

Enclave Orchestration Server - Rest Services

- [Inventory](#)
- [Resource Hierarchy](#)
- [Resource Data Model](#)
- [Versioning](#)
- [Common attributes present in all JSON Response Objects](#)
- [Detailed Description of the Resources](#)
 - [Root](#)
 - [DataSchema](#)
 - [DataSource](#)
 - [QuerySchema](#)
 - [Query](#)
 - [Schedule](#)
 - [Result](#)
 - [Retrieval](#)
 - [Decryption](#)

Inventory

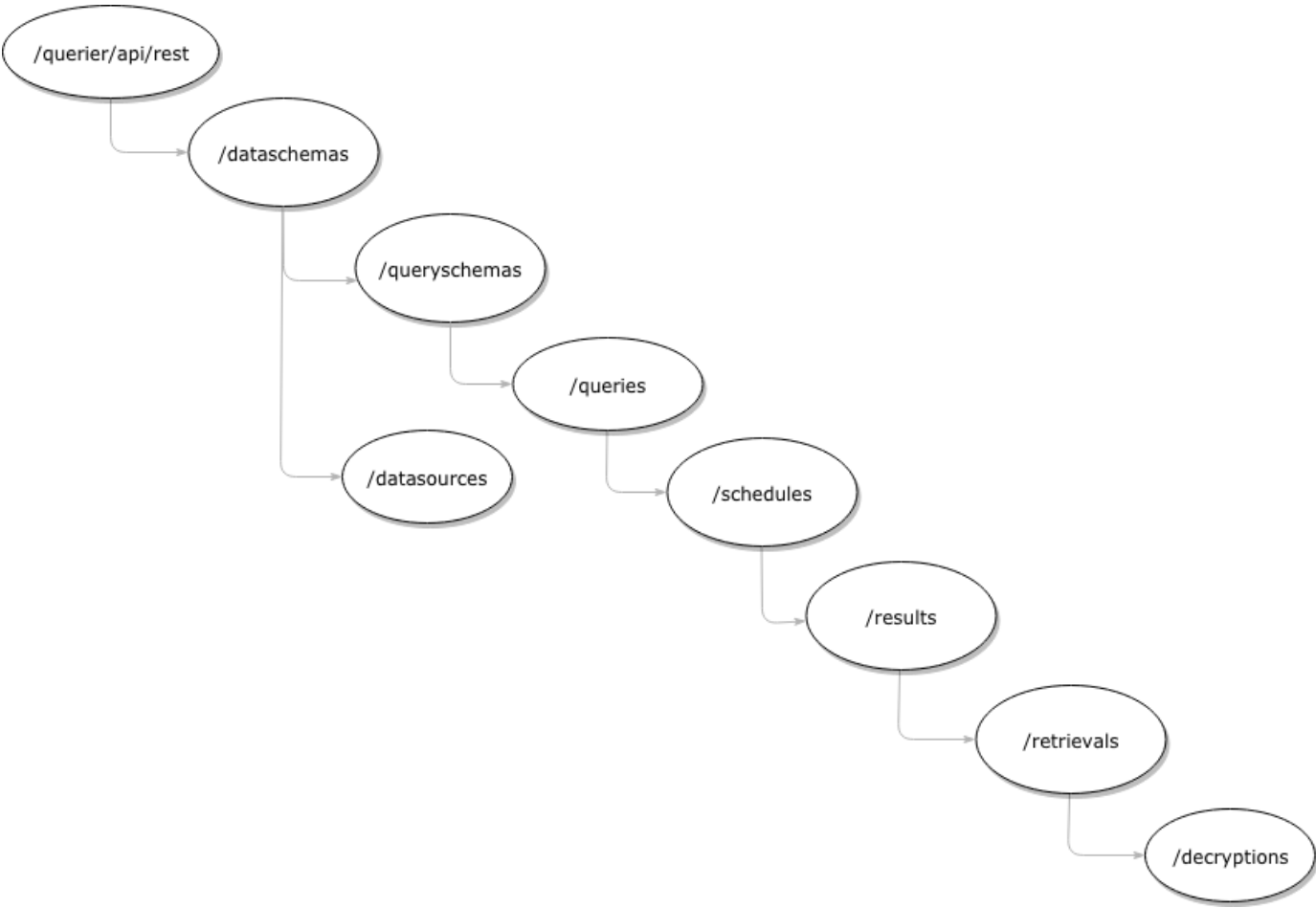
The following table lists all resources provided by the Orchestration Server to be used for the Secure Enclave User Interface.

All Services use JSON format with vendor specific Media Type: *application/vnd.encryptedquery.enclave+json*

