An Introduction to REST Services in Taverna

Stian Soiland-Reyes and Christian Brenninkmeijer
University of Manchester

materials by Katy Wolstencroft and Aleksandra Pawlik

http://orcid.org/0000-0001-9842-9718
http://orcid.org/0000-0002-2937-7819
http://orcid.org/0000-0002-1279-5133
http://orcid.org/0000-0001-8418-6735

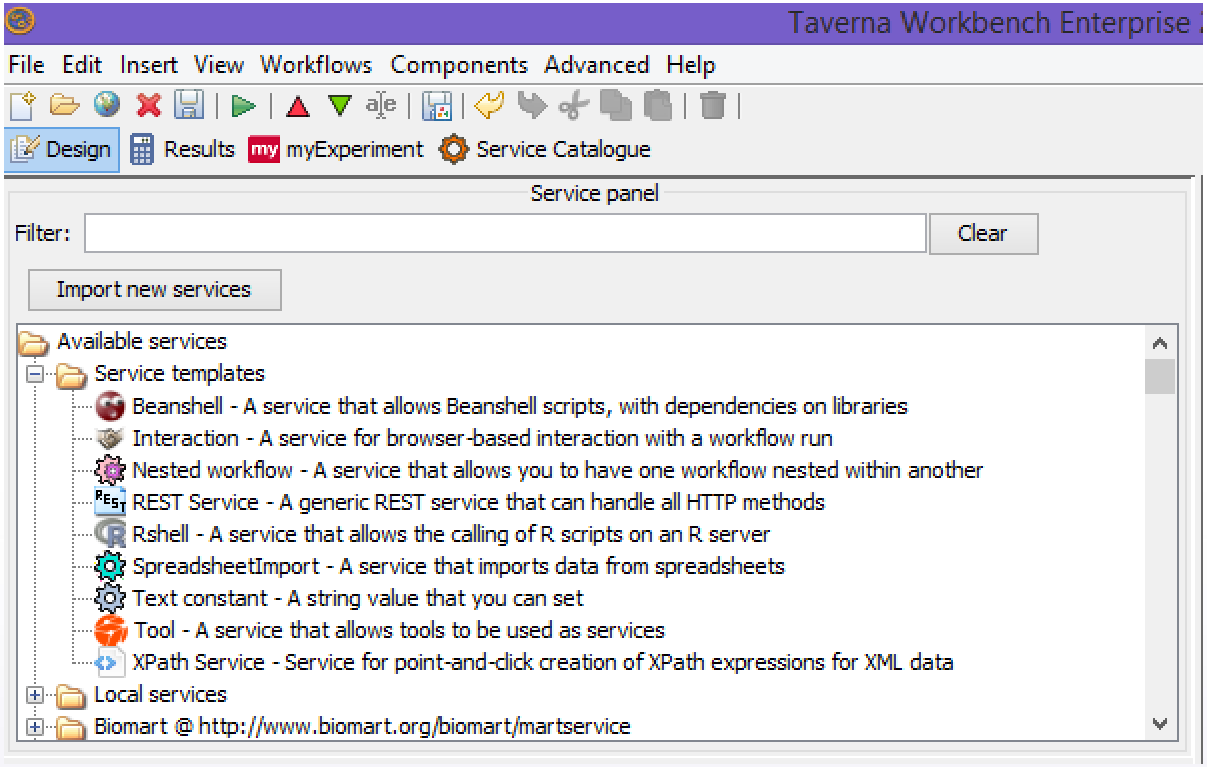
Bonn University, 2014-09-01
http://www.taverna.org.uk/

REST Services

Data and tools on the Web have been exposed in a RESTful manner. Taverna provides a custom processor for accessing such services.

- REST stands for REpresentational State Transfer
- Web services with this type of interface typically expose some of the following four types of operations:
  - GET - to get a resource
  - POST - to make a new resource or to perform a request (such as search)
  - PUT - to update a resource
  - DELETE - to delete a resource

