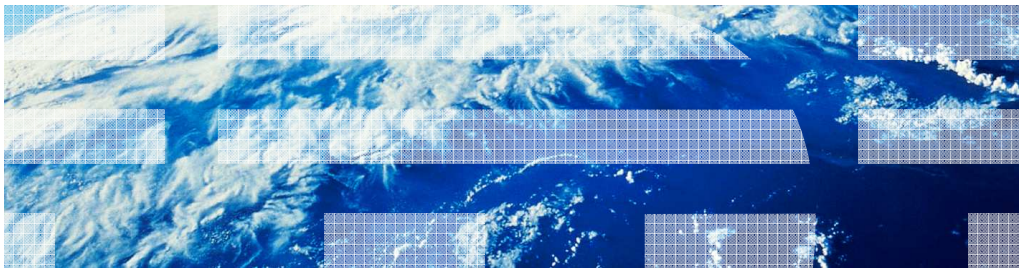
IBM Operational Decision Manager V8.0.1

Decision Server REST API



IBM Operational Decision Manager version 8.0.1 provides a set of APIs that are implemented using Representational State Transfer (REST) services.

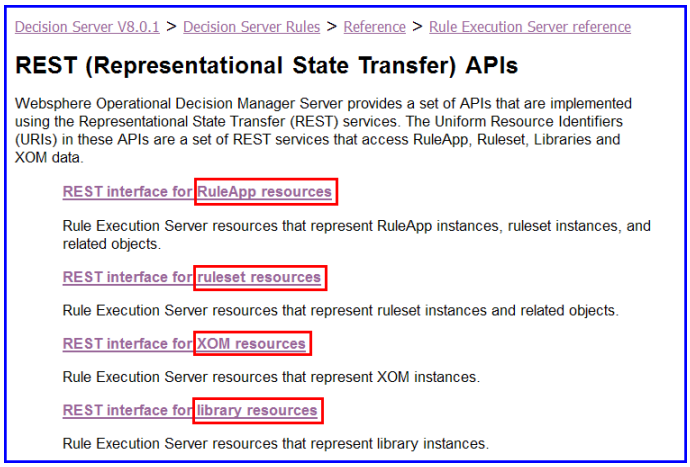
These REST APIs make it easy for you to manage the resources over a remote Rule Execution Server instance through cross-platform access or by JavaScript clients.

Table of contents

The session starts with a brief introduction to the Rule Execution Server REST APIs. Then you are presented with the resource URI format, the supported request parameters, content types, and response status codes. Next you learn the WADL representation of the REST API request and response elements. Finally you learn to use a REST API test tool that is built in the Rule Execution Server console.

