     <xs:documentation>
      Description of this data Source.
     </xs:documentation>
    </xs:annotation>
   </xs:element>

   <xs:element name="dataSchemaName" type="xs:string">
    <xs:annotation>
     <xs:documentation>
      The data schema name that is used by this data source.
     </xs:documentation>
    </xs:annotation>
   </xs:element>

   <xs:element name="type">
    <xs:annotation>
     <xs:documentation>
      The type of this data source.
```

```
    </xs:documentation>
   </xs:annotation>
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:enumeration value="Batch" />
     <xs:enumeration value="Streaming" />
    </xs:restriction>
   </xs:simpleType>
  </xs:element>
```

```
  </xs:sequence>
 </xs:complexType>
</xs:schema>
```

Data Schema XSD

```
<xs:schema
 targetNamespace="http://enquery.net/encryptedquery/dataschema"
 xmlns="http://enquery.net/encryptedquery/dataschema"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified">

 <xs:element name="dataSchema" type="dataSchema">
  <xs:unique name="uniqueFieldNames">
              <xs:selector xpath="field"/>
              <xs:field xpath="name"/>
        </xs:unique>
        <xs:unique name="uniqueFieldPositions">
   <xs:selector xpath="field" />
   <xs:field xpath="position" />
  </xs:unique>
 </xs:element>

 <xs:complexType name="dataSchema">
  <xs:sequence>
   <xs:element name="name" type="xs:string">
    <xs:annotation>
     <xs:documentation>The name of the data
      schema. The name omits
      leading and trailing whitespace, and is case
      sensitive.
     </xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="field" minOccurs="1" maxOccurs="unbounded">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="name" type="xs:string">
       <xs:annotation>
        <xs:documentation>The name of the data
         element. The name omits
         leading and trailing whitespace, and is case
         sensitive.
        </xs:documentation>
       </xs:annotation>
      </xs:element>
      <xs:element name="dataType" type="xs:string">
```

```
     <xs:annotation>
      <xs:documentation>The type of the data
       element. The type name is
       the fully qualified class name, or the
       primitive
       Java type of the
       element.
      </xs:documentation>
     </xs:annotation>
    </xs:element>
    <xs:element name="isArray" type="xs:boolean"
     default="false" minOccurs="0">
     <xs:annotation>
      <xs:documentation>Whether or not the schema element is an
       array
       within the data. False if unspecified.
      </xs:documentation>
     </xs:annotation>
    </xs:element>
    <xs:element name="position" type="xs:int" default="0">
     <xs:annotation>
      <xs:documentation> Index of this field in the data schema. This
       is used to
       be able to lookup fields by position instead of by
       name. In some
       data sources,
       like JDBC, or flat files (TBS, CSV,
       etc.) looking fields by name is
       not possible.
      </xs:documentation>
     </xs:annotation>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
```

```
  </xs:sequence>
 </xs:complexType>
</xs:schema>
```

Execution XSD

```
<xs:schema
 targetNamespace="http://enquery.net/encryptedquery/execution"
 xmlns="http://enquery.net/encryptedquery/execution"
 xmlns:ex="http://enquery.net/encryptedquery/execution"
 xmlns:pag="http://enquery.net/encryptedquery/pagination"
 xmlns:q="http://enquery.net/encryptedquery/query"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified">

 <xs:import schemaLocation="query.xsd"
namespace="http://enquery.net/encryptedquery/query"/>

 <xs:element name="execution" type="execution" />


 <xs:complexType name="Configuration">
  <xs:sequence>
   <xs:element name="entry" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
     <xs:attribute name="key" type="xs:string" use="required" />
     <xs:attribute name="value" type="xs:string" use="required" />
    </xs:complexType>
   </xs:element>
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="execution">
  <xs:sequence>
   <xs:element name="scheduledFor" type="xs:dateTime">
    <xs:annotation>
     <xs:documentation>Time stamp when this Execution is expected to be
run.  This time stamp is rounded to the next minute.</xs:documentation>
    </xs:annotation>
   </xs:element>

   <xs:element name="configuration" type="Configuration" minOccurs="0">
    <xs:annotation>
     <xs:documentation>Additional configuration for this
execution.</xs:documentation>
```