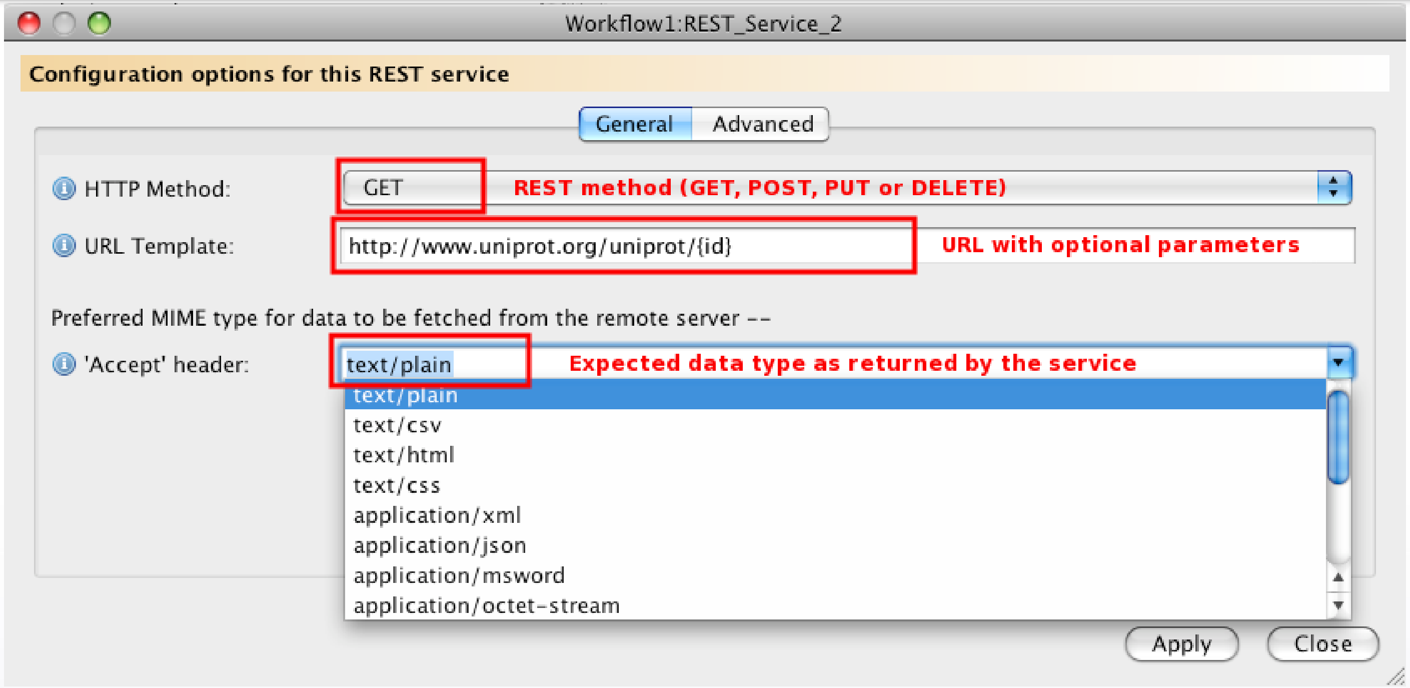
Adding a REST web service



- Available services
  - Service templates
    - Beanshell - A service that allows Beanshell scripts, with dependencies on libraries
    - Interaction - A service for browser-based interaction with a workflow run
    - Nested workflow - A service that allows you to have one workflow nested within another
    - REST Service - A generic REST service that can handle all HTTP methods
    - Rshell - A service that allows the calling of R scripts on an R server
    - SpreadsheetImport - A service that imports data from spreadsheets
    - Text constant - A string value that you can set
    - Tool - A service that allows tools to be used as services
    - XPath Service - Service for point-and-click creation of XPath expressions for XML data
  - Local services
  - Biomart @ http://www.biomart.org/biomart/martservice

- Expand the Service templates folder under Available services in the Service Panel. Select the REST service template and drag n drop it into the Workflow Diagram

Configuring a REST Service



Workflow1:REST_Service_2

Configuration options for this REST service

HTTP Method: GET

REST method (GET, POST, PUT or DELETE)