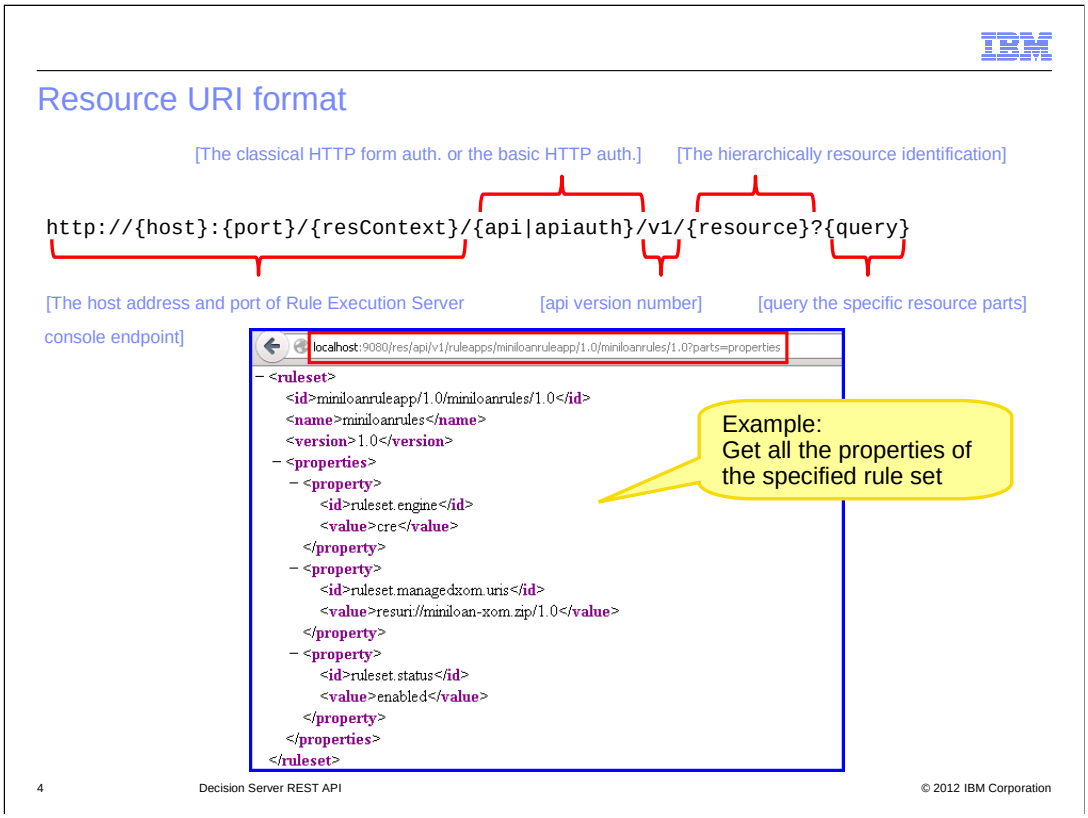
Introduction

- REST APIs: A set of APIs that are implemented using the Representational State Transfer services
- Rule Execution Server REST APIs: URIs that are used to manage the resources on the remote rule execution server
- Resources to access
  - Rule application
  - Rule set
  - XOM
  - Library



The REST APIs are referred to a set of APIs that is implemented by Representational State Transfer services. In Operational Decision Manager version 8.0.1, the newly added Rule Execution Server REST API provides you a simpler way to develop your own client application to work with resources running on the remote Rule Execution Server. Comparing to the other supported APIs, such as JMX API, there is no need to add client library or set complex configuration to use the REST APIs.

The developers can use the REST API to access four types of resources: the rule applications, the rule sets, the execution object models and the libraries.



Each of the Rule Execution Server resources are identified by a unique URI. On the top of the slide, it shows the format of the resource URI.

The first part contains the host address and port number of the Rule Execution Server console endpoint.

The middle “api” or “apiauth” section is to set the REST API context root. If you choose the classic HTTP form authentication, use the "api" parameter. Otherwise, use the "apiauth" parameter for the basic HTTP authentication method.

The "v1" is the current version of the REST API.

Next is the "resource" part. It should contain the hierarchical part of the resource identification. In the example URL, between the version section and the question mark, the resource parameter points to a specific rule set in a rule application in the hierarchical order.

The last "query" contains the query parameters that can be passed to the resource. In the example, it asks for the properties of the rule set.

Request parameters

- HTTP methods:
  - Post for create operation
  - Get for read operation
  - Put for update operation
  - Delete for delete operation
- HTTP header fields:
  - Accept
  - Accept-Language
  - Content-Type
  - X-Method-Override or X-HTTP-Method-Override
- Generic URI parameters:
  - accept
  - accept-language
  - parts use vertical line to separate multiple parts. Example: "parts=name|version|properties"
  - date
- Special parameters for a collection of resources
  - count returns the number of the element
  - *fieldName* filters the result list by key/value pair
  - orderby sorts the result list

The supported operations to work with Rule Execution Server resources are create, read, update, and delete. Their corresponding HTTP methods are post, get, put, and delete.

The REST API URIs also support some HTTP header fields and generic URI parameters. They are listed in the slide. Some of them, such as "accept" and "accept-language," are interchangeable. And you can add multiple parameters together on one resource in a request. When you use "parts" URI parameter to specify a list of request parts, use the vertical lines to separate them, as shown in the example.