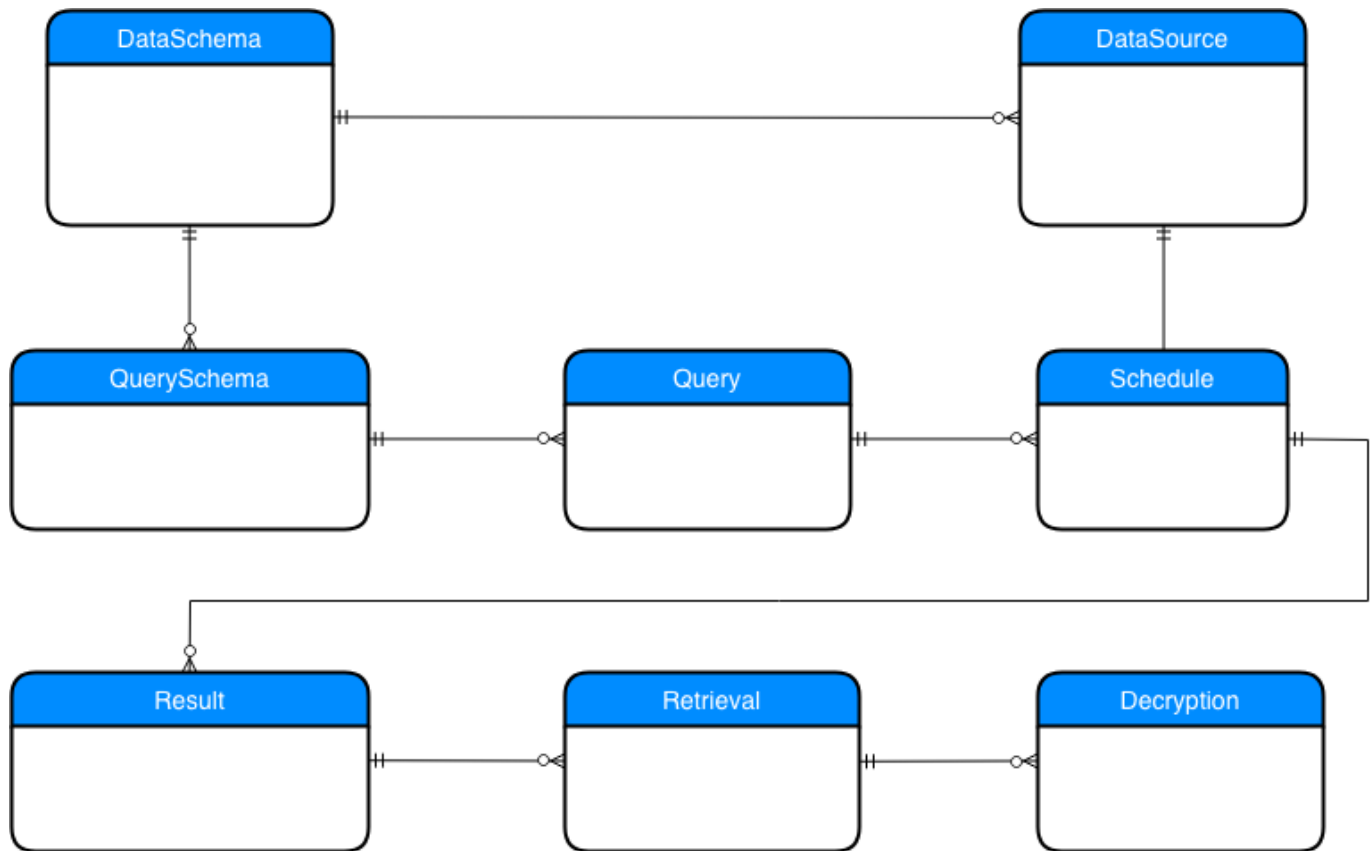
Resource	Description	URI
Root	List of root resources and their URIs. This can be used by the UI to avoid hard coding the various service URIs. For now, there is only one root resource: / <i>dataschemas</i>	/querier/api/rest/
DataSchema	Data Schemas.	/querier/api/rest/dataschemas
DataSource	Data Sources for a given Data Schema.	/querier/api/rest/dataschemas/{ <i>dataSchemaId</i> }/datasources
QuerySchema	Query Schemas for a given Data Schema.	/querier/api/rest/dataschemas/{ <i>dataSchemaId</i> }/queryschemas
Query	Queries for a given Query Schema.	/querier/api/rest/dataschemas/{ <i>dataSchemaId</i> }/queryschemas/{ <i>querySchemaId</i> }/queries
Schedule	Execution schedules for a given Query. A Query can be executed multiple times. Each Query execution is the result of a fulfilled Schedule.	/querier/api/rest/dataschemas/{ <i>dataSchemaId</i> }/queryschemas/{ <i>querySchemaId</i> }/queries/{ <i>queryId</i> }/schedules
Result	Results are produced by the fulfillment of a Schedule, i.e., they are the result of running a Query. Depending on the type of Data Source (Batch or Streaming), there may be more than one result for a given Schedule.	/querier/api/rest/dataschemas/{ <i>dataSchemaId</i> }/queryschemas/{ <i>querySchemaId</i> }/queries/{ <i>queryId</i> }/schedules/{ <i>scheduleId</i> }/results
Retrieval	Retrievals of a query result. Once a Result is available, the actual payload of the result can be fetched from the remote Cloud system. This is a potentially long running operation initiated by the actor. Normally, there will be zero or one Retrieval for each Result, but there may be more in case of system/transmission errors.	/querier/api/rest/dataschemas/{ <i>dataSchemaId</i> }/queryschemas/{ <i>querySchemaId</i> }/queries/{ <i>queryId</i> }/schedules/{ <i>scheduleId</i> }/results/{ <i>resultId</i> }/retrievals

Decryption	Once a Result is retrieved, it can be decrypted. This is a potentially long running operation initiated by the actor. Normally, there will be zero or one Decryptions for each Retrieval, but there may be more in case of system errors. A fulfilled Decryption contains the plain text which is the result of the Query Execution.	<i>/querier/api/rest/dataschemas/{dataSchemaId}/queryschemas/{querySchemaId}/queries/{queryId}/schedules/{scheduleId}/results/{resultId}/retrievals/{retrievalId}/decryptions</i>
------------	--	---

Resource Hierarchy



Resource Data Model



Versioning

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

Media Type: *application/vnd.encryptedquery.enclave+json*

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 /querier/api/rest/dataschemas/42 HTTP/1.1
Accept: application/vnd.encryptedquery.enclave+json; version=1
```

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

```
HTTP/1.1 200 OK
Content-Type: application/vnd.encryptedquery.enclave+json; version=1
```

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

Common attributes present in all JSON Response Objects

The following attributes are common to all the JSON responses.

Field	Type	Description
-------	------	-------------

data.id	string	The unique permanent identifier of the object. In combination with type, uniquely identifies an object,
data.type	string	The type of the object.
data.selfUri	uri	The URI to the object.
included	array of objects	Optional array of referenced (related) objects. The main data object of the response may reference other objects, in these cases, an "included" attribute is added to the response including these referenced objects for the convenience of the client and to avoid additional round trips.

Detailed Description of the Resources

Root

Address: /querier/api/rest/

Verbs: GET

```
GET /querier/api/rest/ HTTP/1.1

{ "data" : [ {
  "id" : "dataschema",
  "selfUri" : "/querier/api/rest/dataschemas",
  "type" : "Resource"
} ] }
```

DataSchema

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

Verbs: GET

Parameters:

Name	Type	Data Type	Description	Optional	Valid with
dataSchemaId	path	string	Id of the Data Schema to retrieve, if not specified, a list of all Data Schemas is retrieved.	true	GET

Data:

Member	Data type	Description	Omitted During
id	string	Unique id	
selfUri	uri	Uri to this data schema	
type	string	Type of entity: hard coded to "DataSchema"	
name	string	Name of this Data Schema	
fields	<i>DataSchemaField</i> []	Array of fields of this Data Schema	GET('dataSchemaId' = null)
selfUri	uri	URI to self	
dataSourcesUri	uri	URI to child Data Sources	GET('dataSchemaId' = null)
querySchemasUri	uri	URI to child Query Schemas	GET('dataSchemaId' = null)

DataSchemaField:

Member	Data type	Description
name	string	Field name.
dataType	string	Data type of this field

isArray	boolean	True if array, false if not.
position	int	The zero-based index of this field in the source data set. Some Data Sources don't support access fields by name, like CSV, TSV, etc.

Examples:

Retrieve list of DataSchemas