URL Template: http://www.uniprot.org/uniprot/{id}

URL with optional parameters

Preferred MIME type for data to be fetched from the remote server --

'Accept' header: text/plain

Expected data type as returned by the service

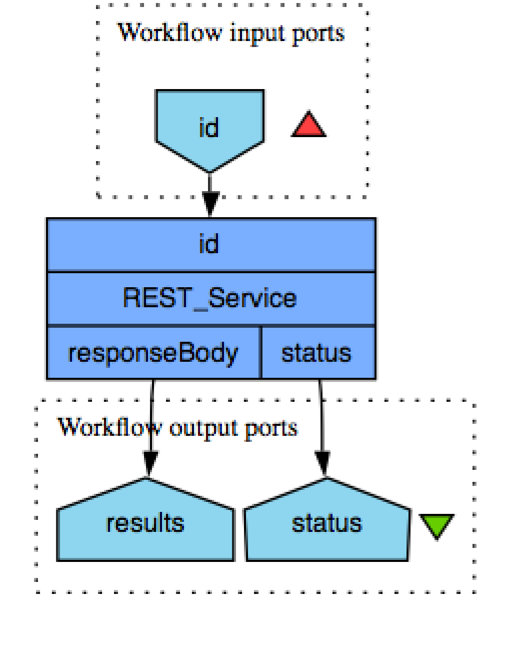
- In a dialog box that pops up, configure the use of a REST service by adding the:
  - URL of the service, *e.g.* http://www.uniprot.org/uniprot/{id}
  - Type of the operation you want to perform on the service, e.g. GET
  - The expected MIME data type as returned by the services. Select text/plain in this case.

Configuring a REST Service

- The URL of the service you enter is actually a template that can take configurable parameters
- In the current example, the name of the parameter is ‘id’ which is enclosed within a pair of braces
- Parameters are used when you do not know their value in advance, *i.e.* at the time of adding the service to the workflow diagram, and they depend on some of the previous services in the workflow.
- The value in braces will then be replaced with the actual value when the workflow is executed.

Building a simple workflow using a REST service

- To complete the building of a simple workflow using the Uniprot REST web service, add:
  - a workflow input for the REST service input port named ‘id’
  - workflow outputs for the responseBody and status output ports of the REST service activity
- The workflow should now look as follows:

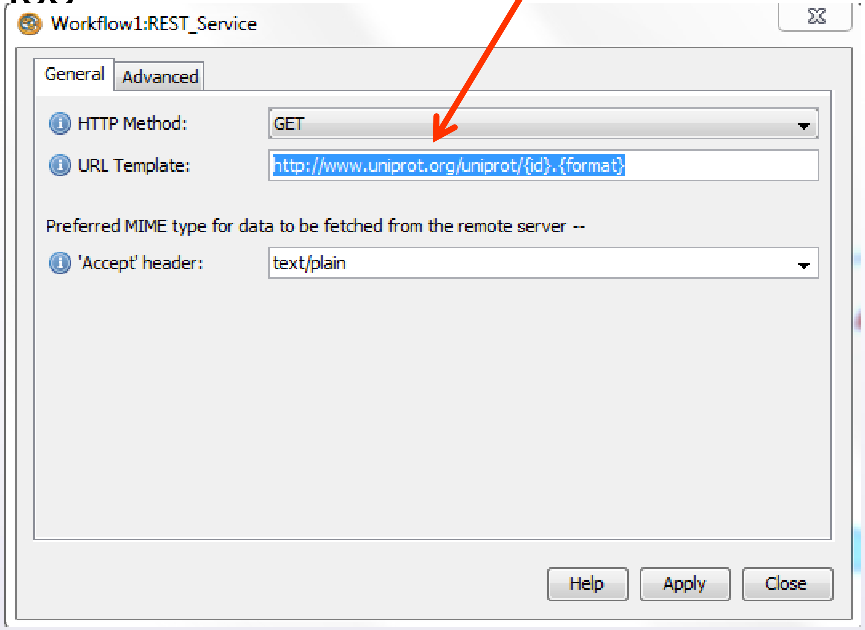


Building a simple workflow using a REST service

- Now try running the workflow using an example Uniprot identifier such as Q99102.fasta
- Note that the presence of the .fasta suffix enables the protein sequence to be returned as text
- What happens when you don’t use the .fasta suffix?
- Try using other suffixes, e.g. xml, txt, rdf and gff
- Further information about the Uniprot REST service is available from http://www.uniprot.org/faq/28

Configuring a REST service

- Let’s add the format (xml, fasta and so on) as another parameter
- Open the REST service configuration and add “.{format}” to the URL Template
  - Hint: In “Design” view, right click on the REST_Service and select Configure REST Service



