R1 Cloud 基于 R1 认证鉴权的 SSO 集成和 API 调用

R1 云认证鉴权及单点登录平台采用标准的 OAuth2，因此可以采用开源的 OAuth2 客户端(请参考 OAuth2 官方)获取访问令牌后调用获取用户信息的接口(/api/user)来获取当前用户

平台也提供了辅助类封装上面说的过程来简化客户的调用，配置和使用过程如下：

1、在客户端应用的 web.xml 中注册 com.icss.as.oauth2.client.sso.SSOFilter,对需要获取到当前用户的 url 进行拦截，注意:filter-mapping 的拦截要在转码 Filter 的后面，其他 Filter 的前面，例如：

```
<filter>
      <filter-name>SSOFilter</filter-name>
      <filter-class>com.icss.as.oauth2.client.sso.SSOFilter</filter-class>
</filter>
<filter-mapping>
      <filter-name>SSOFilter</filter-name>
      <url-pattern>*.do</url-pattern>
</filter-mapping>
```

2、 编辑 WEB-INF/class/OAuth2-config.properties 文件(或通过增加 Filter 参数 configFileName 指定配置文件名称),内容如下：

```
#应用 ID

client_id=TEST-APP

#应用凭证

client_secret=8d42ee5e-d1ab-4d60-abf0-499457a11579

#应用注册的回调地址

#其中 http://localhost:8080/R1AuthricationServer 为认证中心的访问地址,请根据实际情况替换
 注意:此地址必须要被 SSOFilter 拦截

redirect_uri=http://localhost:8080/R1AuthricationServer/client.do

#授权请求的 URL

authorize_url=http://localhost:8080/R1AuthricationServer/oauth2/authorize

#获取令牌的 URL
```

```
access_token_url=http://localhost:8080/R1AuthricationServer/oauth2/access_token
#获取用户信息的 API 调用地址
api_url=http://localhost:8080/R1AuthricationServer/api
```

3、拷贝依赖的 jar 包到应用的 WEB-INF/lib 目录(实际的版本可能不同):

```
r1client.jar
bizframework.jar
commons-httpclient-3.1.jar
commons-codec.jar
slf4j-api-1.6.0.jar
slf4j-log4j12-1.6.0.jar
```

4、获取当前用户的方式:

1) 通过 request 当前用户 id: `request.getRemoteUser()`

2) 当前用户 id: `SSO.getUserId()` (需要 import com.icss.as.oauth2.client.sso.SSO)

3) 当前用户 JSONObject 对象: `SSO.getUser()` (需要 import com.icss.as.oauth2.client.sso.SSO),对象的属性参考获取用户信息

使用客户端调用 API

API Client 的作用是封装 OAuth2 协议过程,简化应用调用服务器资源的过程,有如下场景:

1、以当前用户的身份调用:适用请求由客户(浏览器)发起,引导用户从认证平台登录后,以登录的用户的访问令牌作为服务调用的身份认证信息。

配置和调用步骤：首先按照基于 OAuth2 的单点登陆的说明做好配置，例如通过调用 app 服务查询 secret=b0bff92c-e3e3-4ead-9f57-d85af21577ee 的应用详情：

```
SSO.getAPIClient().parameter("secret", "b0bff92c-e3e3-4ead-9f57-d85af21577ee").call("/app")
```

返回结果为 JSONObject 对象，查看获取应用详细信息说明

2、以指定的用户身份(提供用户名和密码)调用：适用请求不是由客户发起，例如定时任务，程序模拟请求等。

配置和调用步骤：配置 OAuth2-config.properties 文件，与基于 OAuth2 的单点登陆的第二步一致（不需要配置 SSOFilter），例如以 tester 的身份通过调用 app 服务查询 secret=b0bff92c-e3e3-4ead-9f57-d85af21577ee 的应用详情：

```
APIClient.getInstance().setUserName("tester").setPassword("testerPassword").parameter("secret", "b0bff92c-e3e3-4ead-9f57-d85af21577ee").call("/app")
```

返回结果为 JSONObject 对象，查看获取应用详细信息说明

已提供的服务列表

R1 云认证鉴权平台提供的服务列表包括如下文所示，在具体调用时有两种方式：

1. 直接通过 OAuth2 协议调用，获取到访问令牌后请参考使用访问令牌调用 API
2. 使用客户端调用，客户端封装了 OAuth2 协议的过程，使用更加简单，请参考使用客户端调用 API

