

# Taverna Workbench JSON Plug-in (0.0.1-SNAPSHOT)

---

Mark Borkum (University of Southampton)

[m.i.borkum@soton.ac.uk](mailto:m.i.borkum@soton.ac.uk) / [@markborkum](https://twitter.com/markborkum) / [github.com/markborkum](https://github.com/markborkum)

## Acronyms

JSON – JavaScript Object Notation

T2 – Taverna Workbench version 2.x

## Table of Contents

Abstract.....	2
1 JSON Template.....	3
1.1 Adding and Configuring a “JSON Template” Service.....	4
1.1.1 Adding and Configuring an Input Port.....	5
1.1.2 Adding and Configuring an Output Port.....	6
2 JSONPath.....	7
2.1 Adding and Configuring a “JSONPath” Service.....	8
2.1.1 Selecting a JSONPath Expression.....	9

## List of Figures

Figure 1: Taverna Workbench "Service panel" component with "JSON Template" item. ....	3
Figure 2: Taverna Workbench "Insert" menu with “JSON Template” item. ....	3
Figure 3: Configuration dialog for "JSON Template" input port. ....	5
Figure 4: Configuration dialog for "JSON Template" output port. ....	6
Figure 5: Taverna Workbench "Service Panel" component with "JSONPath" item. ....	7
Figure 6: Taverna Workbench "Insert" menu with "JSONPath" item.....	7
Figure 7: Configuration dialog for "JSONPath" service.....	8
Figure 8: Selection dialog for JSONPath expressions. ....	9

## Abstract

This document describes a plug-in for Taverna Workbench that facilitates the generation and use of well-formed and valid JSON data structures in T2 workflows.

The plug-in provides two service templates:

- **JSON Template** – A service that uses a template to generate a JSON data structure.
- **JSONPath** – A service that evaluates JSONPath (XPath for JSON) expressions.

## 1 JSON Template

The purpose of this plug-in is to enable the generation of JSON data structures in T2 workflows.

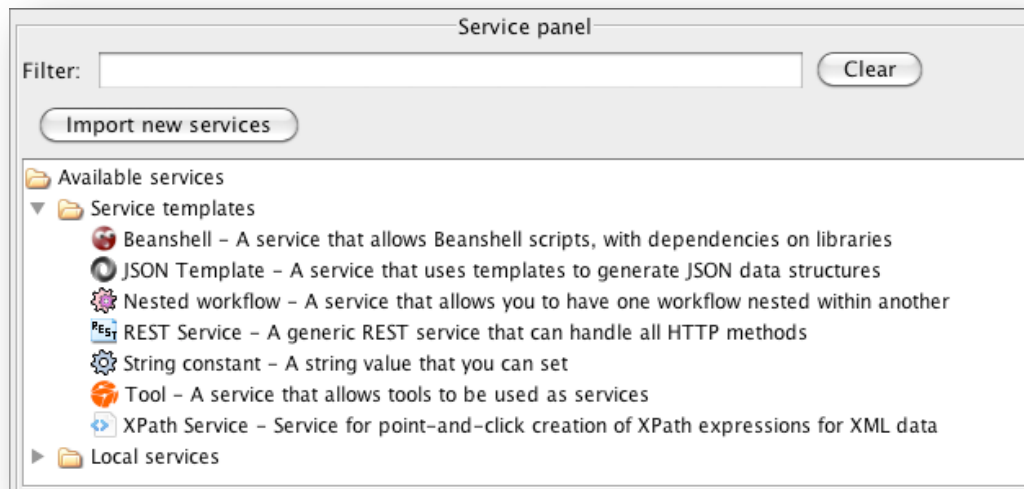


Figure 1: Taverna Workbench "Service panel" component with "JSON Template" item.

One new item is added to the "Service panel" component (see Figure 1):

- **JSON Template** – A service that uses a template to generate a JSON data structure.

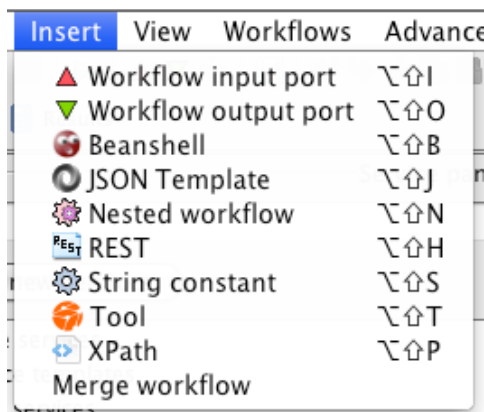


Figure 2: Taverna Workbench "Insert" menu with "JSON Template" item.