```
GET /querier/api/rest/dataschemas HTTP/1.1
Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encryptedquery.enclave+json; version=1

{"data": [
  {
    "id": "123",
    "selfUri": "/querier/api/rest/dataschemas/123",
    "type" : "DataSchema",
    "name": "Phone Records"
  }, {
    "id": "124",
    "selfUri": "/querier/api/rest/dataschemas/124",
    "type" : "DataSchema",
    "name": "Books"
  }
]}
```

Retrieve DataSchema 123

```
GET /querier/api/rest/dataschemas/123 HTTP/1.1
Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encryptedquery.enclave+json; version=1

{"data": {
  "id": "123",
  "selfUri": "/querier/api/rest/dataschemas/123",
  "type" : "DataSchema",
  "name": "Phone Records",
  "datasourcesUri": "/querier/api/rest/dataschemas/123/datasources",
  "queryschemasUri": "/querier/api/rest/dataschemas/123/queryschemas",
  "fields" : [
    { "name": "action", "dataType": "string", "isArray": "false",
"position": 0 },
    { "name": "duration", "dataType": "double", "isArray": "false",
"position": 1 }
  ]
}}
```

DataSource

Address: /querier/api/rest/dataschemas/{*dataSchemaId*}/datasources/{*dataSourceId*}

Verbs: GET

Parameters:

Name	Type	Data Type	Description	Optional
<i>dataSchemaId</i>	path	string	Id of parent Data Schema.	
<i>dataSourceId</i>	path	string	Id of the Data Source to Retrieve	GET

Data:

Member	Data type	Description	Omitted if
id	string	Id of this Data Source	
selfUri	uri	URI to this Data Source	
type	string	Type of entity: hard coded to "DataSource"	
name	string	Name of this Data Source	
description	string	Description of this Data Source	
processingMode	string	Processing mode: Streaming/Batch	GET('dataSourceId' = null)
dataSchema.id	string	Id of this Data Source	GET('dataSourceId' = null)
dataSchema.selfUri	uri	URI to parent Data Schema	GET('dataSourceId' = null)
dataSchema.type	string	Type of entity: hard coded to "DataSchema"	GET('dataSourceId' = null)

Examples:

Retrieve list of Data Sources

```
GET /querier/api/rest/dataschemas/123/datasources HTTP/1.1
Accept: application/vnd.encryptedquery.enclave+json; version=1
```

```
HTTP/1.1 200 OK
Content-Type: application/vnd.encryptedquery.enclave+json; version=1
```

```
{
  "data": [
    {
      "id": "875",
      "selfUri" : "/querier/api/rest/dataschemas/123/datasources/875",
      "type" : "DataSource",
      "name": "Phones-MR"
    },
    {
      "id": "876",
      "selfUri" : "/querier/api/rest/dataschemas/123/datasources/876",
      "type" : "DataSource",
      "name": "Phones-JDBC"
    }
  ]
}
```

Retreive DataSource 875

GET /querier/api/rest/dataschemas/123/datasources/875 HTTP/1.1

Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK

Content-Type: application/vnd.encryptedquery.enclave+json; version=1

```
{
  "data": {
    "id": "875",
    "selfUri" : "/querier/api/rest/dataschemas/123/datasources/875",
    "type" : "DataSource",
    "name": "Phones-MR",
    "description": "Phones, MapReduce Hadoop",
    "dataSchema" : {
      "id" : "123",
      "selfUri" : "/querier/api/rest/dataschemas/123",
      "type" : "DataSchema"
    },
    "processingMode" : "Batch"
  },
  "included" : [ {
    "id" : "123",
    "selfUri" : "/querier/api/rest/dataschemas/123",
    "type" : "DataSchema",
    "name" : "Books",
    "datasourcesUri" : "/querier/api/rest/dataschemas/7/datasources",
    "querySchemasUri" : "/querier/api/rest/dataschemas/7/querieschemas",
    "fields" : [ {
      "name" : "caller",
      "dataType" : "string",
      "isArray" : false,
      "position" : 0
    }, {
      "name" : "callee",
      "dataType" : "string",
      "isArray" : false,
      "position" : 1
    } ]
  } ]
}
```

QuerySchema

Address: /querier/api/rest/dataschemas/{dataSchemaId}/querieschemas/{querySchemaId}

Verbs: GET, POST

Parameters:

Name	Type	Data Type	Description	Optional
<i>dataSchemaId</i>	path	string	Id of parent Data Schema.	
<i>querySchemaId</i>	path	string	Id of the Query Schema	GET

Data:

Member	Data type	Description	Omitted if
id	string	Unique id	POST
selfUri	uri	URI to self	POST
type	string	Type of entity: hard coded to "QuerySchema"	
name	string	Name of this Querychema	
selectorField	string	Name of field to be used as selector	GET('dataSourceId' = null)
fields	<i>QuerySchemaField</i> []	Array of fields of this Query Schema	GET('dataSourceId' = null)
dataSchema.id	string	Id of the parent Data Schema	POST GET('dataSourceId' = null)
dataSchema.selfUri	uri	URI to parent Data Schema	POST GET('dataSourceId' = null)
dataSchema.type	string	Entity type: hard coded to "DataSchema"	POST GET('dataSourceId' = null)
queriesUri	uri	URI to child Queries	POST GET('dataSourceId' = null)

QuerySchemaField:

Member	Data type	Description
name	string	Field name. Must be one of the fields in the associated DataSchema.
lengthType	string	Type of length: fixed/variable
size	int	Size of bytes to return
maxArrayElements	int	If array, how many elements to return.

Examples:

Retrieve Query Schemas for DataSchema 123

```
GET /querier/api/rest/dataschemas/123/querieschemas HTTP/1.1
Accept: application/vnd.encryptedquery.enclave+json; version=1
```

