

16.2.0 Report Engine for RESTful API Reference

The Reporting Engine for RESTful (the Engine) API is used to provide embedding applications the ability to interact with the Engine server installed at your site. The API provides the means to send JSON or XML requests to the Engine, e.g. a report template and URL to a data source; and to receive JSON or XML responses from the Engine, e.g. output generated from a report template.

We also provide example Engine HTTP client applications for a variety of programming languages.

Engine Request Headers

This article shows request and response bodies using the XML format. However, JSON can be used instead. For JSON, all that is required is to provide the corresponding `Content-Type` and `Accept` headers while sending requests to the Engine. For example, if you send a request to the 'v1/version' resource (as shown below), and you would like to receive the response as a JSON string, add `Accept: application/json` to the request's header.

XML and JSON request headers

- Send an XML request to the Engine: `Content-Type: application/xml`
- Receive an XML response from the Engine: `Accept: application/xml`
- Send a JSON request to the Engine: `Content-Type: application/json`
- Receive a JSON response from the Engine: `Accept: application/json`

Reporting Engine for RESTful API

Some APIs require a request body be sent to the resource as well as an HTTP verb. Those are specified below.

Every API returns an HTTP response code to the requesting application. Many APIs also return an XML or JSON response body (such as a report template's output); both are also specified below.

Resource URLs

Resource	Relative URL	HTTP Verb
General	v1/version	GET
Reports	v1/metrics	POST
Reports	v1/reports	POST
Reports	v1/reports/ <i>GUID</i>	GET
Reports	v1/reports/ <i>GUID</i> /status	GET
Reports	v1/reports/ <i>GUID</i>	DELETE

Resource Operations

i If you send the Engine a POST request with the **Async** element set to *true*, you'll receive a response from the Engine with a generated output **GUID**. That GUID can be used in later requests to: fetch a template's output; query a template's output status; or delete a template's output.

Engine Errors


In general, if a request sent to the Engine resulted in an error, the Engine will return an HTTP status code like 404 (Not Found), and a response body with an error. For example, the Engine returns this error when the URI for a template in a POST request to the resource 'v1/reports' is incorrect:

```
<Error>
<Message>An error has occurred.</Message>
<ExceptionMessage>The remote server returned an error: (404) Not Found.
</ExceptionMessage>
<ExceptionType>System.Net.WebException</ExceptionType>
<StackTrace>    at System.Net.WebClient.DownloadDataInternal(Uri address, WebRequest&
request)
at System.Net.WebClient.DownloadData(Uri address)
at RESTfulEngine.BusinessLogic.d.a(Byte[] A_0, String A_1)
at RESTfulEngine.BusinessLogic.d.b(Template A_0)
```

```

at RESTfulEngine.Controllers.ReportsController.GenerateReport(Template template)
at RESTfulEngine.Controllers.ReportsController.ProcessTemplate(Template template)
at RESTfulEngine.Controllers.ReportsController.Post(Template template)</StackTrace>
</Error>

```

 Click to copy

General Resource Operations

Relative URL	v1/version
HTTP Verb	GET
Description	Get the version numbers of the Engine service and underlying Reporting Engine.
Response Code	200 (OK)
Request Body Required?	No
Returns Response Body?	Yes

Response Body Example

```

<Version>
<ServiceVersion>15.2.288.0</ServiceVersion>
<EngineVersion>15.2.288.0</EngineVersion>
</Version>

```

Reports Resource Operations

Relative URL	v1/metrics
HTTP Verb	POST
Description	Returns information stored in the template including data source, input parameters, file type and other information.
Response Code	200 (OK)
Request Body Required?	No
Returns Response	Yes

Body?

