

JSON API Authentication

- [Authenticate by Passing Parameters](#)
- [Master Login Username and Password](#)
- [Basic Http Authentication](#)

Authenticate by Passing Parameters

For JSON API authentication purposes, the following parameters can be posted/appended to each of the JSON API URLs:

- j_username
- j_password
- hash

Example:

Assuming the username and password required is "user1" and "password1" respectively, we can post the username and password to the JSON API using following jQuery script.

```
<script>
$(document).ready(function(){
    $.ajax({
        type: "POST",
        url: 'http://localhost:8080/jw/web/json/workflow/assignment/list/pending',
        data: {
            j_username : 'user1',
            j_password : 'password1'
        },
        success: function(res) {
            console.log(res)
        },
        dataType: "json"
    });
});
</script>
```

If you prefer to use [hashed password](#), you can use the following script.



Please note that the support on Hashed Password is based on the Directory Manager you are using. Some Directory Manager Plugin may not supporting this type of authentication method.

The format and hashing method may vary for each Directory Manager as well.

```
<script>
$(document).ready(function(){
    $.ajax({
        type: "POST",
        url: 'http://localhost:8080/jw/web/json/workflow/assignment/list/pending',
        data: {
            j_username : 'user1',
            hash : 'D012B772672A55A0B561EAA53CA7734E'
        },
        success: function(res) {
            console.log(res)
        },
        dataType: "json"
    });
});
</script>
```

Master Login Username and Password

When authentication using parameters, you are allowed to using a Master Credential to login as other user to performance workflow activities.

To use it, set a Master Login Username and Master Login Password under **System Settings > General Setting**. By setting these values, a different user can be specified by passing in the "loginAs" parameter.



Please note that only enable this when it is necessary. Leaking of your Master Credential will allow others to perform all the unwanted JSON API calls.

System Administration Settings	
Master Login Username	<input type="text" value="master"/>
Master Login Password	<input type="password" value="*****"/> <i>master login hash:E505CF727D214A68CB03DA25DA978500</i>

Assuming the master login username and master login password is "master" and "master" respectively, the master login hash will be "E505CF727D214A68CB03DA25DA978500".

The following example showcases how to use a Master Credential to login as "user1".

```
<script>
$(document).ready(function(){
$.ajax({
type: "POST",
url: 'http://localhost:8080/jw/web/json/workflow/assignment/list/pending',
data: {
j_username : 'master',
j_password : 'master',
loginAs : 'user1'
},
success: function(res) {
console.log(res)
},
dataType: "json"
});
});
</script>
```

Using master login hash:

```
<script>
$(document).ready(function(){
$.ajax({
type: "POST",
url: 'http://localhost:8080/jw/web/json/workflow/assignment/list/pending',
data: {
j_username : 'master',
hash : 'E505CF727D214A68CB03DA25DA978500',
loginAs : 'user1'
},
success: function(res) {
console.log(res)
},
dataType: "json"
});
});
</script>
```

Basic Http Authentication

Since V4, Joget supports Basic HTTP Authentication in JSON API authentication, you can pass the credentials in the header.

Example:

Assuming the username and password required is "user1" and "password1" respectively, we can set the Basic Auth header to the JSON API using the following jQuery script.

```
<script>
  $(document).ready(function(){
    $.ajax({
      type: "POST",
      url: 'http://localhost:8080/jw/web/json/workflow/assignment/list/pending',
      beforeSend: function (xhr) {
        xhr.setRequestHeader ("Authorization", "Basic dXNlcjE6cGFzc3dvcnQx");
      },
      success: function(res) {
        console.log(res)
      },
      dataType: "json"
    });
  });
</script>
```