```
HTTP/1.1 200 OK
Content-Type: application/vnd.encryptedquery.enclave+json; version=1
```

```
{ "data": [
  {
    "id": "456",
    "selfUri": "/querier/api/rest/dataschemas/123/querieschemas/456",
    "type" : "QuerySchema",
    "name": "My First Query Schema",
  },
  {
    "id": "789",
    "selfUri": "/querier/api/rest/dataschemas/123/querieschemas/789",
    "type" : "QuerySchema",
    "name": "Another Query Schema"
  }
]}
```

Retrieve Query Schema 456 of Data Schema 123

GET /querier/api/rest/dataschemas/123/querieschemas/456 HTTP/1.1

Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK

Content-Type: application/vnd.encryptedquery.enclave+json; version=1

```
{ "data": {
  "id": "456",
  "selfUri": "/querier/api/rest/dataschemas/123/querieschemas/456",
  "type" : "QuerySchema",
  "name": "My First Query Schema",
  "selectorField" : "caller",
  "fields": [
    { "name": "caller", "lengthType": "fixed", "size": 12,
      "maxArrayElements": 5 },
    { "name": "callee", "lengthType": "fixed", "size": 12,
      "maxArrayElements": 5 },
    { "name": "time_stamp", "lengthType": "fixed", "size": 20,
      "maxArrayElements": 0 }
  ],
  "dataSchema" : {
    "id" : "123",
    "selfUri" : "/querier/api/rest/dataschemas/123",
    "type" : "DataSchema"
  },
  "queriesUri":
"/querier/api/rest/dataschemas/123/querieschemas/456/queries"
},
"included" : [ {
  "id" : "123",
  "selfUri" : "/querier/api/rest/dataschemas/123",
  "type" : "DataSchema",
  "name" : "Test Data Schema 123",
  "dataSourcesUri" : "/querier/api/rest/dataschemas/123/datasources",
  "querySchemasUri" :
"/querier/api/rest/dataschemas/123/querieschemas",
  "fields" : [ {
    "name" : "caller",
    "dataType" : "string",
    "isArray" : false,
    "position" : 0
  } ]
} ]
}
```

Create a new Query Schema

POST /querier/api/rest/dataschemas/123/queryschemas HTTP/1.1

```
{
  "type": "QuerySchema",
  "name": "Query Schema 1",
  "selectorField": "field1",
  "fields": [{
    "name": "field1",
    "lengthType": "fixed",
    "size": 10,
    "maxArrayElements": 1
  }]
}
```

HTTP/1.1 201 Created

Content-Type: application/vnd.encryptedquery.enclave+json; version=1

Location: /querier/api/rest/dataschemas/123/queryschemas/9001

```
{ "data": {
  "id" : "9001",
  "selfUri": "/querier/api/rest/dataschemas/123/queryschemas/9001",
  "type": "QuerySchema",
  "name": "Query Schema 1",
  "selectorField": "field1",
  "fields": [{
    "name": "field1",
    "lengthType": "fixed",
    "size": 10,
    "maxArrayElements": 1
  }],
  "dataSchema" : {
    "id" : "123",
    "selfUri" : "/querier/api/rest/dataschemas/123",
    "type" : "DataSchema"
  },
  "queriesUri":
"/querier/api/rest/dataschemas/123/queryschemas/9001/queries"
},
  "included" : [ {
    "id" : "123",
    "selfUri" : "/querier/api/rest/dataschemas/123",
    "type" : "DataSchema",
    "name" : "Books",
    "dataSourcesUri" : "/querier/api/rest/dataschemas/123/datasources",
    "querySchemasUri" :
"/querier/api/rest/dataschemas/123/queryschemas",
    "fields" : [ {
      "name" : "field1",
```

```
"dataType" : "string",  
"isArray" : false,  
"position" : 0
```

```
} ]  
}  
}
```

Query

Address: /querier/api/rest/dataschemas/{*dataSchemaId*}/queryschemas/{*querySchemaId*}/queries/{*queryId*}

Verbs: GET, POST

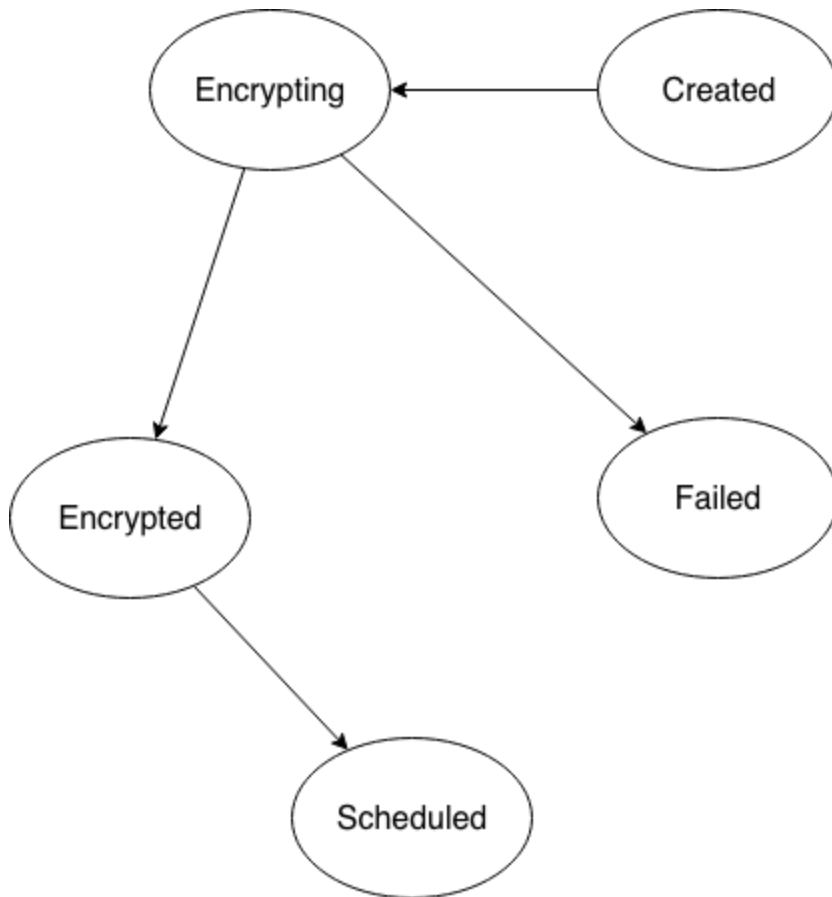
Parameters:

Name	Type	Data Type	Description	Optional
<i>dataSchemaId</i>	path	string	Id of parent Data Schema.	
<i>querySchemaId</i>	path	string	Id of parent Query Schema	
<i>queryId</i>	path	string	Id of the Query to retrieve	GET

Data:

Member	Data type	Description	Omitted if
id	string	Unique id of this Query	POST
selfUri	uri	URI to this Query	POST
type	string	Type of entity: hard coded to "Query"	
name	string	Name of this Query	
status	string	Status of this query: Created, Encrypting, Encrypted, Sscheduled, Failed	POST
dataPartitionBitSize	int		GET('queryId' = null)
hashBitSize	int		GET('queryId' = null)
bitSet	int		GET('queryId' = null)
paillierBitSize	int		GET('queryId' = null)
certainty	int		GET('queryId' = null)
embedSelector	boolean		GET('queryId' = null)
querySchema.id	string	Id of the parent QuerySchema	POST, GET('queryId' = null)
querySchema.selfUri	uri	URI to parent Query Schema	POST, GET('queryId' = null)
querySchema.type	string	Type of entity: hard coded to "QuerySchema"	POST, GET('queryId' = null)
schedulesUri	uri	URI to child Schedules	POST, GET('queryId' = null)

Query Status Transitions



Examples:

List of Queries