Response Body Example

```
<Metrics xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <TemplateType>docx</TemplateType>
  <Datasources>
    <string>MANF_DATA_2009</string>
  </Datasources>
  <Tags>
    <Tag>
      <XmlType>12</XmlType>
      <Text><wr:out select='/Root/Overhead/ForTheYearEnded' type='DATE'
format='category:date;type:0;format:m/d/yyyy;' nickname='[Year Ending Date]'
datasource='MANF_DATA_2009' /></Text>
      <NodeName>Out</NodeName>
      <HasNode>true</HasNode>
      <Node>/Root/Overhead/ForTheYearEnded</Node>
    </Tag>
  </Tags>
  [...]
</Metrics>
```

Relative URL	v1/reports
HTTP Verb	POST
Description	Generate output from a report template. All desired output generation parameters should be coded within the request's body.
Response Code	200 (OK)
Request Body Required?	Yes
Returns Response Body?	Yes

Request Body Example

This simple request body will return the template's output in a response body. It includes the URIs of the report template and data source; the data source nickname; the data source type; and the output format. Send this POST request with the `Content-Type: application/xml` header.

Note the URIs are only an example. They must be changed to reflect your RESTful Engine installation.

```
<Template>
```

```

<Uri>http://windward.restfulengine.net:8080/SampleTemplates/Manufacturing.docx</Uri>
<OutputFormat>pdf</OutputFormat>
<Async>false</Async>
<Datasources>
  <Datasource>
    <Name>MANF_DATA_2009</Name>
    <Type>xml</Type>
    <Uri>http://windward.restfulengine.net:8080/SampleTemplates/Manufacturing.xml</Uri>
  </Datasource>
</Datasources>
</Template>

```

Response Body Example

The Engine delivers output from a report template as a Base64-encoded string within a response body (when the **Async** element is set to *false*). To receive an XML response body use the `Accept: application/xml` header in the POST request.

```

<Report xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Data>JVBERi0xLj [...] JUVPRgo=</Data>
</Report>

```

Relative URL	v1/reports/ <i>GUID</i>
HTTP Verb	GET
Description	Retrieve output generated asynchronously. <i>GUID</i> was returned by a previous asynchronous POST request.
Response Code	200 (OK) 202 (Accepted) - if the output generation is still in progress 404 (Not Found) - if the requested output doesn't exist
Request Body Required?	No
Response Body Returned?	Yes - same as a POST to 'v1/reports' above.

Relative URL	v1/reports/ <i>GUID</i>
HTTP Verb	DELETE
Description	Delete the previously asynchronously-generated output.
Response Code	200 (OK)

	404 (Not found) – if the requested report does not exist
Request Body Required?	No
Returns Response Body?	No

Relative URL	v1/reports/ <i>GUID</i> /status
HTTP Verb	GET
Description	Get the status of asynchronously-generated output.
Response Code	200 (OK) – the output is ready 202 (Accepted) – the output generation is still in progress 404 (Not found) – the requested output does not exist 500 (Internal server error) – an error occurred during output generation
Request Body Required?	No
Returns Response Body?	No

Request Body Elements

Here are the XML elements that can be included in an XML request body.

```
<Template>

  <!-- The source of the template - embedded or external. -->
  <!-- Embed template as a Base64-encoded string. -->
  <Data/>
  <!-- The location of the external template. -->
  <Uri/>

  <!-- Generate the report in the provided format. -->

<OutputFormat>pdf|docx|xlsx|pptx|html|csv|rtf|jpg|png|svg|eps|bmp|gif</OutputFormat>

  <!-- Generate the report asynchronously. The report can be queried or obtained
  later with additional requests. -->
  <Async>true|false</Async>

  <!-- This is for Windward Scout installations only. Ignore if not Windward Scout. --
```

```

>
<ApiKey/>

<!-- Format of the template. Auto-determined if not provided. -->
<Format>docx|xlsx|pptx</Format>

<!-- Data source description section. These elements will be applied in order.
Note: Use SQL elements for SQL-only data sources. -->
<Datasources>
  <Datasource>
    <!-- The nickname of the data source you specified in the Connection Editor
in Report Designer. Use the value NoName if the data source name within the template is
empty or absent. -->
    <Name/>

    <!-- The data source type. -->
    <Type>sql|xml|xml2|json|odata|salesforce|salesforceoauth</Type>

    <!-- For SQL data sources -->
    <ClassName>System.Data.SqlClient</ClassName>

    <!-- The connection string from the Connection Editor when you set up the
data source in Report Designer. -->
    <ConnectionString/>

    <!-- For XML, JSON, or OData data sources - embedded or external -->
    <Data/> <!-- Embed data source as a Base64-encoded string. -->
    <Uri/> <!-- The location of the external data source. -->

    <!-- The user access token and soap endpoint for salesforceoauth -->
    <SalesforceAccessToken/>
    <SalesforceSoapEndpoint/>

    <!-- The user credentials for JSON and OData data sources (optional). -->

    <Username/>
    <Password/>

    <!-- The domain name for JSON and OData data sources (optional). -->
    <Domain/>

    <!-- The OData version. The Engine will use 1 if not provided. -->
    <ODataVersion>1|2|3|4</ODataVersion>

    <!-- The OData protocol. The Engine will use identity if not provided. -->
    <ODataProtocol>identity|basic|credentials|windowsauth</ODataProtocol>

    <!-- If an XSD file is used with your XML data source, use one of the
following. -->
    <!-- Embed the XSD file as a Base64-encoded string. -->