The JSON Template service can also be added to the current workflow using the "Insert" menu (see Figure 2).

## 1.1 Adding and Configuring a “JSON Template” Service

1. Expand **Service templates** under **Available services** in the **Service panel**.
2. Select **JSON Template**.
3. Drag **JSON Template** and drop it into the **Workflow diagram**.
4. In the dialog that appears, configure the service as explained next.

JSON Template services may have any number of input and output ports.

The following conditions trigger a non-fatal warning:

- The service has no input ports.<sup>1</sup>
- The service has no output ports.<sup>2</sup>
- The service defines an input port that is never referenced.

The following conditions trigger a fatal exception:

- The service has no input or output ports.

---

<sup>1</sup> Services with no input ports can be used to generate valid and well-formed JSON data structures (the values of the output ports).

<sup>2</sup> Services with no output ports can be used to validate JSON data structures (the values of the input ports).

### 1.1.1 Adding and Configuring an Input Port

1. Select the **Input ports** tab.
2. Click the **Add input port** button.
3. In the new tab that appears (see Figure 3), configure the input port as explained next.

Input ports are renamed by double-clicking the corresponding tab title.

Input ports are removed by closing the corresponding tab.

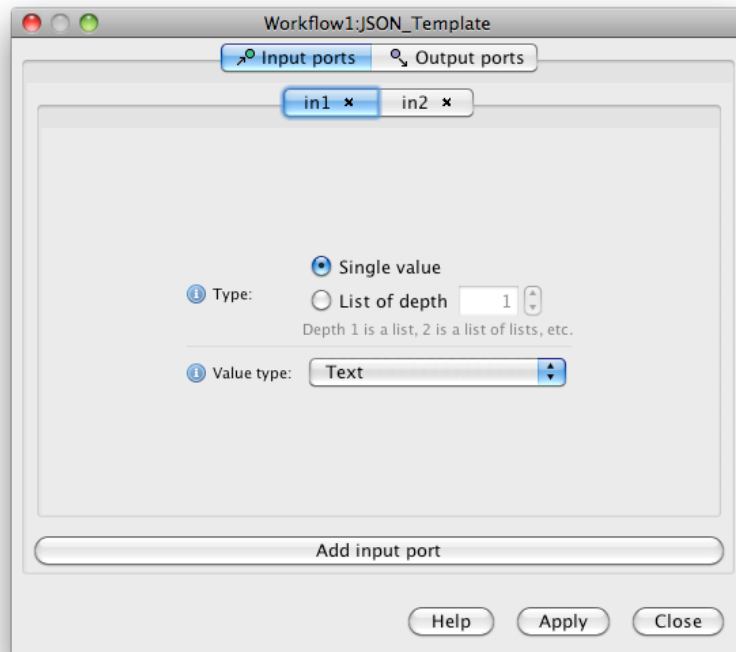


Figure 3: Configuration dialog for "JSON Template" input port.

Input ports have two configurable parameters:

- **Type** – Specifies the granularity of the input port: single value (0), list of values (1), list of lists of values (2), etc.
- **Value type** – Specifies the data type for values of the input port:
  - **JSON** – Parses the value as a JSON data structure. If successful, then inserts the result, otherwise, triggers a fatal exception.
  - **Text** – Inserts the value as a quoted and escaped JSON string.

### 1.1.2 Adding and Configuring an Output Port

1. Select the **Output ports** tab.
2. Click the **Add output port** button.
3. In the new tab that appears (see Figure 4), configure the output port as explained next.

Output ports are renamed by double-clicking the corresponding tab title.

Output ports are removed by closing the corresponding tab.