```
GET /querier/api/rest/dataschemas/123/queryschemas/456/queries HTTP/1.1
Accept: application/vnd.encryptedquery.enclave+json; version=1
```

```
HTTP/1.1 200 OK
Content-Type: application/vnd.encryptedquery.enclave+json; version=1
```

```
{
  "data" : [ {
    "id" : "199",
    "selfUri" :
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/199",
    "type" : "Query",
    "name" : "Query 2",
    "status" : "Created"
  } ]
}
```

Retrieve Query 974 of QuerySchema 456

```
GET /querier/api/rest/dataschemas/123/queryschemas/456/queries/974
HTTP/1.1
Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encryptedquery.enclave+json; version=1

{
  "data": {
    "id": "974",
    "selfUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974",
    "type" : "Query",
    "name": "Sample Query",
    "status": "Encrypted",
    "querySchema" : {
      "id" : "456",
      "selfUri" : "/querier/api/rest/dataschemas/123/queryschemas/456",
      "type" : "QuerySchema"
    },
    "schedulesUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedule
s",
    "parameters": {
      "hashBitSize" : "8",
      "dataChunkSize" : "1"
    },
    "selectorValues": [
      "410-203-3243",
      "675-755-8753",
      "768-334-1234"
    ],
    "embedSelector": true
  },
  "included" : [ {
    "id" : "304",
    "selfUri" : "/querier/api/rest/dataschemas/123/queryschemas/456",
    "type" : "QuerySchema",
    "name" : "Query Schema 0",
    "selectorField" : "title",
    "fields" : [ {
      "name" : "caller",
      "lengthType" : "fixed",
      "size" : 100,
      "maxArrayElements" : 1
    }, {
      "name" : "callee",
      "lengthType" : "fixed",
      "size" : 100,
      "maxArrayElements" : 1
    } ],
  }
```



```
    "dataSchema" : {
      "id" : "7",
      "selfUri" : "/querier/api/rest/dataschemas/123",
      "type" : "DataSchema"
    },
    "queriesUri" :
"/querier/api/rest/dataschemas/123/queryschemas/456/queries"
  }, {
    "id" : "123",
    "selfUri" : "/querier/api/rest/dataschemas/123",
    "type" : "DataSchema",
    "name" : "Phone Records",
    "dataSourcesUri" : "/querier/api/rest/dataschemas/123/datasources",
    "querySchemasUri" :
"/querier/api/rest/dataschemas/123/queryschemas",
    "fields" : [ {
      "name" : "caller",
      "dataType" : "string",
      "isArray" : false,
      "position" : 0
    }, {
      "name" : "callee",
      "dataType" : "string",
      "isArray" : false,
      "position" : 1
    } ]
  }
```

```
}  
}  
]
```

Schedule

Address: /querier/api/rest/dataschemas/{*dataSchemaId*}/queryschemas/{*querySchemaId*}/queries/{*queryId*}/schedules/{*scheduleId*}

Verbs: GET, POST

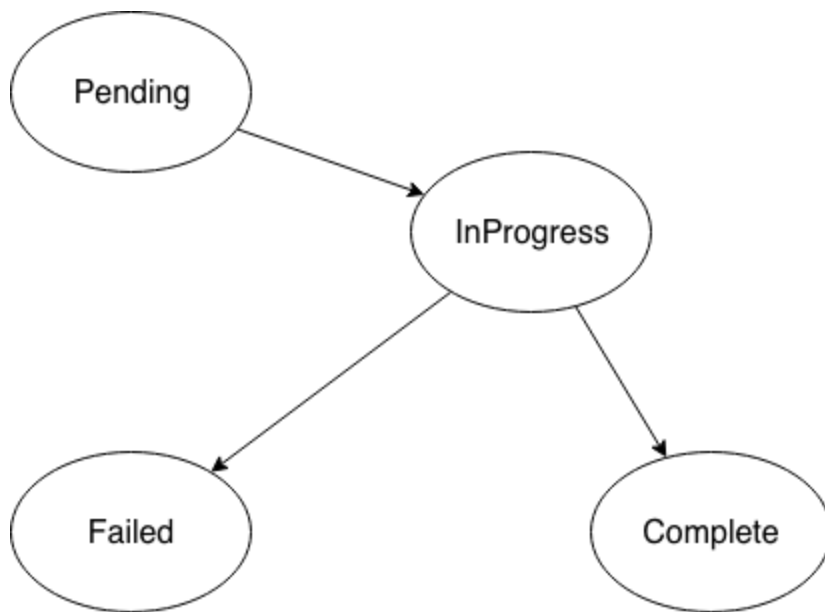
Parameters:

Name	Type	Data Type	Description	Optional
<i>dataSchemaId</i>	path	string	The Data Schema Id	
<i>querySchemaId</i>	path	string	Id of the Query Schema	
<i>queryId</i>	path	string	Id of the Query to retrieve	
<i>scheduleId</i>	path	string	Id of the Schedule to retrieve	GET

Data:

Member	Data type	Description	Omitted if
id	string	Unique id	POST
selfUri	uri	URI to this Schedule	POST
type	string	Type of entity: hard coded to "Schedule"	
startTime	timestamp	When is the query scheduled to run.	
status	string	Status of this schedule: Pending, InProgress, Complete, Failed	POST
errorMsg	string	If status is failed, the error message.	GET('scheduleId' = null)
parameters	Map	Key value additional parameters	GET('scheduleId' = null)
dataSource.id	string	Id of Data Source to Query in this schedule	GET('scheduleId' = null)
dataSource.selfUri	uri	URI to the Data Source on which to run the Query.	POST, GET('scheduleId' = null)
dataSource.type	string	Type of entity: hard coded to "DataSource"	POST, GET('scheduleId' = null)
query.id	string	Id of the parent Query	POST, GET('scheduleId' = null)
query.selfUri	uri	URI to parent Query	POST, GET('scheduleId' = null)
query.type	string	Type of entity: hard coded to "Query"	POST, GET('scheduleId' = null)
resultsUri	uri	URI to list of child results	POST, GET('scheduleId' = null)

Schedule Status transitions:



Examples:

List of Schedules for Query 974

