Login method extension

SuperMap iServer provides the OAuth2Client interface, which is used to follow the third-party login extension in OAuth2 protocol. The third party login: QQ, Sina Weibo and Renren. The QQ and Sina Weibo are the built-in login way of SuperMap iServer.

OAuth2Client interface introduction

com.supermap.services.security.OAuth2Client interface has following methods:

- getRedirectURI(String clientID, String state, String redirectUri);

This method is used to get the jumping URL of OAuth.

- getAccesstoken(String clientID, String clientSecret, String code, String redirectUri) throws IOException;

This method is used to access Token.

- getUserID(String accesstoken) throws IOException;

This method is used to get the user ID.

- getUserInfo(String token, String clientID, String userID) throws IOException;

This method is used to get the user information.

Extension and configuration flow

Here we have an example to show the extension. Use the Renren login method.

1. The implement class of Renren account login

Implement the RenRenLoginExtended class. Inherit from the OAuth2Client interface, as shown below:

```
package com.supermap.services.rest.resources.impl;
import java.io.IOException;
import java.net.URI;
import java.net.URLEncoder;
import com.supermap.services.security.OAuth2Client;
import com.supermap.services.security.OAuthUserInfo;
import org.apache.commons.io.IOUtils;
import java.net.URISyntaxException;
import org.json.JSONException;
import org.json.JSONObject;
public class RenRenLoginExtended implements OAuth2Client {
  // Renren OAuth2 authentication root address
    public static final String OAUTH_URL_RENREN = System.getProperty("OAUTH_RENREN",
"https://graph.renren.com");
    private static final String GET_CODE_URI_RENREN = OAUTH_URL_RENREN
        + "/oauth/authorize?client_id=%s&response_type=code&redirect_uri=%s&state=%s&display=page";
    private static final String GET_TOKEN_BY_CODE_RENREN = OAUTH_URL_RENREN +
"/oauth/token?client_id=%s&client_secret=%s&grant_type=authorization_code&code=%s&redirect_uri=%s";
    private static final String useridUrl = "https://api.renren.com/v2/user/get?access_token=%s";
    private static final String userInfoUrl =
"https://api.renren.com/v2/user/get?access_token=%s&client_id=%s&userID=%s";

    //Get the OAuth2 jumping URI
    public String getRedirectURI(String clientID, String state, String redirectUri) {
      return String.format(GET_CODE_URI_RENREN, clientID, redirectUri, state);
    }

    //Get the access Token
    public String getAccesstoken(String clientID, String clientSecret, String code, String redirectUri) throws
IOException {
      String tokenurl = String.format(GET_TOKEN_BY_CODE_RENREN, clientID,
clientSecret,code,redirectUri);
      try {
        String tokenResult = IOUtils.toString(new URI(tokenurl), "utf-8");
        JSONObject tokenObj = new JSONObject(tokenResult);
        return tokenObj.getString("access_token");
      } catch (JSONException e) {
        throw new IOException(e);
      }catch (URISyntaxException e) {
        throw new IOException(e);
      }
    }
    // Get the user ID
    public String getUserID(String accesstoken) throws IOException {
      String useridfourl = String.format(useridUrl, URLEncoder.encode(accesstoken, "utf-8"));
      try {
        String useridresult = IOUtils.toString(new URI(useridfourl), "utf-8");
        JSONObject useridObj = new JSONObject(useridresult);
        return useridObj.getJSONObject("response").getString("id");
      } catch (JSONException e) {
        throw new IOException(e);
      }catch (URISyntaxException e) {
        throw new IOException(e);
      }
    }
    //Get the user information
    public OAuthUserInfo getUserInfo(String token, String clientID, String userID) throws IOException {
      String userinfourl = String.format(userInfoUrl, URLEncoder.encode(token, "utf-8"), clientID, userID);
      OAuthUserInfo userinforesult = new OAuthUserInfo();
      String content = null;
      try {
        content = IOUtils.toString(new URI(userinfourl), "utf-8");
        JSONObject userinfoJson = new JSONObject(content).getJSONObject("response");
        userinforesult.figureurl = userinfoJson.getString("avatar");
        userinforesult.nickName = userinfoJson.getString("name");
        return userinforesult;
      } catch (JSONException e) {
        throw new IOException(e);
      } catch (URISyntaxException e) {
        throw new IOException(e);
      }
    }
  }
```

Copy the com folder that contains the compiled RenRenLoginExtended.class file to the SuperMap iServer Web application, that is %SuperMap iServer_HOME%\webapps\iserver\WEB-INF\classes (first create a classes folder in this directory).