```

```

        <SchemaData/>
        <!-- Location of external XSD file. -->
        <SchemaUri/>

        <!-- Variable section (must be within the <Datasources/> element). List of
variables to use with the data source. -->
        <Variables>
            <Variable>
                <!-- The name of the variable. -->
                <Name/>
                <!-- The type of variable. The Engine will use text if not provided.
-->
                <Type>text|int|float|datetime</Type>
                <!-- The variable's value. -->
                <Value/>
            </Variable>
        </Variables>
    </Datasource>
</Datasources>

    <!-- Settings section. These settings can be applied to the output being generated.
Apply all that are needed. These settings correspond to members of the .NET Reporting
Engine Report class. -->

    <!-- Template's version. Auto-determined if not provided. -->
    <Version/>
    <CopyMetadata>never|nodatasource|always</CopyMetadata>
    <Description/>
    <!-- Name of the output file. -->
    <Title/>
    <Subject/>
    <Keywords/>
    <Hyphenate>off|template|on</Hyphenate>
    <Locale/>
    <TrackImports/>
    <Timeout/>
    <RemoveUnusedFormats/>
    <License/>
</Template>

```

Example POST Request

Here's an example of an XML RESTful POST request body to output the Windward 'Manufacturing' sample. (Note the URIs are only an example. They must be changed to reflect your RESTful Engine installation.):


```

<Template>
  <Uri>http://windward.restfulengine.net:8080/SampleTemplates/Manufacturing.docx</Uri>
  <OutputFormat>pdf</OutputFormat>
  <Async>false</Async>
  <Datasources>
    <Datasource>
      <Name>MANF_DATA_2009</Name>
      <Type>xml</Type>
      <Uri>http://windward.restfulengine.net:8080/SampleTemplates/Manufacturing.xml</Uri>
    </Datasource>
  </Datasources>
  <CopyMetadata>always</CopyMetadata>
  <Description>PDF sample report</Description>
  <Title>testreport.pdf</Title>
  <Subject>Internet Marketing Report sample</Subject>
  <Keywords>none</Keywords>
  <Hyphenate>on</Hyphenate>
  <TrackImports>false</TrackImports>
  <Timeout>10</Timeout>
  <RemoveUnusedFormats>false</RemoveUnusedFormats>
  <License>licenseKey</License>
</Template>

```

i Avoid any empty elements in your XML POST request body, e.g. "<Title></Title>". Always provide a relevant value as in the example above.

Example Clients for the Engine

There are a variety of HTTP clients available you can use to interact with the Engine, including [curl](#) and [Postman](#). We also provide several example Engine HTTP clients in the Engine installation for different programming languages:

- **[Report Engine for RESTful C# Client]**
- **[Report Engine for RESTful Ruby Client]**
- **[Report Engine for RESTful Java Client]**
- **[Report Engine for RESTful PHP Client]**
- **[Report Engine for RESTful Python Client]**
- **[Report Engine for RESTful Typescript/JavaScript Client]**

Changes in Version 16.3.0

In version 16.3.0 additional output options were added to the RESTful Engine. The following formats are supported in version 16.3.0 and later:

- JPG
- PNG
- SVG
- EPS
- BMP
- GIF