```
GET
/querier/api/rest/dataschemas/123/querieschemas/456/queries/974/schedules
HTTP/1.1
Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK
Content-Type: application/vnd.encryptedquery.enclave+json; version=1

{ "data" : [{
  "id": "1010",
  "selfUri":
"/querier/api/rest/dataschemas/123/querieschemas/456/queries/974/schedules/1010",
  "type" : "Schedule",
  "startTime": "2018-04-23T18:25:43.511Z",
  "status": "Complete"
},{
  "id": "1011",
  "selfUri":
"/querier/api/rest/dataschemas/123/querieschemas/456/queries/974/schedules/1011",
  "type" : "Schedule",
  "startTime": "2018-11-15T14:25:43.511Z",
  "status": "Pending"
}]
}
```

Retrieve Schedule 1010

GET

/querier/api/rest/dataschemas/123/querieschemas/456/queries/974/schedules/1010 HTTP/1.1

Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK

Content-Type: application/vnd.encryptedquery.enclave+json; version=1

```
{
  "data" : {
    "id" : "1010",
    "selfUri" :
"/querier/api/rest/dataschemas/123/querieschemas/456/queries/974/schedules/1010",
    "type" : "Schedule",
    "startTime" : 1534432180687,
    "status" : "Pending",
    "parameters" : {
      "param1" : "value1",
      "windowIterations" : 30,
      "param2" : "value2",
      "windowSizeInSeconds" : 1000
    },
    "resultsUri" :
"/querier/api/rest/dataschemas/123/querieschemas/456/queries/974/schedules/1010/results",
    "query" : {
      "id": "974",
      "type" : "Query",
      "selfUri":
"/querier/api/rest/dataschemas/123/querieschemas/456/queries/974"
    },
    "dataSource" : {
      "id": "875",
      "type" : "DataSource",
      "selfUri": "/querier/api/rest/dataschemas/123/datasources/875"
    }
  },
  "included" : [ {
    "id" : "456",
    "selfUri" : "/querier/api/rest/dataschemas/123/querieschemas/356",
    "type" : "QuerySchema",
    "name" : "Query Schema 456",
    "selectorField" : "title",
    "fields" : [ {
      "name" : "caller",
      "lengthType" : "fixed",
      "size" : 100,
```

```
    "maxArrayElements" : 1
  }, {
    "name" : "callee",
    "lengthType" : "fixed",
    "size" : 100,
    "maxArrayElements" : 1
  } ],
  "dataSchema" : {
    "id" : "123",
    "selfUri" : "/querier/api/rest/dataschemas/123",
    "type" : "DataSchema"
  },
  "queriesUri" :
"/querier/api/rest/dataschemas/123/querieschemas/456/queries"
}, {
  "id" : "974",
  "selfUri" :
"/querier/api/rest/dataschemas/123/querieschemas/456/queries/974",
  "type" : "Query",
  "name" : "Test Query 1",
  "status" : "Scheduled",
  "querySchema" : {
    "id" : "456",
    "selfUri" : "/querier/api/rest/dataschemas/123/querieschemas/456",
    "type" : "QuerySchema"
  },
  "schedulesUri" :
"/querier/api/rest/dataschemas/123/querieschemas/456/queries/974/schedule
s",
  "parameters" : {
    "hashBitSize" : "8",
    "dataChunkSize" : "1"
  },
  "selectorValues" : [ "888-982-9832" ],
  "embedSelector" : true
}, {
  "id" : "123",
  "selfUri" : "/querier/api/rest/dataschemas/123",
  "type" : "DataSchema",
  "name" : "Books",
  "dataSourcesUri" : "/querier/api/rest/dataschemas/123/datasources",
  "querySchemasUri" :
"/querier/api/rest/dataschemas/123/querieschemas",
  "fields" : [ {
    "name" : "caller",
    "dataType" : "string",
    "isArray" : false,
    "position" : 0
  }, {
    "name" : "callee",
```

```
        "dataType" : "string",
        "isArray" : false,
        "position" : 1
    } ]
}, {
    "id" : "875",
    "selfUri" : "/querier/api/rest/dataschemas/123/datasources/875",
    "type" : "DataSource",
    "name" : "Flink-JDBC-Derby-Books",
    "description" : "Flink Engine running on a Derby Phone Records
Database",
    "dataSchema" : {
        "id" : "123",
        "selfUri" : "/querier/api/rest/dataschemas/123",
        "type" : "DataSchema"
    },
    "processingMode" : "Batch"
} ]
}
```

