OA接口说明

时间：2020-09-27

# 1．接口使用说明

此接口适用第三方系统在OA中发起表单流程，前提必须在OA中设计相应的流程表单。

## 获取token

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}**

|  |  |  |
| --- | --- | --- |
| **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） |

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（仅供参考）：

|  |
| --- |
| 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);

## OA流程发起接口调用

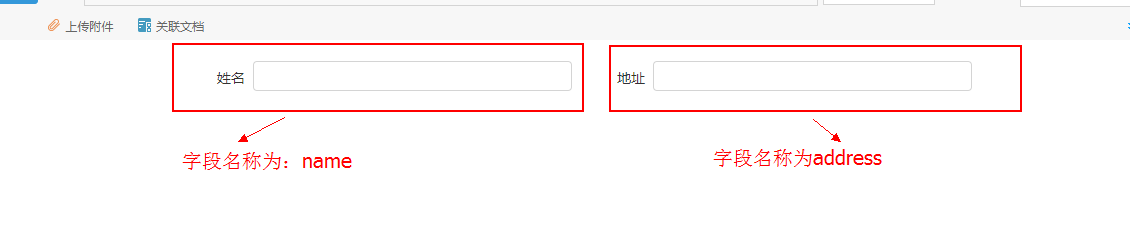
1. http请求方式：POST
2. http://ip:port/seeyon/rest/flow/{templateCode}
3. 例如：
4. <http://127.0.0.1/seeyon/rest/flow/A0001>
5. 传递数据方式为json

**参数说明：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **数据类型** | **是否必须** | **说明** |
| templateCode | String | 是 | 模板编号 |
| senderLoginName | String | 是 | 发起者的登录名（登录协同的登录名） |
| subject | String | 是 | 协同的标题 |
| param | String | 是 | 状态 (0表示发起，1表示待发) |
| transfertype | String | 是 | json(data为json格式) |
| data | String | 是 | 为json字符串 |
| token | String | 是 | 1.1说明中获取的值 |
| 说明：token一定要在请求头中传递 | | | |

data说明：

data是json字符串，data中key字段需要OA和第三方系统约定好，比方说OA中有 一个表单，如图：



Data的数据结构为：

|  |
| --- |
| Map<String, Object> dataMap = new HashMap<>();  dataMap.put("name", "11111111");  dataMap.put("address", "2222222");  data= com.alibaba.fastjson.JSONObject.toJSONString(dataMap) |

代码示例：

|  |
| --- |
| String token = TokenUtil.getToken();  System.out.println(token);  String templateCode=”bpm0916”;  String url = “http://localhost:80/seeyon/rest/flow/”+ templateCode;  try (CloseableHttpClient client = HttpClients.createDefault()) {  HttpPost hpost = new HttpPost(url);  HttpResponse hResponse = null;  hpost.setHeader("Content-Type", "application/json;charset=utf-8");  hpost.addHeader("token", token);  Map<String, Object> data1 = new HashMap<String, Object>();  data1.put("templateCode", templateCode);  data1.put("senderLoginName", "yanyi");  data1.put("subject", "bpm集成测试六六");  Map<String, Object> dataMap = new HashMap<>();  dataMap.put("name", "11111111");  dataMap.put("address", "2222222");  data1.put("data", com.alibaba.fastjson.JSONObject.toJSONString(dataMap));  data1.put("param", "0");  data1.put("transfertype", "json");  String requestParam = JSONObject.toJSONString(data1);  StringEntity entity = new StringEntity(requestParam, "UTF-8");  hpost.setEntity(entity);  hResponse = client.execute(hpost);  if (hResponse.getStatusLine().getStatusCode() == HttpStatus.SC\_OK) {  HttpEntity httpEntity = hResponse.getEntity();  String result = EntityUtils.toString(httpEntity, "UTF-8");  System.out.println(result);  }  } catch (Exception e) {  e.printStackTrace();  } |

接口返回说明：

接口返回的是字符串，也就是发起事项的id，这个值需要和第三方系统的表单做关联，主要是为了关联OA流程结束后的审批意见。

流程结束后的审批意见的回写需要第三方系统提供一个接口，由oa调用把相关表单审批意见回写到第三方系统。