**OA登陆认证接口说明文档**

**日期：2020-09-12**

**1.获取token接口说明**

接口是rest接口，在请求oa rest接口时必须传入token，获取接口说明：

http请求方式：POST

http://ip:port/seeyon/rest/token

原始的HTTP请求如下所示

POST http://127.0.0.1/seeyon/rest/token HTTP/1.1

Host: 127.0.0.1

Accept: application/json

Content-Type: application/json

{"userName":"rest","password":"123456"}

请求实例:

http://IP:port/seeyon/rest/token/{restusername}/{password}

ip和port为OA的访问地址和端口

<http://58.218.126.21:29810/>

rest账户：rest 密码：e2365293-2569-452c-82fb-feafb17cbc85

参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| userName | 是 | REST用户登录名（由OA提供） |
| password | 是 | REST用户密码（由OA提供） |

返回说明

成功返回JSON字符串，失败返回-1。

成功：

{"id": "013a2a1e-a0b0-4f66-b533-da0563f89c6c"}

失败：

{"id": "-1"}

代码示例：

|  |
| --- |
| **public static String getToken() {**  **String token = "";**  **String url = "http://127.0.0.1:80/seeyon/rest/token";**  **CloseableHttpClient client = HttpClients.createDefault();**  **HttpPost httpPost = null;**  **HttpResponse response = null;**  **httpPost = new HttpPost(url);**  **httpPost.setHeader("Content-Type", "application/json;charset=utf-8");**  **String requestParams = "{\"userName\":\"rest\",\"password\":\"ab571a1b-f731-444f-8521-a8d4a8727d9e\"}";**  **StringEntity postingString = new StringEntity(requestParams, "utf-8");**  **httpPost.setEntity(postingString);**  **try {**  **response = client.execute(httpPost);**  **response.setHeader("Cache-Control", "no-cache");**  **System.out.println(response.getStatusLine().getStatusCode());**  **if (response.getStatusLine().getStatusCode() == HttpStatus.SC\_OK) {**  **String resultString = EntityUtils.toString(response.getEntity(), "utf-8").replaceAll(" ", "");**  **JSONObject jsonObject = JSON.parseObject(resultString);**  **token = (String) jsonObject.get("id");**  **}**  **} catch (IOException e) {**  **e.printStackTrace();**  **}**  **return token;**  **}** |

**传递Token**

**在HTTP请求的header中，使用token属性，例如post请求：**

HttpPost post = new HttpPost（）

post.addHeader("token",token);

**2.登陆认证接口地址：**

http请求方式：POST

http://ip:port/seeyon/rest/verifyLogin/login

Accept: application/json

Content-Type: application/json

{" loginName":"7903918147349529931"} 就是uuid 这个是随便写的字符串

参数说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **是否必须** | **说明** |
| loginName | 是 | 参数在body中就是uid  业务系统绑定OA账号信息，用户输入登录名，后台传uid |
| token | 是 | 参数在header中 |

返回说明

返回结果为json。

**成功：**

{

    "msg": "请求成功",

    "code": 0,

    "data": {

        "loginName": "7903918147349529931", //这个是随便写的字符串

    }

}

**失败：**

{

    "msg": "xxxxx",

    "code": 1,

 }

|  |  |
| --- | --- |
| **状态码** | **说明** |
| 0 | 成功 |
| 1 | 账号不存在 |
| 2 | 密码不正确 |

**代码示例：**

|  |
| --- |
| **public static String doPost(String name,String pwd,String token) throws IOException {**  **CloseableHttpClient client = HttpClients.createDefault();**  **HttpPost post = new HttpPost("http://127.0.0.1:80/seeyon/rest/verifyLogin/login");**  **Map<String, Object> map = new HashMap<>();**  **map.put("loginName", name);**  **String pairs=JSON.toJSONString(map);**  **StringEntity formEntity = new StringEntity(pairs, "UTF-8");**  **post.setHeader("Content-Type", "application/json;charset=utf-8");**  **//设置post请求头**  **post.addHeader("token",token);**  **post.setEntity(formEntity);**  **// 使用HttpClient发起请求，返回response**  **CloseableHttpResponse response = client.execute(post);**  **response.setHeader("Cache-Control", "no-cache");**  **String resultString = "";**  **if (response.getStatusLine().getStatusCode() == HttpStatus.SC\_OK) {**  **resultString = EntityUtils.toString(response.getEntity(), "utf-8").replaceAll(" ", "");**  **}**  **return resultString;**  **}** |