

(SSO)

- [JSON API](#)
- [HttpJSON API](#)
- [JavaScript API](#)
-

Joget Workflow

JSON API

- '/web/json/directory/user/sso' [JSON API](#)
- [JSON API](#) or
- """"

```
<script>
  $(document).ready(function(){
    $.ajax({
      type: "POST",
      url: 'http://localhost:8080/jw/web/json/directory/user/sso?callback=callbackFunction',
      data: {
        username: 'admin',
        password: 'admin'
      },
      success: function(res) {
        console.log("username (" + res.username + ") is " + ((res.isAdmin !== undefined && res.isAdmin
=== "true")?"admin":"not an admin"));
      },
      dataType: "json"
    });
  });
</script>
```

HttpJSON API

- V4 Joget WorkflowHTTPJSON API
- "user1""password1"jQueryBasic AuthJSON API

```
<script>
  $(document).ready(function(){
    $.ajax({
      type: "POST",
      url: 'http://localhost:8080/jw/web/json/directory/user/sso',
      beforeSend: function (xhr) {
        xhr.setRequestHeader ("Authorization", "Basic dXNlcjE6cGFzc3dvcmQx");
      },
      success: function(res) {
        console.log("username (" + res.username + ") is " + ((res.isAdmin !== undefined && res.isAdmin
=== "true")?"admin":"not an admin"));
      },
      dataType: "json"
    });
  });
</script>
```

JavaScript API

- jQueryutil.js
- AssignmentManager.loginSSO
-

```
<script type="text/javascript" src="http://localhost:8080/jw/js/jquery/jquery-1.9.1.min.js"></script>
<script type="text/javascript" src="http://localhost:8080/jw/js/json/util.js" ></script>

<script type="text/javascript" >
$(document).ready(function(){
    var loginCallback = {
        success : function(response){
            if(response.username != "roleAnonymous"){
                alert("login successfully");
            }else{
                alert("login fail");
            }
        }
    };
    AssignmentManager.login('http://localhost:8080/jw', 'admin', 'admin', loginCallback);
});
</script>
```

- [Web ServiceSSO](#)

```

import org.joget.apps.workflow.security.WorkflowUserDetails;
import org.joget.directory.model.service.DirectoryManager;
import org.joget.workflow.model.service.WorkflowUserManager;
import org.joget.apps.app.service.AppUtil;
import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.security.core.Authentication;
import org.joget.directory.model.User;
import org.joget.workflow.util.WorkflowUtil;
import org.springframework.security.core.context.SecurityContextHolder;
import javax.servlet.http.HttpSession;
import javax.servlet.http.HttpServletRequest;
import org.springframework.security.web.savedrequest.HttpSessionRequestCache;
import org.springframework.security.web.savedrequest.SavedRequest;

//Get service beans
DirectoryManager dm = (DirectoryManager) AppUtil.getApplicationContext().getBean("directoryManager");
WorkflowUserManager workflowUserManager = (WorkflowUserManager) AppUtil.getApplicationContext().getBean(
("workflowUserManager"));

//Login as "clark"
String username = "clark";
User user = dm.getUserByUsername(username);

if (user != null) {
    WorkflowUserDetails userDetail = new WorkflowUserDetails(user);

    //Generate an authentication token without a password
    Authentication auth = new UsernamePasswordAuthenticationToken(userDetail, userDetail.getUsername(),
userDetail.getAuthorities());

    //Login the user
    SecurityContextHolder.getContext().setAuthentication(auth);
    workflowUserManager.setCurrentThreadUser(user.getUsername());

    // generate new session to avoid session fixation vulnerability
    HttpServletRequest httpRequest = WorkflowUtil.getHttpServletRequest();
    HttpSession session = httpRequest.getSession(false);
    if (session != null) {
        SavedRequest savedRequest = (SavedRequest) session.getAttribute("SPRING_SECURITY_SAVED_REQUEST_KEY");
        session.invalidate();
        session = httpRequest.getSession(true);
        if (savedRequest != null) {
            session.setAttribute("SPRING_SECURITY_SAVED_REQUEST_KEY", savedRequest);
        }
    }
}
}

```