```
}
```

Result

Address: /querier/api/rest/dataschemas/{*dataSchemaId*}/queryschemas/{*querySchemaId*}/queries/{*queryId*}/schedules/{*scheduleId*}/results/{*resultId*}

Verbs: GET

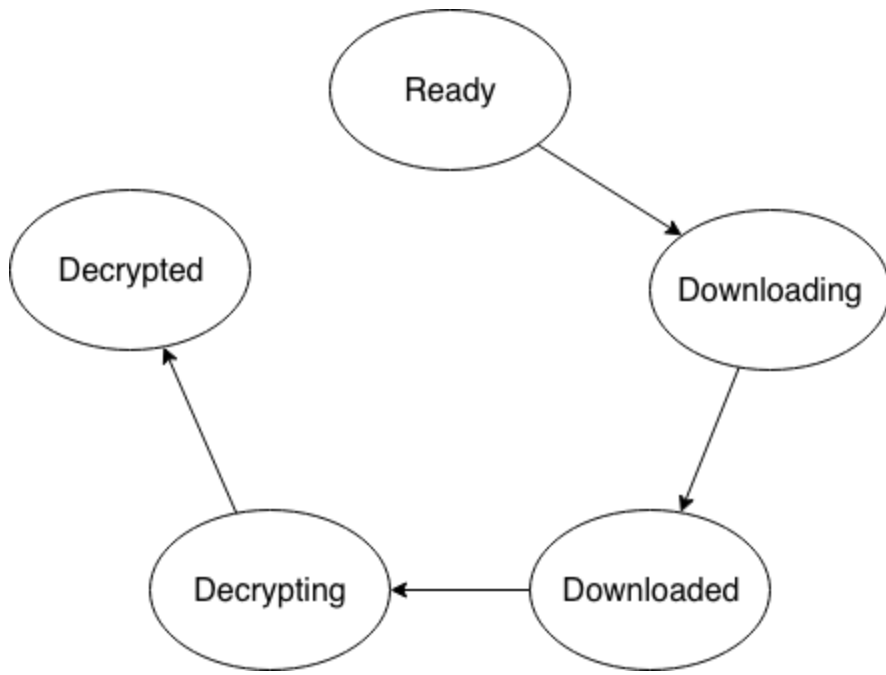
Parameters:

Name	Type	Data Type	Description	Optional
<i>dataSchemaId</i>	path	string	The Data Schema Id	
<i>querySchemaId</i>	path	string	Id of the Query Schema	
<i>queryId</i>	path	string	Id of the Query to retrieve	
<i>scheduleId</i>	path	string	Id of the Schedule to retrieve	
<i>resultId</i>	path	string	Id of the Result to retrieve	GET

Data:

Member	Data type	Description	Omitted if
id	string	Unique id	
selfUri	uri	URI to this Result	
type	string	Type of entity: hard coded to "Result"	
status	string	Status of this Result: Ready, Downloading, Downloaded, Decrypting, Decrypted	
schedule.id	string	Id of parent Schedule	
schedule.selfUri	uri	URI to parent Schedule	GET('resultId' = null)
schedule.type	string	Type of entity: hard coded to "Schedule"	GET('resultId' = null)
retrievalsUri	uri	URI to list of child retrievals	GET('resultId' = null)
windowStartTime	time stamp	Begin time of this result. Milliseconds since January 1, 1970 UTC.	
windowEndTime	time stamp	End time of this result. Milliseconds since January 1, 1970 UTC.	

Result Status transitions:



Examples:

Retrieve List of Results of Schedule 1010

GET

/querier/api/rest/dataschemas/123/querieschemas/456/queries/974/schedules/1010/results HTTP/1.1

Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK

Content-Type: application/vnd.encryptedquery.enclave+json; version=1

```
{ "data" : [{
  "id": "2010",
  "selfUri":
"/querier/api/rest/dataschemas/123/querieschemas/456/queries/974/schedules/1010/results/2010",
  "type" : "Result",
  "status": "Ready",
  "windowStartTime" : 1552662180008,
  "windowEndTime" : 1552662185683
}, {
  "id": "2011",
  "selfUri":
"/querier/api/rest/dataschemas/123/querieschemas/456/queries/974/schedules/1010/results/2011",
  "type" : "Result",
  "status": "Downloading",
  "windowStartTime" : 1552662180008,
  "windowEndTime" : 1552662185683
}]
}
```

Retrieve Result 2010

GET

/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010 HTTP/1.1

Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK

Content-Type: application/vnd.encryptedquery.enclave+json; version=1

```
{ "data" : {
  "id": "2010",
  "selfUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010",
  "type" : "Result",
  "status": "Downloaded",
  "retrievalsUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals"
  "schedule": {
    "id": "1010",
    "selfUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010",
    "type" : "Schedule"
  },
  "windowStartTime" : 1552662180008,
  "windowEndTime" : 1552662185683
},
"included" : [ ....]
}
```

Retrieval

Address: /querier/api/rest/dataschemas/{*dataSchemaId*}/queryschemas/{*querySchemaId*}/queries/{*queryId*}/schedules/{*scheduleId*}/results/{*resultId*}/retrievals/{*retrievalId*}

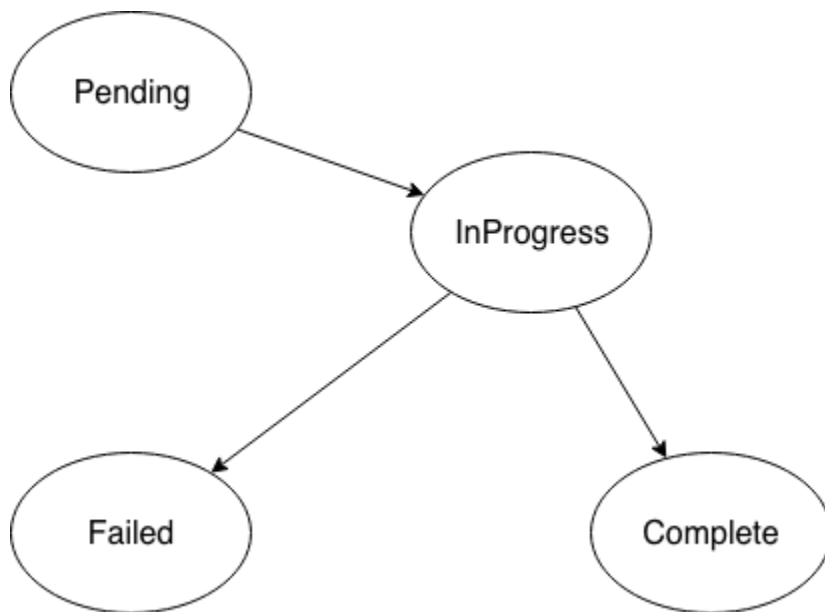
Verbs: GET, POST

Parameters:

Name	Type	Data Type	Description	Optional
<i>dataSchemaId</i>	path	string	The Data Schema Id	
<i>querySchemaId</i>	path	string	Id of the Query Schema	
<i>queryId</i>	path	string	Id of the Query	
<i>scheduleId</i>	path	string	Id of the Schedule	
<i>resultId</i>	path	string	Id of the Result	
<i>retrievalId</i>	path	string	Id of the Retrieval	GET,POST

Data:

Member	Data type	Description	Omitted if
id	string	Unique id	POST
selfUri	uri	URI to self	
type	string	Type of entity: hard coded to "Retrieval"	
status	string	Status of this Retrieval: Pending, InProgress, Complete, Failed	POST
result.id	string	Id of parent Result	GET('retrievalId' = null)
result.selfUri	uri	URI to parent Result	GET('retrievalId' = null)
result.type	string	Type of entity: hard coded to "Result"	GET('retrievalId' = null)
decryptionsUri	uri	URI to list of child decryptions	GET('retrievalId' = null)

Retrieval Status transitions:**Examples:**

Retrieve List of Retrievals of Result 2010

GET

/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals HTTP/1.1

Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK

Content-Type: application/vnd.encryptedquery.enclave+json; version=1