1.5.1.获取用户信息

```
调用方式：方案一： APIClient client=SSO.getAPIClient();//集成了 SSO

          方案二： APIClient client=APIClient.getInstance().setUserName("userName").setPassword("password");//直接指定用户

          获取 client 对象后，再根据如下代码获取 json 对象：

          JSONObject resut1 = client.call("/user");
```

返回结果示例：

```
{

"personUuid" : "4089e314403d26ae01403d26aee90000",
```

```
"userId" : "tester",

"fullName" : "tester",

"email" : "tester@tester.com",

"accountType" : "1",

"accountStat" : "2",

"certificateId" : "",

"loginWay" : "",

"cardType" : "0",

"idNum" : "",

"personCode" : "",

"orgName" : "",

"gender" : "男",

"orderNo" : "1",

"telNo" : "",

"otherInfo" : "",

"createTime" : "2013-08-02 11:51:20",

"createUser" : "",

"updateTime" : "2013-08-20 11:35:35",

"updateUser" : "",

"isAdministrator" : "false"

}
```


1.5.2.获取应用详细信息

调用方式：方案一： APIClient client=SSO.getAPIClient();//集成了 SSO

方案二： APIClient client=APIClient.getInstance().setUserName("userName").setPassword("password");//直接指定用户

获取 client 对象后，再根据如下代码获取 json 对象：

```
JSONObject resut2 = client.parameter("secret", "b0bff92c-e3e3-4ead-9f57-d85af21577ee")
                                .call("/app");
```

返回结果示例：

```
{
        "id" : "297eff024076626901407662690 50000",
        "creationDate" : "2013-08-13 14:34:55",
        "appID" : "testapp",
        "memo" : "1",
        "expireDuration" : "604800",
        "appName" : "测试应用",
        "allowedImplicitGrant" : "false",
        "allowedClientCredentials" : "false",
        "secret" : "b0bff92c-e3e3-4ead-9f57-d85af21577ee",
        "skipConsent" : "true",
        "includePrincipal" : "false",
        "iconURL" : "",
        "useRefreshTokens" : "false",
        "resourceserverId" : "",
        "status" : "1",
        "orderNO" : "100",
        "clientScope" : "",
        "secretTime" : "2013-08-20 17:20:41",
        "redirectUris" : "",
        "developerId" : "0000000000000000001",
        "developerName" : "中软国际-部门一"
}
```


1.5.3.注册开发商

调用示例一（通过客户端调用,客户端的配置请参考使用客户端调用 API 部分）

```
 /*
  * 参数说明:
  * userId 登录帐号（必填）
  * fullName 全名 （必填）
```

```
   * email 邮箱 （必填）

   * personUrl 个人主页（必填）

   * appID  应用 ID（必填）

   * appName 应用名称 （必填）

   * redirectUris 重定向地址

   */
```

方案一： APIClient client=SSO.getAPIClient();//集成了 SSO

方案二： APIClient client=APIClient.getInstance().setUserName("userName").setPassword("password");//直接指定用户

获取 client 对象后，再根据如下代码获取 json 对象：

```
JSONObject resut3=client.parameter("userId", "icss")

         .parameter("email", "icss@163.com")

         .parameter("fullName", "icss")

         .parameter("personUrl", "http://icss.com.cn")

         .parameter("appID", "OA-CLOUD")

         .parameter("appName", "云办公自动化")

         .parameter("redirectUris", "http://localhost:8080/R1AuthricationServer/client.jsp")

         .call("/registerDeveloper");

resut3.optString("activeURL");//获取用户激活的 URL 地址
```

返回结果(JSON)：

```
{

  "personuuid" : "4089e35e427866c00142788d63850005",

  "activeURL" : "http://localhost:8080/R1AuthricationServer/resetPasswod",

  "authorize_url" : "http://localhost:8080/oauth2/authorize",

  "access_token_url" : "http://localhost:8080/oauth2/access_token",

  "api_url" : "http://localhost:8080/api",

  "client_secret" : "29d4daa5-5044-4b43-b4bc-34644da83e26"

}
```