To better organize a collection of the resources in the returned response, you can add some special parameters. The count parameter tells you the number of the resources in the collection. You can add key-value pairs as query parameters to filter the results. You can also sort the result list with pagination by using "orderby" parameter.

Content type and response status code

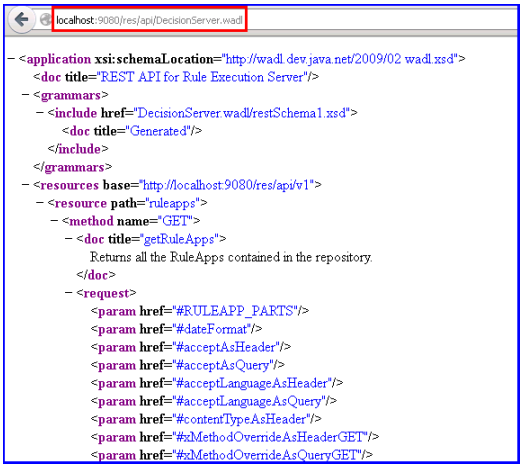
- Content-Type
  - application/xml (default)
  - application/json
  - application/octet-stream (file type)
- Status codes
  - Success
    - 200 OK
    - 201 Created
    - 204 No Content
  - Error
    - 400 Bad Request ← Invalid or missing request parameters
    - 401 Unauthorized
    - 403 Forbidden
    - 404 Not Found
    - 406 Not Acceptable ← unsupported content type or encoding
    - 412 Invalid syntax ← The regular expression specified in the *fieldname* parameter is not valid for collection types
  - Unexpected Error
    - 500 Internal Server Error

The Content-Type field set in HTTP header indicates the media type of the data, that is sent in request, or returned in response. XML is the default content type of the response. Other supported content types are the JavaScript Object Notation object and the file type.

The response status code informs you of the failure or success of the REST request. The listed codes are all common standard HTTP status codes.

WADL representation

- Shows the details of the request or response elements using REST API
  - Help you to monitor the development and documentation synchronization
- URL to access: `http://{host}:{port}/{resContext}/{api|apiauth}/DecisionServer.wadl`
  - To show inline XSD in the WADL code, add `inline` query parameter
    `http://{host}:{port}/{resContext}/{api|apiauth}/DecisionServer.wadl?inline=true`



WADL stands for “Web Application Description Language.” It is typically used to provide a machine process-able description of HTTP-based web applications.

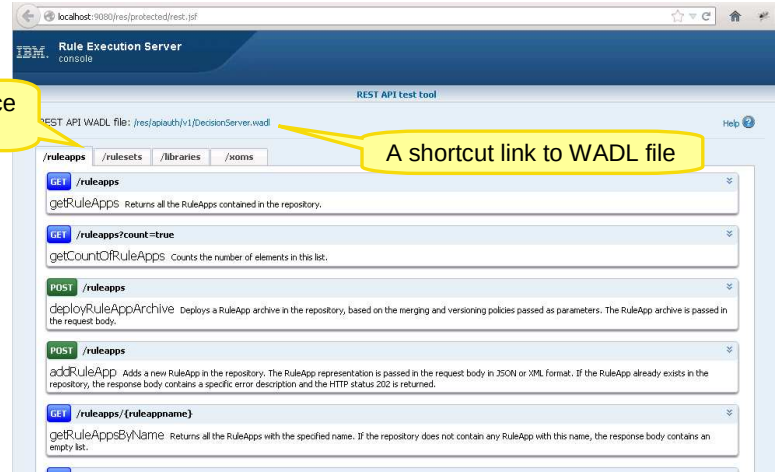
You can use the format of REST resource URIs to retrieve the WADL representation of the REST API. It is commonly used when you develop your client application using the REST APIs. The detailed request and response element information in the WADL file helps you to keep the synchronization between development and documentation.

To include the XSD in the WADL code, add the “inline” query parameter.

