

# Javascript API

- AssignmentManager
  - completeAssignment(baseUrl, activityId, redirect)
  - completeAssignmentWithVariable(baseUrl, activityId, variableData, redirect)
  - getCurrentUsername(baseUrl, callback)
  - login(baseUrl, username, password, callback)
  - loginWithHash(baseUrl, username, hash, callback)
  - logout(baseUrl)
  - withdrawAssignment(baseUrl, activityId)
- ConnectionManager
  - ajaxJsonp(url, callback, params)
  - get(url, callback, params, xss)
  - post(url, callback, params)
- FormUtil
  - getField(fieldId)
  - getFieldsAsQueryString(fields)
  - getGridCells(cellFieldId)
  - getGridCellValues(cellFieldId)
  - getValue(fieldId)
  - getValues(fieldId)
- JPopup
  - create(id, title, width, height)
  - hide(id)
  - show(id, url, params, title, width, height, action)
- UI
  - adjustPopUpDialog(dialogbox)
  - escapeHTML(content)
  - getPopUpHeight(height)
  - getPopUpWidth(width)
  - userviewThemeParams()
- UriUtil
  - constructQueryString(params)
  - encodeUriParam(url)
  - getUriParams(url)
  - mergeRequestQueryString(queryString1, queryString2)

## AssignmentManager

- URL : /jw/js/json/util.js
- Auto included in all userview pages.
- Used to deal with assignment of a logged in user.

### **completeAssignment(baseUrl, activityId, redirect)**

#### **Description**

Completes an assignment with a specific process instance id & activity instance id

#### **Parameters**

- baseUrl - base URL of Joget Workflow, e.g., 'http://localhost/jw'
- activityId - activity instance id of the assignment to be completed
- redirect - a URL to redirect to after the assignment is completed (optional)

#### **Sample code**

```
AssignmentManager.completeAssignment('http://localhost/jw', '1_1_activity',  
'http://localhost/completed.jsp');
```

### **completeAssignmentWithVariable(baseUrl, activityId, variableData, redirect)**

#### **Description**

Completes an assignment with a specific process instance id & activity instance id with option to set workflow variables

#### Parameters

- baseUrl - base URL of Joget Workflow, e.g., 'http://localhost/jw'
- activityId - activity instance id of the assignment to be completed
- variableData - variables to be set. All variable name must prefix with "var\_"
- redirect - a URL to redirect to after the assignment is completed (optional)

#### Sample code

```
AssignmentManager.completeAssignmentWithVariable('http://localhost/jw',  
'1_1_activity', 'var_status=new&var_id=123', 'http://localhost/completed.jsp');
```

## getCurrentUsername(baseUrl, callback)

#### Description

Gets the current logged in username in Joget Workflow

#### Parameters

- baseUrl - base URL of Joget Workflow, e.g., 'http://localhost/jw'
- callback - a callback function after a successful call

#### Sample code

```
var callback = {  
  success : function(response){  
    //response.username  
    if(response.username != "roleAnonymous"){  
      console.log("Username is " + response.username);  
    }else{  
      console.log("User is anonymous");  
    }  
  }  
};  
AssignmentManager.getCurrentUsername('http://localhost/jw', callback);
```

## login(baseUrl, username, password, callback)

#### Description

Login the user in Joget Workflow

#### Parameters

- baseUrl - base URL of Joget Workflow, e.g., 'http://localhost/jw'
- username - username for user to login
- password - password for user to login
- callback - a callback function after a successful call (optional)

#### Sample code

```

var callback = {
  success : function(response){
    //response.username && response.isAdmin
    if(response.username != "roleAnonymous"){
      console.log("username (" + response.username + ") is " +
((response.isAdmin !== undefined && response.isAdmin === "true")?"admin":"not an
admin"));
    }else{
      console.log("Fail to login user!");
    }
  }
};
AssignmentManager.login('http://localhost/jw', 'admin', 'admin', callback);

```

## loginWithHash(baseUrl, username, hash, callback)

### Description

Login the user with hash in Joget Workflow

### Parameters

- baseUrl - base URL of Joget Workflow, e.g., 'http://localhost/jw'
- username - username for user to login
- hash - hashed password for user to login. (refer to [Hashed Password](#))
- callback - a callback function after a successful call (optional)

### Sample code

```

var callback = {
  success : function(response){
    //response.username && response.isAdmin
    if(response.username != "roleAnonymous"){
      console.log("username (" + response.username + ") is " +
((response.isAdmin !== undefined && response.isAdmin === "true")?"admin":"not an
admin"));
    }else{
      console.log("Fail to login user!");
    }
  }
};
AssignmentManager.loginWithHash('http://localhost/jw', 'admin',
'14ACD782DCFE2BCDE2B271CCD559477', callback);