1.5.4.创建用户

调用示例一（通过客户端调用,客户端的配置请参考使用客户端调用 API 部分）

```
        /*
         * 参数说明：
         * userId 登录帐号（必填）
         * fullName 全名 （必填）
         * email 邮箱 （必填）
         * cardType 证件类型（非必填）
         * idNum  证件号码（非必填)
         * personCode 人员编码（非必填)
         * orgName 用户单位（非必填)
         * gender 性别 （非必填）
         * orderNo 排序后 （非必填）
         * telNo 联系电话 （非必填）
         * otherInfo 备注 （非必填）
         */
```

方案一： APIClient client=SSO.getAPIClient();//集成了 SSO

方案二： APIClient client=APIClient.getInstance().setUserName("userName").setPassword("password");//直接指定用户

获取 client 对象后，再根据如下代码获取 json 对象：

```
JSONObject resut4=client.parameter("userId", "test")
                                                .parameter("fullName", "测试")
                                                .parameter("email", "test@163.com")
                                                .parameter("cardType", "身份证")
                                                .parameter("idNum", "222")
                                                .parameter("personCode", "00003")
                                                .parameter("orgName", "中软")
                                                .parameter("gender", "男")
                                                .parameter("orderNo", "001")
                                                .parameter("telNo", "13233913419")
```

```
                                                .parameter("otherInfo", "备注")
                                                .call("/createUser");
```

返回结果(JSON)：

```
创建用户成功：{
  "result" : "success",
  "personUuid" : "4089e35e427866c00142788d63850005"
}
创建用户失败：{
  "result" : "failure",
  "error" : "失败原因"
}
```

1.5.5.更新用户信息

调用示例一（通过客户端调用,客户端的配置请参考使用客户端调用 API 部分）

```
          /*
           * 参数说明：
           * personUuid 用户 UUID （必填）
           * userId 登录帐号（必填）
           * fullName 全名 （必填）
           * email 邮箱 （必填）
           * cardType 证件类型（非必填）
           * idNum  证件号码（非必填）
           * personCode 人员编码
           * orgName 用户单位
           * gender 性别 （非必填）
           * orderNo 排序后 （非必填）
           * telNo 联系电话 （非必填）
           * otherInfo 备注 （非必填）
           */
方案一： APIClient client=SSO.getAPIClient();//集成了 SSO
```

```
方案二： APIClient client=APIClient.getInstance().setUserName("userName").setPassword
("password");//直接指定用户

获取 client 对象后，再根据如下代码获取 json 对象：

JSONObject resut6=client.parameter("personUuid", "4089e35e427866c00142788d63850005")
                                                .parameter("userId", "test")
                                                .parameter("fullName", "测试 1")
                                                .parameter("email", "tes1t@163.com")
                                                .parameter("cardType", "身份证 1")
                                                .parameter("idNum", "2221")
                                                .parameter("personCode", "00001")
                                                .parameter("orgName", "中软 1")
                                                .parameter("gender", "男")
                                                .parameter("orderNo", "002")
                                                .parameter("telNo", "13233913419")
                                                .parameter("otherInfo", "备注 1")
                                                .call("/updateUser");
```

返回结果(JSON)：

```
更新成功：{"result" : "success"}

更新失败：{"result" : "failure", "error" : "失败原因"}
```

1.5.6.删除用户

调用示例一（通过客户端调用,客户端的配置请参考使用客户端调用 API 部分）

```
/*

 * 参数说明：

 * personUuids 用户 UUID （必填）

 *            单条删除 例如：parameter("personUuids", "4089e35142736fd6014273fb5c47000
8");

 *            批量删除 例如：parameter("personUuids", "4089e35142736fd60142738d2108000
5,4089e35142736fd6014273fb5c470008,.......")

 */

方案一： APIClient client=SSO.getAPIClient();//集成了 SSO
```

方案二： APIClient client=APIClient.getInstance().setUserName("userName").setPassword("password");//直接指定用户

获取 client 对象后，再根据如下代码获取 json 对象：

```
JSONObject resut6=client.parameter("personUuids", "4089e35142736fd60142738d21080005,4089e35142736fd6014273fb5c470008")
                                .call("/deleteUser");
```

返回结果(JSON)：

删除成功：{"result" : "success"}

删除失败：{"result" : "failure", "error" : "失败原因"}