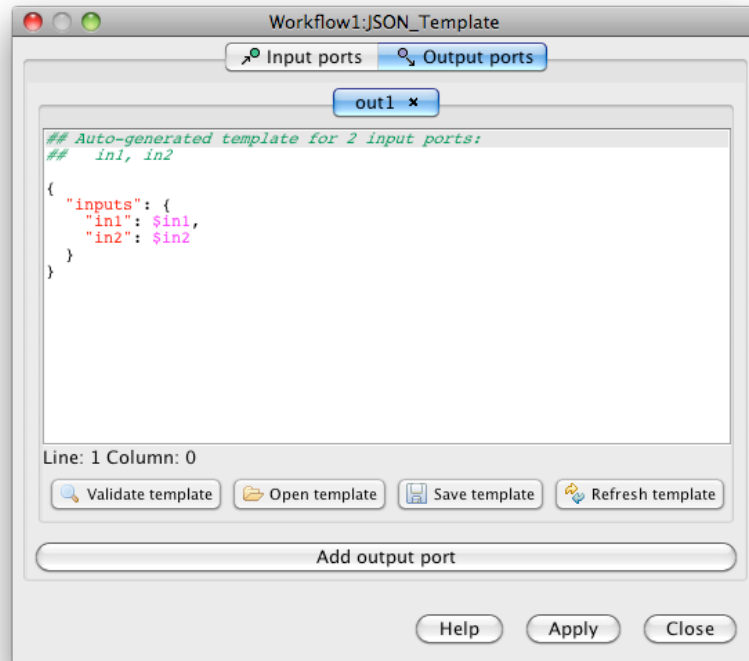


Figure 4: Configuration dialog for "JSON Template" output port.

Output ports have one configurable parameter:

- **Template** – Specifies the structure of the value of the output port.

Templates use JSON syntax with embedded input port references:

*$\$inputPortName$*   
or  
 *$\${inputPortName}$*

Comments start with “##” and continue to the end of the line.

Whitespace that is not part of a double-quoted string is not significant.

The following conditions trigger a fatal exception:

- The template references an undefined input port.
- The result of rendering the template is not a valid and well-formed JSON data structure.

## 2 JSONPath

The purpose of this plug-in is to enable the use of JSONPath<sup>3</sup> expressions in T2 workflows.

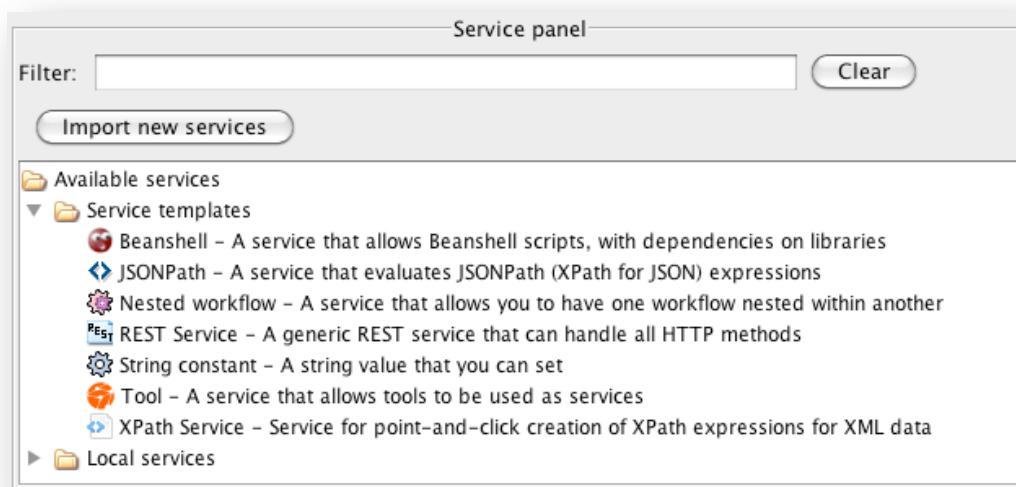


Figure 5: Taverna Workbench "Service Panel" component with "JSONPath" item.

One new item is added to the "Service Panel" component (see Figure 5):

- **JSONPath** – A service that evaluates JSONPath (XPath for JSON) expressions.

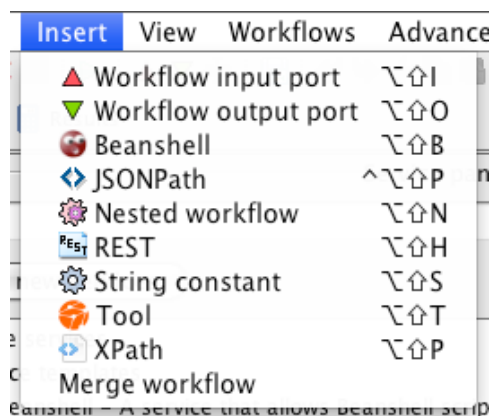


Figure 6: Taverna Workbench "Insert" menu with "JSONPath" item.

The JSONPath service can also be added to the current workflow using the "Insert" menu (see Figure 6).