```
    </xs:annotation>
   </xs:element>

   <xs:element name="submittedOn" type="xs:dateTime" minOccurs="0">
    <xs:annotation>
     <xs:documentation>Time stamp when this Execution was originally
received. System assigned, not present for
submissions.</xs:documentation>
    </xs:annotation>
   </xs:element>

   <xs:element name="startedOn" type="xs:dateTime" minOccurs="0">
    <xs:annotation>
     <xs:documentation>Time stamp when this Execution was started.
System assigned, not present for submissions.</xs:documentation>
    </xs:annotation>
   </xs:element>

   <xs:element name="completedOn" type="xs:dateTime" minOccurs="0">
    <xs:annotation>
     <xs:documentation>Time stamp when this Execution finished running.
System assigned, not present for submissions.</xs:documentation>
    </xs:annotation>
   </xs:element>

   <xs:element name="errorMessage" minOccurs="0" type="xs:string">
    <xs:annotation>
     <xs:documentation>Error message of this execution if failed.
Optional. Not present for submissions.</xs:documentation>
    </xs:annotation>
   </xs:element>

   <xs:element name="cancelled" minOccurs="0" type="xs:boolean">
    <xs:annotation>
     <xs:documentation>True if job was canceled. Optional. Not present
for submissions.</xs:documentation>
    </xs:annotation>
   </xs:element>

   <!-- Keep query at the end, since it can be beig, to allow parsing
the small
    data elements without having to load the query -->
   <xs:element name="query" type="q:Query" minOccurs="0">
    <xs:annotation>
     <xs:documentation>Query to be executed execute. Only present for
submissions, since this can be very large.</xs:documentation>
    </xs:annotation>
   </xs:element>
```

```
  </xs:sequence>
 </xs:complexType>
</xs:schema>
```

Result XSD

```
<xs:schema targetNamespace="http://enquery.net/encryptedquery/result"
xmlns="http://enquery.net/encryptedquery/result"
 xmlns:res="http://enquery.net/encryptedquery/resource"
 xmlns:resp="http://enquery.net/encryptedquery/response"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified">

 <xs:import schemaLocation="response.xsd"
namespace="http://enquery.net/encryptedquery/response" />
 <xs:import schemaLocation="resource.xsd"
namespace="http://enquery.net/encryptedquery/resource" />

 <xs:complexType name="ResultResource">
  <xs:complexContent>
   <xs:extension base="res:resource">
     <xs:sequence>
      <xs:element name="createdOn" type="xs:dateTime">
      <xs:annotation>
       <xs:documentation>Time stamp when this result was
created.</xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element name="windowStart" type="xs:dateTime">
      <xs:annotation>
       <xs:documentation>Start time stamp of this result time
window.</xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element name="windowEnd" type="xs:dateTime">
      <xs:annotation>
       <xs:documentation>End time stamp of this result time
window.</xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element name="execution" type="res:resource">
      <xs:annotation>
       <xs:documentation>Reference to the parent
Execution.</xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element name="payload" type="resp:Response" minOccurs="0">
```

```
      <xs:annotation>
       <xs:documentation>The response data payload of this
result.</xs:documentation>
      </xs:annotation>
     </xs:element>
    </xs:sequence>
   </xs:extension>
```

```
  </xs:complexContent>
 </xs:complexType>
</xs:schema>
```

Response XSD

```
<xs:schema targetNamespace="http://enquery.net/encryptedquery/response"
 xmlns="http://enquery.net/encryptedquery/response"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:q="http://enquery.net/encryptedquery/query"
 elementFormDefault="qualified"
 version="2.0">

 <xs:import schemaLocation="query.xsd"
namespace="http://enquery.net/encryptedquery/query" />

 <xs:element name="response" type="Response"/>

 <xs:complexType name="Response">
  <xs:sequence>
   <xs:element name="queryInfo" type="q:QueryInfo" />
   <xs:element name="resultSet" minOccurs="0" maxOccurs="unbounded"
type="ResultSet"/>
  </xs:sequence>
  <xs:attribute name="schemaVersion" type="xs:decimal" use="required"/>
 </xs:complexType>

 <xs:complexType name="ResultSet">
  <xs:sequence>
   <xs:element name="result" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
     <xs:attribute name="column" type="xs:int" use="required"/>
     <xs:attribute name="value" type="xs:base64Binary" use="required"/>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:schema>
```