TMGR CONFIGURATION WITH OAUTH AUTHENTICATION

Contents

1.0. Configuring TMGR with OAuth Authentication

1.1. Edit web.xml file

1. In **WEB-INF/web.xml** file set profile **oauthAuthentication** instead of defaultAuthentication. It should look like this:

```
<context-param>
        <param-name>spring.profiles.active</param-name>
        <param-value>oauthAuthentication,....</param-value>
</context-param>
```

1.2. Edit GS4TR_HOME/config/services.properties

Edit **services.properties** file by adding the following properties: (NOTE: Don't forget to set protocols (http/https) while setting uri properties!)

1. **oauth.authorizationEndpointUri** represents TPTAuth Server authorization uri (oauth.authorizationEndpointUri=https://ex.transperfect.com/connect/authorize).

2. **oauth.jwksUri** represents Json Web Key Set Uri (e.g. oauth.jwksUri=https://sso-qa.transperfect.com/.well-known/openid-configuration/jwks).

3. **oauth.clientId** represents clientID (your TMGR instance identifier) that you get from TPTAuth server (e.g. oauth.clientId=someClientId). Enter the same value on property **oauth.audience** (e.g. oauth.audience=someClientId).

4. **oauth.clientRedirectUri** represents client redirect Uri (Uri where TPTAuth redirect user after successful login). It is built when add "**/SSO**" to domain Uri of TMGR. Also, value of oauth.clientRedirectUri must be white-listed on TPTAuth server side.

5. **oauth.changePasswordUrl** representsTPTAuth change password uri, where TPTAuth user will be sent when choose changePassword option(e.g. oauth.changePasswordUrl=https://sso-qa.transperfect.com/Manage/ChangePassword)

6. **oauth.logoutUrl** represents TPTAuth logout url (e.g. oauth.logoutUrl=https://sso-qa.transperfect.com/connect/endsession)

7. **oauth.checkUserSessionUri** represents uri for TPTAuth check user session (e.g. oauth.checkUserSessionUri=https://sso-qa.transperfect.com/connect/checksession)

8. **oauth.checkUserSessionPeriod** represents a period of user session status check in TPTAuth in seconds (e.g. oauth.checkUserSessionPeriod=5).

9. Edit **csp.contentTemplate** property, locate **frame-src 'self'** and add TPTAuth server uri after 'self', blank separated. TPTAuth server uri can be found in some of the previous properties like oauth.logoutUrl, oauth.checkUserSessionUri etc. (csp.contentTemplate= … frame-src 'self' https://sso-qa.transperfect.com;…).

10. **oauth.support.mailAddress** represents support email address which will be set in tptAuth warning messages (e.g. tmgrproduction_tpt@transperfect.com). With default value it would look like this: *The Email address you have entered is incorrect. Please contact support team at support@translations.com*.

11. **oauth.refreshTokenInterval** represents interval for refreshing tokens (default value is 720 seconds) to avoid using expired tokens (e.g. oauth.refreshTokenInterval=720). **IMPORTANT:** *This value MUST BE LOWER than 15 minutes.*

2.0. Rest TPTAuth Configuration Instructions

2.1. Edit GS4TR_HOME/config/services.properties

1. Set property **rest.oauth.grantType** to **password** (e.g. restoauth.grantType=password).

2. Admin on tptAuth needs to create values for next properties: **rest.oauth.username**, **rest.oauth.password**, **rest.oauth.clientId** and **rest.oauth.clientSecret**. Following are examples of created values:

   rest.oauth.username= tmgr_admin@test.com
   rest.oauth.password=58p5rkH2B7z7sDjj
   rest.oauth.clientId= RTHxbKtqS5fmqFSjY3U3kVQuvvwPbw2E
   rest.oauth.clientSecret= aDp5K4t788KRNGvJsFmthtN2GqC6C5J8

3. Set property **rest.oauth.scope** to **IdPManager** (e.g. rest.oauth.scope=IdPManager).

4. Set property **rest.oauth.tokenEndpoint** to token and point url (e.g. rest.oauth.tokenEndpoint=https://sso-qa.transperfect.com/connect/token).

5. Set property **rest.oauth.apiEndPoint** to url api endpoint (e.g. rest.oauth.apiEndPoint=https://idpmanager-qa.transperfect.com/api).

2.2. Edit GS4TR_HOME/ config/tptOAuth.properties

TMGR is using this properties file for creating user on tpt oauth with configured details. This config should be populated by admin on tpt oauth.

1. Set property **tpt.addUser.locked** to true or false. Default value is false. If you set true, when TMGR create a user, created user will be locked (e.g. tpt.addUser.locked=false).

2. Set property **tpt.addUser.disabled** to true or false. Default value is false. If you set true, when TMGR create a user, created user will be disabled (e.g. tpt.addUser.disabled=false).

3. Set property **tpt.addUser.changepasswordOnNextLogin** to true or false. Default value is true. If you set true, when TMGR create a user, created user must change password on next login (e.g. tpt.addUser.changePasswordOnNextLogin=true).

4. Set property **tpt.addUser.passwordResetAllowed** to true or false. Default value is true. If you set to true, when TMGR create a user, created user can reset password (e.g. tpt.addUser.passwordResetAllowed=true.

5. Set property **tpt.addUser.passwordPolicy**. Default value is empty. Admin on tpt oauth should provide password policy.

6. Admin on tpt oauth should create clients and set up client name. Each instance of TMGR should have client name in **tpt.addUser.clients** property (e.g. tpt.addUser.client=TMGR, TMGR IdPClient).

7. Admin on tpt oauth should create organizations. There is a new property, **tpt.addUser.defaultOrganization**=Transperfect. If you want to change the default organization, set property **tpt.addUser.organizations** to match organization name you have created (e.g. tpt.addUser.organizations=Test Organization).