```
{ "data" : [{
  "id": "3010",
  "selfUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals/3010",
  "type" : "Retrieval",
  "status": "Complete"
}, {
  "id": "3011",
  "selfUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2011/retrievals/3011"
  "type" : "Retrieval",
  "status": "InProgress"
}]
}
```

Retrieve Retrieval 3010

GET

/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals/3010 HTTP/1.1

Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK

Content-Type: application/vnd.encryptedquery.enclave+json; version=1

```
{ "data" : {
  "id": "3010",
  "status": "Complete",
  "selfUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals/3010",
  "type" : "Retrieval",
  "result" : {
    "id" : "12",
    "selfUri" :
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010",
    "type" : "Result"
  },
  "decryptionsUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals"
},
"included" : [ ... ]
}
```

Decryption

Address: /querier/api/rest/dataschemas/{*dataSchemaId*}/queryschemas/{*querySchemaId*}/queries/{*queryId*}/schedules/{*scheduleId*}/results/{*resultId*}/retrievals/{*retrievalId*}/decryptions/{*decryptionId*}

Verbs: GET, POST

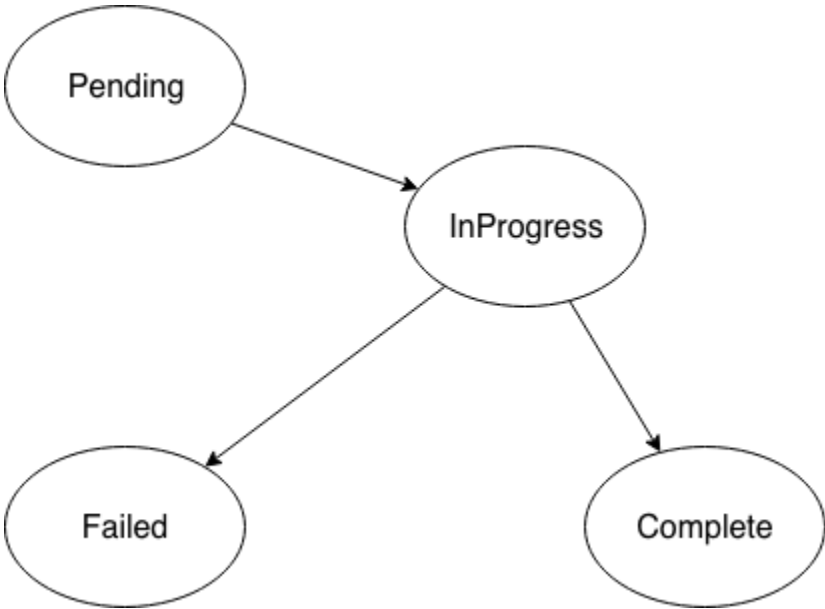
Parameters:

Name	Type	Data Type	Description	Optional
<i>dataSchemaId</i>	path	string	The Data Schema Id	
<i>querySchemaId</i>	path	string	Id of the Query Schema	
<i>queryId</i>	path	string	Id of the Query	
<i>scheduleId</i>	path	string	Id of the Schedule	
<i>resultId</i>	path	string	Id of the Result	
<i>retrievalId</i>	path	string	Id of the Retrieval	
<i>decryptionId</i>	path	string	Id of the Decryption	GET, POST

Data:

Member	Data type	Description	Omitted if
id	string	Unique id of this Decryption	POST
selfUri	uri	URI to self	
type	string	Type of entity: hard coded to "Decryption"	
status	string	Status of this Decryption: pending, complete, failed	POST
retrieval.id	string	Id of parent Retrieval	GET('decryptionId' = null)
retrieval.selfUri	uri	Uri of the parent Retrieval	GET('decryptionId' = null)
retrieval.type	string	Type of entity: hard coded to "Retrieval"	GET('decryptionId' = null)

Decryption Status Transitions



Examples:

Retrieve List of Retrievals of Result 2010

GET

/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals/3010/decryptions HTTP/1.1

Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK

Content-Type: application/vnd.encryptedquery.enclave+json; version=1

```
{ "data" : [{
  "id": "4010",
  "selfUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals/3010/decryptions/4010",
  "type" : "Decryption",
  "status": "Complete"
}, {
  "id": "4011",
  "selfUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2011/retrievals/3011/decryptions/4011",
  "type" : "Decryption",
  "status": "Failed"
}]
}
```

Retrieve Decryption 4010

GET

/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals/3010/decryptions/4010 HTTP/1.1

Accept: application/vnd.encryptedquery.enclave+json; version=1

HTTP/1.1 200 OK

Content-Type: application/vnd.encryptedquery.enclave+json; version=1

```
{ "data" : {
  "id": "4010",
  "selfUri":
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals/3010/decryptions/4010",
  "type" : "Decryption",
  "status": "Complete",
  "retrieval": {
    "id" : 3010,
    "selfUri" :
"/querier/api/rest/dataschemas/123/queryschemas/456/queries/974/schedules/1010/results/2010/retrievals/3010",
    "type" : "Retrieval"
  }
},
"included" : [ ... ]
}
```