JSON Tool

Definition
The JSON Tool enables one to issue a JSON web service call, and to save the returned data into Joget's form data and/or into the process's workflow variable.



Figure 1: JSON Tool Properties - Configure JSON Tool

Name	Description	
JSON URL	URL to be called.	
Call Type	● GET ● POST	
POST Method	POST Parameters POST Parameters as JSON Payload Custom JSON Payload	
POST Parameters	When POST Method is set to "POST Parameters", these parameters will be sent as a UrlEncodedFormEntity.	
	When POST Method is set to "POST Parameters as JSON Payload", these parameters will be sent as a StringEntity in a form of an escaped JSON string.	
Custom JSON Payload	Write your own JSON to be the payload. It will be sent as a StringEntity.	
	This option is available only when "Custom JSON Payload" in selected.	
Request Headers	Add name(s) and value(s) to the request header.	
No Response Expected	Expected Check if no response is expected, so that even if there is a response, this tool will simply ignore it.	
	Using this option will also disable "store to form" and "store to workflow variable" properties.	
Debug Mode	Show relevant debug entries in the server log for debugging purposes.	

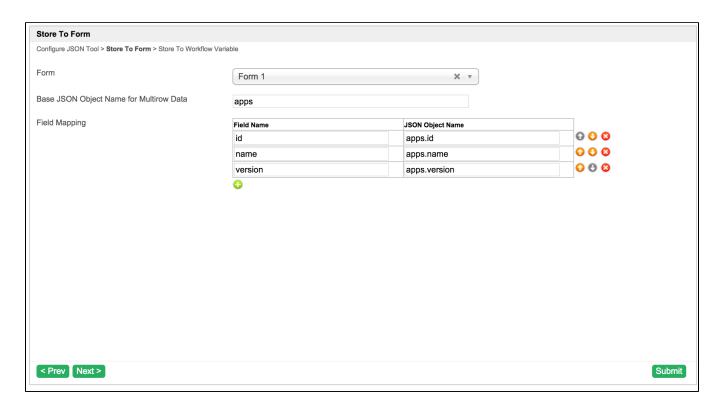


Figure 2: JSON Tool Properties - Store to Form

Name	Description		
Form	Target form to store data to.		
Base JSON Object Name for Multirow Data	Name of the object that contains an array to be based on.		
Field Mapping	Mapping with JSON data with Form fields.		
	Name	Description	
	Field Name	Form field ID	
	JSON Object Name	JSON property name	

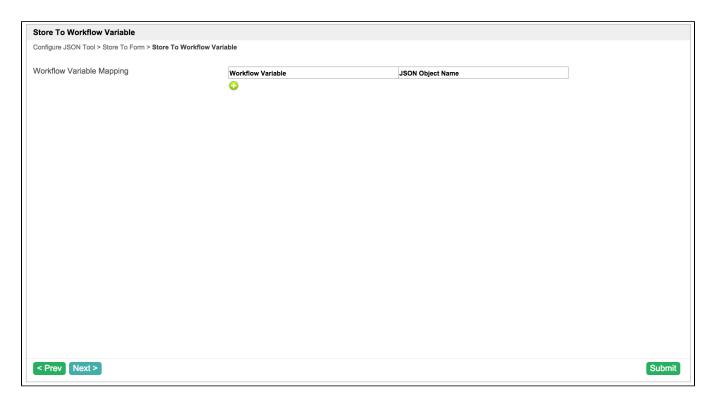


Figure 3: JSON Tool Properties - Store to Workflow Variable

Name	Description		
Workflow Variable Mapping	Name	Description	
	Workflow Variable	Workflow Variable Name.	
	JSON Object Name	JSON property name.	

In figure 2 and 3, one will need to specify on how to treat the returned data. The returned data may be saved as form data or/add to be saved into process's workflow variable. The example used in this article shows on how one can store multi-row data into a form data table.

Sample JSON API: http://localhost:8080/jw/web/json/apps/published/userviews

Sample JSON Result:

```
"apps": [{
        "id": "crm",
        "name": "CRM",
        "userviews": [{
           "id": "crm_userview_sales",
            "description": "",
            "name": "CRM: Sales Force Automation",
            "url": "/jw/web/userview/crm/crm_userview_sales",
            "version": 1
        }],
        "version": 1
   }, {
        "id": "hr_expense",
        "name": "HR Expenses Claim App",
        "userviews": [{
           "id": "hr_expense_userview",
            "description": "",
            "name": "HR Expenses Claim App for Admin",
            "url": "/jw/web/userview/hr_expense/hr_expense_userview",
            "version": 2
        }],
        "version": 2
   }, {
    "id": "isr",
    "' "Int
        "name": "Internal Service Request",
        "userviews": [{
           "id": "isr",
            "description": "",
            "name": "Internal Service Request App",
            "url": "/jw/web/userview/isr/isr",
            "version": 1
        }],
        "version": 1
    }, {
        "id": "ims",
        "name": "Simple Invoicing",
        "userviews": [{
            "id": "ims",
            "name": "Simple Invoicing",
            "url": "/jw/web/userview/ims/ims",
            "version": 1
        }],
        "version": 1
   }]
}
```

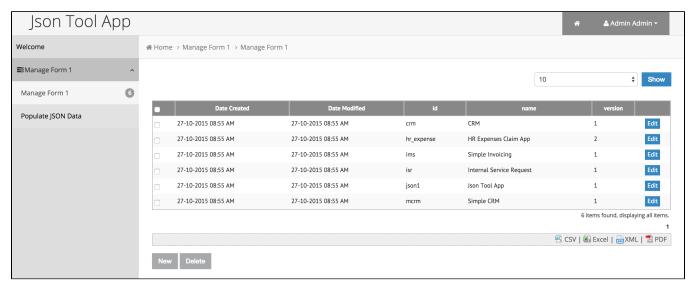


Figure 4: Sample JSON data saved in Joget form data table

You can download the test app from APP_json1-1-20151026181405.jwa.

Related Tutorials

Configure JSON Tool based on the JSON data structure