2. Configuration of "Renren account"

Create a new extendedOAuth.xml file in the root directory of %SuperMap iServer_HOME%webapps/iserver/WEB-INF, containing the followings:

```
<?xml version="1.0" encoding="UTF-8"?>
<extendedOAuthSettings>
    <extendedOAuthSetting>
        <loginType>RENREN</loginType>

<oAuth2ClientClass>com.supermap.services.rest.resources.impl.RenRenLoginExtended</oAuth2ClientClass>
     </extendedOAuthSetting>
</extendedOAuthSettings>
```

The <extendedOAuthSettings> is the configuration set of third party login method extension. It can contains multiple <extendedOAuthSetting> labels. Each <extendedOAuthSetting> label corresponds to a third-party login extension configuration that follows the OAuth2 protocol. The contents in <extendedOAuthSetting> correspond to the configuration of ExtendedOAuthSetting:

- <loginType>: the extension login type. Here is the "RENREN".
- <oAuthLoginResourceClass>: the detailed implement class.

Add the following contents to the root node of iserver-system.xml, which is located in iServer_HOME%webapps/iserver/WEB-INF:

```
<server>
        ...
      <oAuthConfigs>
      <oAuthConfig>
         <id>1</id>
         <enabled>true</enabled>
         <loginType>RENREN</loginType>
         <buttonText>Renren Login</buttonText>
         <clientSecret>b7544a9cc9524bac9641346f3720384b</clientSecret>
         <clientID>fc23a7eefde348cca487c3bab60861be</clientID>
         <redirectDomain>iserver.supermap.com</redirectDomain>
         <loginIcon>renren.png</loginIcon>
      </oAuthConfig>
   </oAuthConfigs>
   <!--<oAuthMetas>
      <oAuthMeta><meta property="qc:admins" content="4323423424235" /></oAuthMeta>
   </oAuthMetas>-->
</server>
```

The <oAuthConfigs> is the configuration set of third party login method. It can contains multiple <oAuthConfig> labels. Each <oAuthConfig> label corresponds to a configuration of the login method. The contents in <oAuthConfig> correspond to the configuration of OAuthConfig ：

- <id>: the configuration ID that follows the OAuth2 protocol.
- <enabled>: whether to enable the third party login method.
- <loginType>: the third party login type that follows the OAuth2 protocol. Here it is "RENREN".
- <buttonText>: the button name displayed in the login interface. Here it is Renren Login.
- <clientSecret>: The AppSecret when applying the application. Here it is the Secret Key when apply the Renren account.
- <clientID>: The Appkey when applying the application. Here is the API KEY when apply the Renren

account.

- <redirectDomain>: The callback domain name when applying the application. Here it is the "iserver.supermap.com".
- <loginIcon>: The login picture ID displayed in the login page. Here it is the "renren.png:. This picture is located in ./staticFiles/img foder of iserver-all.jar in %SuperMap iServer_HOME%webapps/iserver/WEB-INF/lib. If <loginType> is QQ or Weibo, this label can be ignored.

<oAuthMetas> label represents the meta info (will be added to the HEAD label in the HTML codes in the first page) used to verify website address. Extending the Renren account does not need to fill with in the meta data.

3. View the extension result

When you finish above steps, representing you have added the Renren login method. In the third party login configuration (http://iserver.supermap.com:8090/iserver/manager/security/oauthconfig), you can view the configuration. These information correspond to configuration in <oAuthConfig> label. Access the login page of iPortal, iServer or iExpress, you can see that the Renren account button is added. Please refer to Usage of Third-party Loogin for more information.