HTTP Method: GET

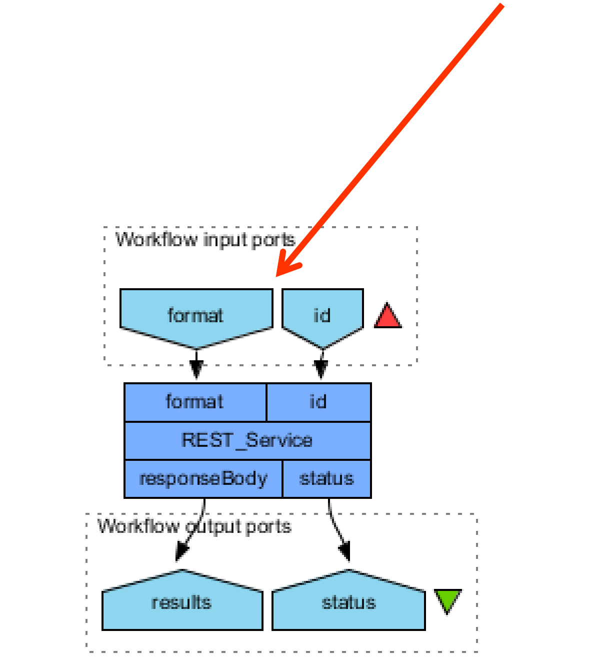
URL Template: http://www.uniprot.org/uniprot/{id}.{format}

Preferred MIME type for data to be fetched from the remote server --

'Accept' header: text/plain

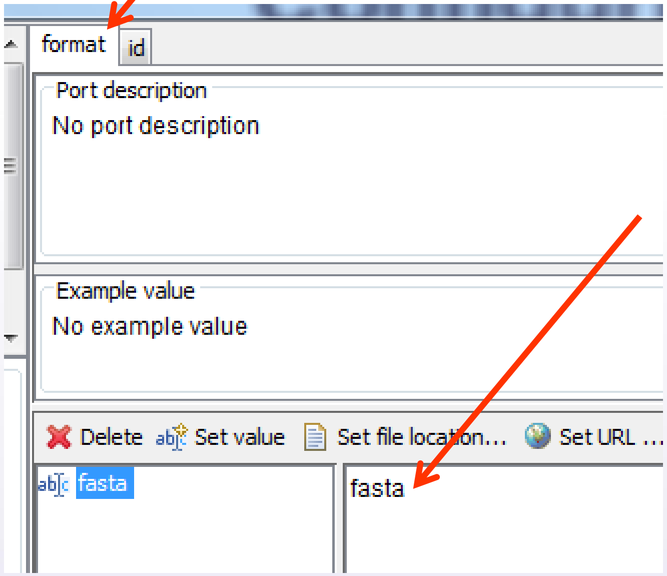
Configuring a REST service

- □ Now we need to add an input port to the workflow



Configuring a REST service

- Now when we run the workflow we need to specify the format as an input, the id must not include the format