Web-based test tool

- Link available in Rule Execution Server console

- The tool is launched in a new window



8 Decision Server REST API

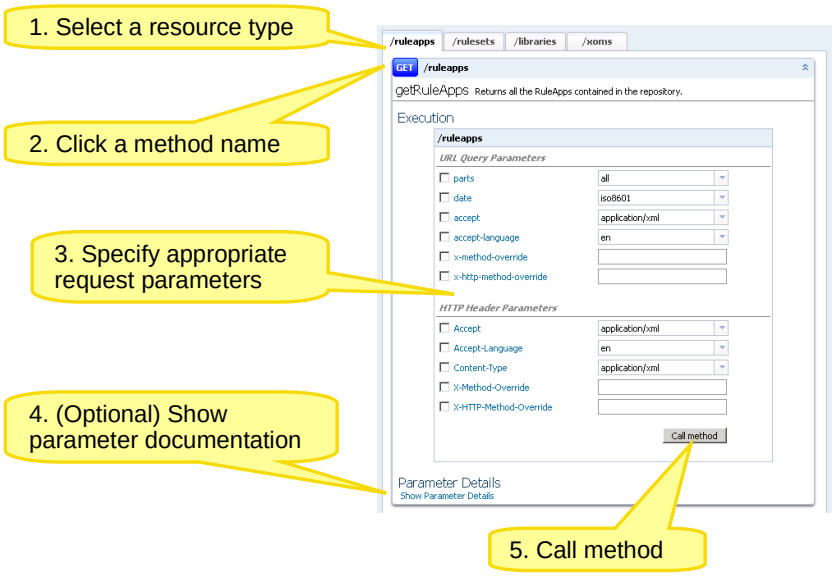
An interactive test tool is provided with the Rule Execution Server REST API. You can use it to learn more about the API and test the requests that you plan to use in your application.

To start the tool, click the “REST API” link on the top right menu of the Rule Execution Server console. The tool is launched in a separate browser tab.

A shortcut link to the REST API WADL file is provided at the top of the test tool page for your convenience. You can use it to check the documentation of the methods and parameters while you write requests in the graphical tool.

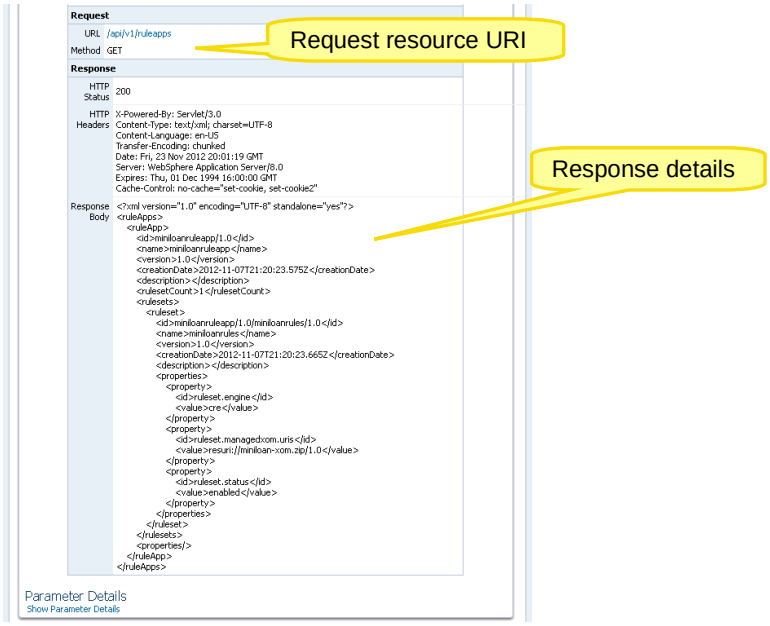
The test tool contains four tabbed resource pages. Each page contains available methods that you can test with the selected resource and its associated parameters.