```

## logout(baseUrl)

### Description

Logs out the current logged in username in Joget Workflow

### Parameters

- baseUrl - base URL of Joget Workflow, e.g., 'http://localhost/jw'

### Sample code

```
AssignmentManager.logout('http://localhost/jw');
```

## withdrawAssignment(baseUrl, activityId)

### Description

Withdraws an assignment with a specific activity instance id

**Deprecated** since v3, the concept of accept & withdraw assignment is removed.

### Parameters

- baseUrl - base URL of Joget Workflow, e.g., 'http://localhost/jw'
- activityId - activity instance id of the assignment to be withdrawn

### Sample code

```
AssignmentManager.withdrawAssignment('http://localhost/jw', '1_1_activity');
```

## ConnectionManager

- URL : /jw/js/json/util.js
- Auto included in all userview pages.
- Convenient method to do AJAX call.

## ajaxJsonp(url, callback, params)

### Description

Initiates Ajax call with JSONP

### Parameters

- url - URL to initiate Ajax call
- callback - a callback function after a successful call (optional)
- params - parameters for the call (optional)

### Sample code

```
var callback = {
  success : function(response){
    //do something
  }
};
ConnectionManager.ajaxJsonp('http://localhost/test', callback, 'id=4&name=test');
```

## get(url, callback, params, xss)

### Description

Initiates Ajax GET to a specific URL

### Parameters

- url - URL for GET action
- callback - a callback function after a successful call (optional)

- params - parameters for the call (optional)
- xss - use when the URL is cross-domain (optional)

#### Sample code

```
var callback = {
  success : function(response){
    //do something
  }
};
ConnectionManager.get('http://localhost/test', callback, 'id=6&name=test', false);
```

## post(url, callback, params)

#### Description

Initiates Ajax POST to a specific URL

#### Parameters

- url - URL for GET action
- callback - a callback function after a successful call (optional)
- params - parameters for the call (optional)

#### Sample code

```
var callback = {
  success : function(response){
    //do something
  }
};
ConnectionManager.post('http://localhost/test', callback, 'id=6&name=test');
```

## FormUtil

- URL : /jw/js/json/formUtil.js
- Auto included in page using form.
- Convenient method to interact with form field.

## getField(fieldId)

#### Description

Used to gets the field object of a form field

#### Parameters

- fieldId - id of a form field

#### Sample code

```
var field = FormUtil.getField("field1");
$(field).val("test"); //set value
```

## getFieldsAsUrlQueryString(fields)

#### Description

Used to generates the fields value as url query parameter string

#### Parameters

- fields - an array contains objects with "field", "param" and "defaultValue" attributes.
  - field : id of a form field
  - param : parameter name to be used
  - defaultValue : value to be used when the field return empty value (Optional)

#### Sample code

```
var fields = [
  {"field": "field1", "param": "p_field1"},
  {"field": "field2", "param": "p_field2"},
  {"field": "field3", "param": "p_field3", "defaultValue": "default value"},
];

var queryString = FormUtil.getFieldsAsUrlQueryString(fields);
console.log(queryString);
//p_field1=Field1%20value&p_field2=Field2%20value;Field2%20second%20value&p_field3=def
ault%20value
```

### getGridCells(cellFieldId)

#### Description

Used to gets the cell objects of every rows of a grid field

#### Parameters

- cellFieldId - Grid Field Id and the cell id separated by a dot ".". Eg. gridId.cellId

#### Sample code

```
var cells = FormUtil.getGridCells("gridId.field1");
$(cells).each(function(){
  //do something
});
```

### getGridCellValues(cellFieldId)

#### Description

Used to gets the cell values of every rows of a grid field and return it in an array.

#### Parameters

- cellFieldId - Grid Field Id and the cell id separated by a dot ".". Eg. gridId.cellId

#### Sample code

```
var values = FormUtil.getGridCellValues("gridId.field1");
for (var i = 0; i < values.length; i++) {
  console.log(values[i]); //i equals to row number start from 0
}
```

### getValue(fieldId)

#### Description

Used to gets the value of a form field.

#### Parameters

- fieldId - id of a form field

#### Sample code

```
var value = FormUtil.getValue("field1");
```

## getValues(fieldId)

#### Description

Used to gets the values of a form field. Values will return in an array.

#### Parameters

- fieldId - id of a form field

#### Sample code

```
var values = FormUtil.getValues("field1");
for (var i = 0; i < values.length; i++) {
    console.log(values[i]);
}
```

## JPopup

- URL : /jw/js/json/ui\_ext.js
- Auto included in all userview pages.
- Convenient method to create/show/hide a popup dialog to display a page.

## create(id, title, width, height)

#### Description

Used to creates a popup dialog element.

