

SSO

```
JARjar ${}/tomcat8/lib
```

SSO

```
var version = java.lang.System.getProperty("java.version");
if (version.startsWith("1.8.0")) {
    load("nashorn:mozilla_compat.js");
}
//BI
var macoHost = "http://bi.xxx.com";

//BI
var redirectUrl = request.getParameter("redirectUrl");
if(redirectUrl != null && redirectUrl.length() > 0) {
    redirectUrl = java.net.URLEncoder.encode(redirectUrl, "utf-8")
}
//
var urlStr = macoHost+"/plugin/sso/auth.do?userRedirectUrl="+redirectUrl;
//header, header
var userAgent = request.getHeader("User-Agent");
logger.debug("User Agent: "+userAgent);
if(userAgent != null && userAgent.indexOf("xapp")>-1){
    urlStr =
macoHost+"/plugin/sso/appmobile/auth.do?app=xapp&redirectUrl="+redirectUrl;
}
logger.debug("url str: "+urlStr);
result.setSuccess(true);
//
response.sendRedirect(urlStr);
```

: /plugin/sso/auth

```
var version = java.lang.System.getProperty("java.version");
if (version.startsWith("1.8.0")) {
    load("nashorn:mozilla_compat.js");
}
importPackage(java.lang);
importPackage(java.util);
importPackage(org.apache.commons.lang3.time);
```

```
importPackage(org.apache.commons.lang3);
importPackage(com.alibaba.fastjson);
importPackage(org.apache.commons.codec.binary);
importPackage(org.apache.http.impl.client);
importPackage(org.apache.http.client.methods);
importPackage(org.apache.http);
importPackage(com.weibaobiao.command);

//BI
var macoHost = "http://bi.xxx.com";

//
var baseUrl = "https://idms.app.com/ms_oauth";
//
var client_id="xxx";
var client_secret = "yyy";

//
var callbackUrl = macoHost+"/plugin/sso/auth.do";
callbackUrl = java.net.URLEncoder.encode(callbackUrl, "utf-8");

//BI
var redirectUrl = macoHost+"/workbench/home.do";

//
var viewreport = request.getParameter("viewreport");
logger.warn("viewreport: "+viewreport);
if(viewreport != null && viewreport === "1") {
    var theUrl = request.getParameter("redirectUrl");
    logger.warn("theUrl: "+theUrl);
    if(theUrl != null && theUrl !== "") {
        redirectUrl = java.net.URLEncoder.encode(theUrl);
    }
}

//code
var code = request.getParameter("code");
logger.warn("oauth code: "+code);

//
var error_message = request.getParameter("error_message");

//
var nowTime = new Date().getTime()+"";
logger.warn("nowTime: "+nowTime);

var debugObj = new JSONObject();

if(error_message != null && error_message !== ""){
    result.setSuccess(false);
    result.setData(error_message);
} else if(code == null || code === "") {
```

```

//code

//session
    var userRedirectUrl = request.getParameter("userRedirectUrl");
    logger.warn("userRedirectUrl: "+userRedirectUrl);
    if(userRedirectUrl != null && userRedirectUrl.length > 0) {
        request.getSession().setAttribute("userRedirectUrl",
userRedirectUrl);
    }

//
    var goUrl =
baseUrl+"/oauth2/authorize?client_id="+client_id+"&response_type=code&redi
rect_uri="+callbackUrl+"&scope=xxx"+"&redirectUrl="+redirectUrl;
    logger.warn("OAuth2 url: "+goUrl);
    response.sendRedirect(goUrl);
} else {

//codecode

    var userRedirectUrl =
request.getSession().getAttribute("userRedirectUrl");
    request.getSession().setAttribute("userRedirectUrl", null);
    if(userRedirectUrl != null && userRedirectUrl.length > 0 &&
userRedirectUrl != "null") {
        logger.warn("userRedirectUrl: "+userRedirectUrl);
        redirectUrl = userRedirectUrl;
    }

    logger.warn("redirectUrl: "+redirectUrl);

//

    var access_token = null;

    try{

        //access_token
        if(true){
            debugObj.put("code", code);
            var authStr = client_id+ ":"+client_secret;
            authStr = ""+new
java.lang.String(org.apache.commons.codec.binary.Base64.encodeBase64(authS
tr.getBytes("utf-8")), "utf-8");
            logger.error("Authorization: "+"Basic " + authStr);
            var goUrl =
baseUrl+"/oauth2/tokens?grant_type=authorization_code&redirect_uri="+callb
ackUrl+"&code="+code;
            logger.warn("Auth code url: "+goUrl);
            var httpClient =
org.apache.http.impl.client.HttpClients.createDefault();
            var post = new HttpPost(goUrl);
            post.addHeader("Authorization", "Basic " + authStr);

```

```

        var resp = httpclient.execute(post);
        var status = resp.getStatusLine().getStatusCode();
        logger.error("status: "+status);
        if (status === 200) {
            var entity = resp.getEntity();
            var content = org.apache.http.util.EntityUtils.toString(entity);
            logger.error("content: "+content);
            var respObj = JSON.parseObject(content);
            var access_token = respObj.getString("access_token");
            //token
            //var access_token = StringUtils.substringBetween(content,
            "\"access_token\": \"", "\"");
            logger.error("access_token: "+access_token);
        }

        result.setData("[\""+status+"] "+code);
    }

    //
    if(access_token != null) {
        debugObj.put("access_token", access_token);
        var goUrl = baseUrl+"/resources/userprofile";
        logger.warn("User profile url: "+goUrl);
        var httpclient =
org.apache.http.impl.client.HttpClients.createDefault();
        var post = new HttpGet(goUrl);
        post.addHeader("Authorization", access_token);
        var resp = httpclient.execute(post);
        var status = resp.getStatusLine().getStatusCode();
        logger.error("status: "+status);
        if (status === 200) {
            var entity = resp.getEntity();
            var content = org.apache.http.util.EntityUtils.toString(entity);
            logger.error("User profile: "+content);
            var contentObj = JSON.parseObject(content);
            var uid = contentObj.getString("uid");
            logger.error("uid: "+uid);
        }

        //
        var userDao = beanService.getBean("userDaoImpl");
        var loginUser = userDao.findByUsername(uid);

        //
        if(loginUser == null) {
            uid = uid.toLowerCase();
            loginUser = userDao.findByUsername(uid);
        }
        logger.error("loginUser: "+loginUser);

        //
        if(loginUser == null){

```

```

        var sb = new StringBuilder();

sb.append("<p>-----  

-----<p/>");
        sb.append("<p><p/>");

sb.append("<p>-----  

-----<p/>");

        response.setContentType("text/html;charset=utf-8");
        request.getSession().setAttribute("pageTitle", "DaaS
Workbench sccess denied.");
        request.getSession().setAttribute("message",
sb.toString());
        var urlStr = "/view/message.do";
        response.sendRedirect(urlStr);
    }else{

        //BI
        var userService =
beanService.getBean("userServiceImpl");
        var uc = new UserCommand();
        uc.setUsername(uid);
        uc.setPassword(loginUser.getPassword());
        userService.userLogin(request, uc, false);
        response.sendRedirect(redirectUrl);
        logger.error("Redirect to: "+redirectUrl);
    }
}

    }
}catch(e){
    logger.error(e.toString());
}
}

```

```
result.setSuccess(true);
```