Using the test tool

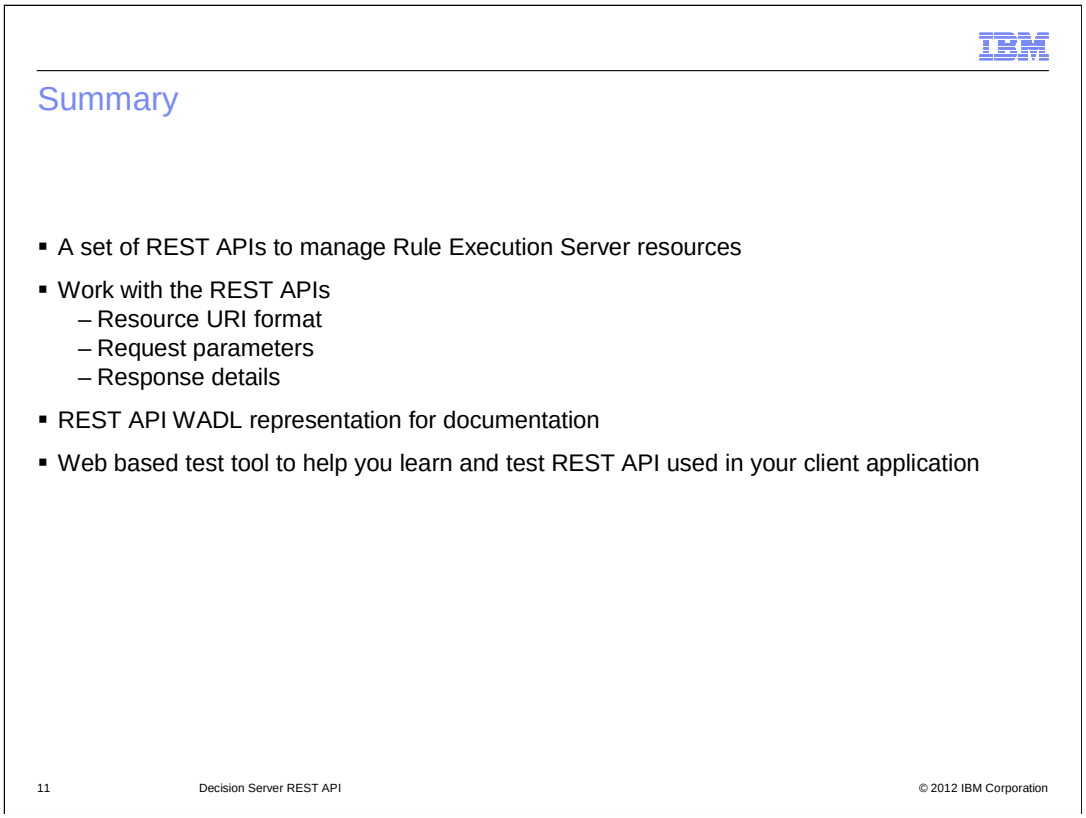


To use the REST API test tool, you first select a resource type by clicking the resource tab. The colored method name is a toggle button to expand or collapse the whole method parameter area. Once the area is opened, you specify the appropriate parameters in the request. You can always click the “Show Parameter Details” link at the bottom of the method area to check the list of available parameters and their documentation. Once all parameters are set, click the **Call method** button to send the request.

Response result in the test tool



The response result is extended in the method area to display the request URI and response details. You can see from the example in the screen capture, the get method returns successful HTTP status code and headers. In the response body area, the rule apps are displayed in XML format with all the parts included.



IBM Operational Decision Manager version 8.0.1 provides a set of REST APIs to manage the resources over the remote Rule Execution Server.

In this session, you first learned the REST API resource URI format, supported HTTP method, various request parameters, and the content type and status codes.

Then you were introduced to the REST API WADL representation. Furthermore with a step by step walkthrough, you learned how to use the REST API web tool to test the REST requests and check the responses that are used in your client application.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_ODM801_DecisionServerRestAPI.ppt

This module is also available in PDF format at: ../ODM801_DecisionServerRestAPI.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information