#### Parameters

- id - an unique identifier of the popup dialog
- title - a title to display for on the top of popup dialog (Optional)
- width - width of the popup dialog box (Optional)
- height - height of the popup dialog box (Optional)

#### Sample code

```
JPopup.create("testPopup", "Test Popup Dialog");
```

## hide(id)

Used to hides a created and shown popup dialog element.

#### Parameters

- id - an unique identifier of the popup dialog

#### Sample code

```
JPopup.hide( "testPopup" );
```

## show(id, url, params, title, width, height, action)

### Description

Used to creates a popup dialog element.

### Parameters

- id - an unique identifier of the popup dialog
- url - an URL of a page to show in popup dialog
- params - a JSON object to pass parameter and its value (Optional)
- title - a title to display for on the top of popup dialog (Optional)
- width - width of the popup dialog box (Optional)
- height - height of the popup dialog box (Optional)
- action - Get/Post. Default to Post (Optional)

### Sample code

```
var params = {  
    id : "1",  
    name : "test"  
};  
  
JPopup.show( "testPopup", "http://www.joget.org", params );
```

## UI

- URL : /jw/js/json/ui.js
- Auto included in all userview pages.
- Convenient method to retrieve value for UI usages.

## adjustPopUpDialog(dialogbox)

### Description

Used by system to align the popup dialog to the center of the screen.

### Parameters

- dialogbox - the popup dialog object

### Sample code

```
UI.adjustPopUpDialog(JPopup.dialogboxes[ "testPopup" ] );
```

## escapeHTML(content)

### Description

Used to escapes HTML syntax in a value

### Parameters

- content - content to be escapes



### Sample code

```
var content = "<p>test content</p>";
var escapedContent = UI.escapeHTML(content);
console.log(escapedContent); // &lt;p&gt;test content&lt;/p&gt;
```

## getPopUpHeight(height)

### Description

Used by the system to calculate the height of a popup dialog for current screen size to support mobile device.

### Parameters

- height - Max height of the popup dialog. Default to "90%".

### Sample code

```
var height = UI.getPopUpHeight("500px");
```

## getPopUpWidth(width)

### Description

Used by the system to calculate the width of a popup dialog for current screen size to support mobile device.

### Parameters

- width - Max width of the popup dialog. Default to "90%".

### Sample code

```
var width = UI.getPopUpWidth("800px");
```

## userviewThemeParams()

### Description

Used by system to gets the query parameters string that contains the meta of current userview theme in used.

### Sample code

```
var url = "http://localhost/jw/form/embed?" + UI.userviewThemeParams();
```

## UrlUtil

- URL : /jw/js/json/util.js
- Auto included in all userview pages.
- Convenient method to deal with URL.

## constructUrlQueryString(params)

### Description

Used to generate a query string based on a parameters object

## Parameters

- params - an object contains all parameter as attribute name and its values in array

## Sample code

```
var params = {
  "name" : ["joget"],
  "email" : ["info@joget.org", "test@joget.org"]
};
var queryString = UrlUtil.constructUrlQueryString(params);
console.log(queryString); //
name=joget&email=info%40joget%2Eorg&email=test%40joget%2Eorg
```

## encodeUrlParam(url)

### Description

Used to encodes the URL parameters in a URL.

### Parameters

- url - URL with parameters to be encode. Note: it use "&" and "=" as separator.

### Sample code

```
var url =
"http://localhost/jw/test?name=joget&email=info@jogte.org&email=test@joget.org";

var encodedUrl = UrlUtil.encodeUrlParam(url);
console.log(encodedUrl); //
http://localhost/jw/test?name=joget&email=info%40joget%2Eorg&email=test%40joget%2Eorg
```

## getUrlParams(url)

### Description

Used to gets an object contains all parameter as attribute name and its value in an URL.

### Parameters

- url - URL to be parses to retrieve all parameters and its value in array.

### Sample code

```
var url =
"http://localhost/jw/test?name=joget&email=info@jogte.org&email=test@joget.org";

var params = UrlUtil.getUrlParams(url);
console.log(params); // {"name" : ["joget"], "email" : ["info@joget.org",
"test@joget.org"]}
```

## mergeRequestQueryString(queryString1, queryString2)

### Description

Used to merge 2 URL query parameters strings into one query string.

### Parameters

- queryString1 - first query parameters string
- queryString2 - second query parameters string. If a parameter is exist in both query strings, the value in second query string will override the first one.

#### Sample code

```
var q1 = "name=joget&email=info@jogte.org&email=test@joget.org";
var q2 = "name=joget team&phone=012345678";

var queryString = UrlUtil.mergeRequestQueryString(q1, q2);
console.log(queryString); //
name=joget%20team&email=info%40joget%2Eorg&email=test%40joget%2Eorg&phone=012345678
```