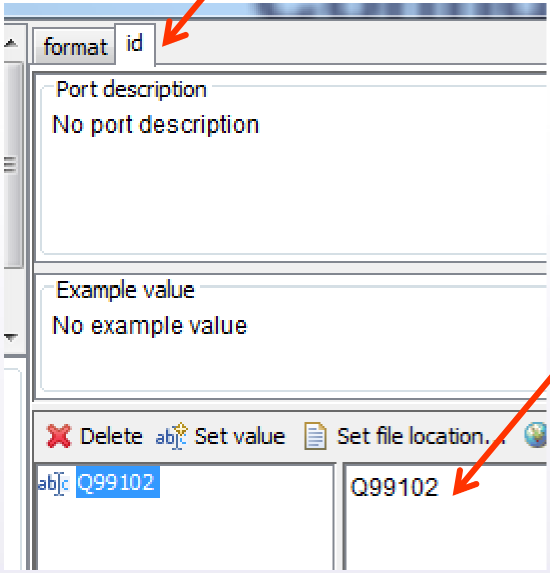


Port description
No port description

Example value
No example value

fasta

fasta



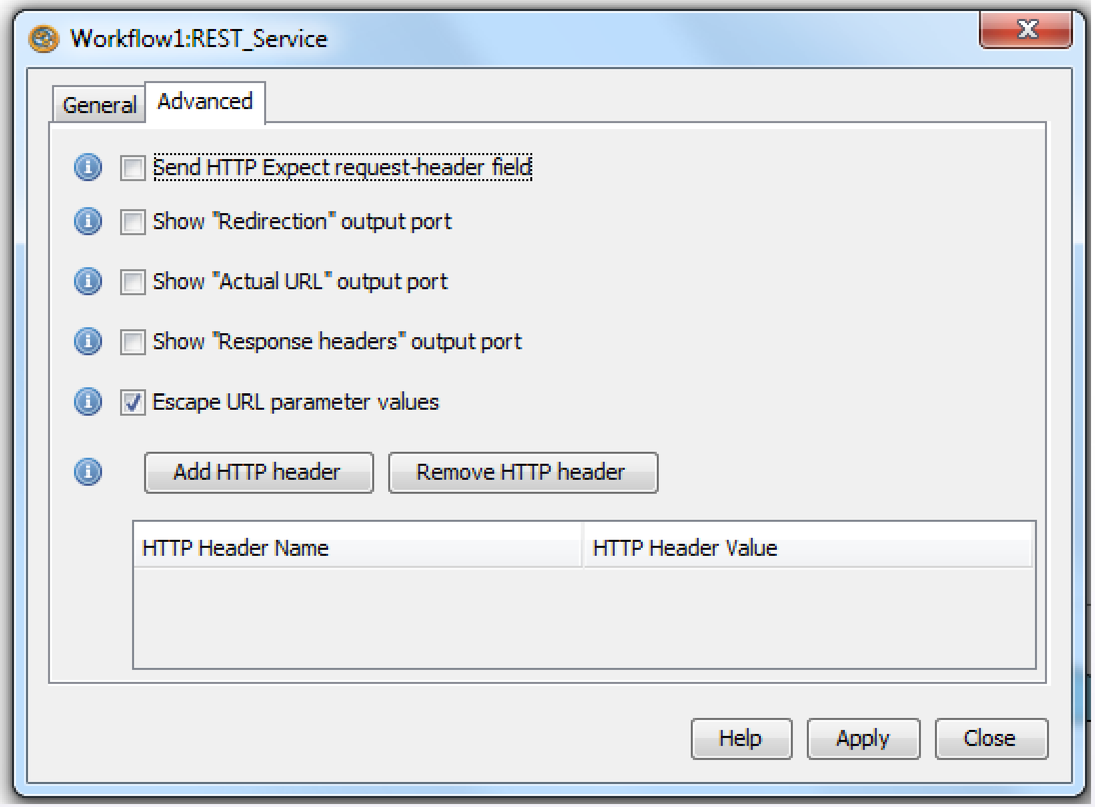
Port description
No port description

Example value
No example value

Q99102

Q99102

Advanced configuration of REST services



Workflow1:REST_Service

☐ Send HTTP Expect request-header field

☐ Show "Redirection" output port

☐ Show "Actual URL" output port

☐ Show "Response headers" output port

☑ Escape URL parameter values

HTTP Header Name	HTTP Header Value

Advanced configuration of REST services

- Send HTTP Expect request-header
  - This field option allows Taverna to set a special "Expect" header when sending a request to the REST service. Client requests using the POST method will expect to receive a 100-Continue or Redirect response from the service to indicate that the client should proceed to send the POST data.
  - This mechanism allows clients to avoid sending large amounts of data over the network twice when the service could reject or redirect the request. Selecting this option may significantly improve performance when large volumes of data are to be sent to the service and authentication or a redirect from the original URL to the one specified by the service is likely.

Advanced configuration of REST services

- Redirection output port
  - The Show "Redirection" output port option makes the service's redirection output port visible as this output port is hidden by default. The port will contain the URL of the final redirect that has yielded the output data on the responseBody port.

- Show "Actual URL" output port
  - If this box is selected, then the REST service will have a port called "actualURL". When the service is invoked, the URL that is used as a result of replacing all the parameter values is output on the port.

Advanced configuration of REST services

- Show "Response headers" output port
  - If this box is selected, then the REST service will have a port called "responseHeaders". When the service is invoked, the list of HTTP headers from the REST service call will be output on the port. Note that if the calling of the service includes redirection, then it is the final set of HTTP headers that is output.

- Escape URL parameter values
  - If this box is selected (the default) then when values are passed to the REST input and those values are included in the URL, they are escaped so that they can form part of a valid URL. It is not a good idea to uncheck this box.

Further exercises

- There are many more REST services available in bioinformatics
- Try to find these using BioCatalogue and test them using Taverna