<sup>3</sup> JSONPath – <http://goessner.net/articles/JsonPath/>

## 2.1 Adding and Configuring a “JSONPath” Service

1. Expand **Service templates** under **Available services** in the **Service panel**.
2. Select **JSONPath**.
3. Drag **JSONPath** and drop it into the **Workflow diagram**.
4. In the dialog that appears (see Figure 7), configure the service as explained next.

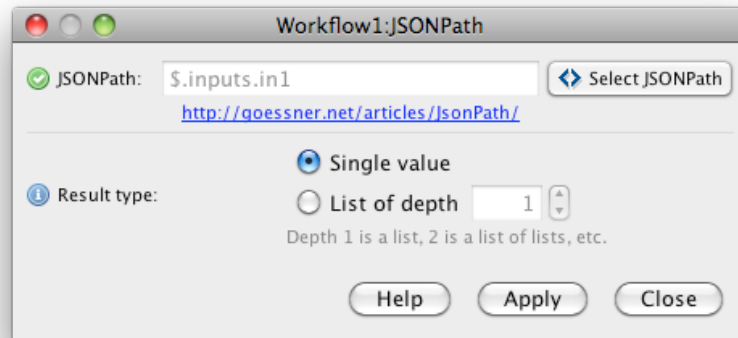


Figure 7: Configuration dialog for "JSONPath" service.

JSONPath services have two configurable parameters:

- **JSONPath** – Specifies the JSONPath expression.
- **Result type** – Specifies the granularity of the result of the evaluation of the specified JSONPath expression: single value (0), list of values (1), list of lists of values (2), etc.

JSONPath services have one input port:

- **in** – A JSON data structure.

JSONPath services have either one or two output ports:

- **out** – The result; a JSON data structure.
- **outAsList** – The value of the “out” output port as a list; available if the result of the evaluation is a list.

The following conditions raise a fatal exception:

- The JSONPath expression is empty or invalid.



### 2.1.1 Selecting a JSONPath Expression

1. In the configuration dialog for the JSONPath service, click the **Select JSONPath** button.
2. In the selection dialog that appears (see Figure 8):
  - a. Specify a JSON data structure.
  - b. Click the **Import** button to generate a tree using the specified JSON data structure.
  - c. Either:
    - i. Select the tree node that corresponds to the required JSON value (the JSONPath expression will be updated automatically); or
    - ii. Specify a JSONPath expression (matching tree nodes will be selected automatically).
3. Click the **OK** button to close the selection dialog.

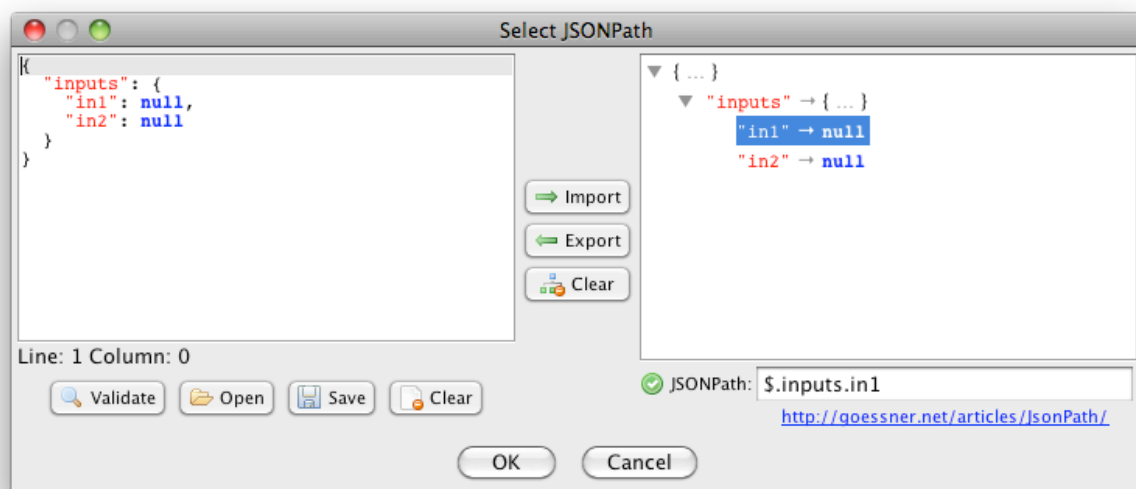


Figure 8: Selection dialog for JSONPath expressions.