

JEECG 智能开发平台

Online 表单对外接口使用文档

2017/03/28

www.jeecg.org

Jeecg 社区

目 录

| | |
|-----------------------------------|-----------|
| 1. 密钥配置 | 3 |
| 2. 请求参数示例与说明 | 3 |
| 3. 表单信息查询接口 | 7 |
| 3.1. 请求地址 | 7 |
| 3.2. 求参数说明 | 7 |
| 3.3. 返回结果 | 7 |
| 3.3.1. 单表模型 | 7 |
| 3.3.2. 主子表模型 | 8 |
| 3.3.3. 返回结果 JSON 参数说明 | 9 |
| 4. 表单信息删除接口 | 10 |
| 4.1. 请求地址 | 10 |
| 4.2. 请求参数说明 | 10 |
| 4.3. 返回结果 | 10 |
| 5. 表单信息添加接口 | 11 |
| 5.1. 请求地址 | 11 |
| 5.2. 请求参数说明 | 11 |
| 5.2.1. 单表 DATA 数据 json 格式: | 11 |
| 5.2.2. 主表附表 data 数据 json 格式 | 12 |
| 5.3. 返回结果 | 12 |
| 6. 表单信息更新接口 | 13 |
| 6.1. 请求地址 | 13 |
| 6.2. 请求参数说明 | 13 |
| 6.2.1. 单表 DATA 数据 json 格式 | 13 |
| 6.3. 返回结果 | 13 |

1. 秘钥配置

org.jeecgframework.web.cgform.controller.build.[CgFormDataController](#).SIGN_KEY

说明： 暂时代码写死，后期可扩展

2. 请求参数示例与说明

请求参数分为两个部分 第一部分是参数

| 参数名 | 是否必须 | 类型 | 描述 | 示例 |
|--------------|------|------------|-------------------|--|
| body | 是 | JSONString | 请求内容参数以 json 形式存储 | <pre>body: { "id": "40281381537e969401537eb9902d0006", "tableName": "jform_contact", "data": { "sex": "0", "name": "张山丰" } }</pre> |
| X-JEECG-SIGN | 是 | String | 在请求头里面添加，用于验证签名 | <pre>"X-JEECG-SIGN" : "735CE07A2AB9C1872983B09C85A770D9"</pre> |

http://地址:端口/项目名称/api/cgFormDataController.do?addFormInfo

请求代码示例

Header 添加签名

```
"X-JEECG-SIGN" : "735CE07A2AB9C1872983B09C85A770D9"
```

请求参数名为 body 内容为 json 字符串

```
body: {
  "id": "40281381537e969401537eb9902d0006",
  "tableName": "jform_contact",
  "data": {
    "sex": "0",
    "name": "张山丰"
  }
}
```

返回

```
{
  "msg": "新增表单数据成功",
  "success": true
}
```

签名规则:

将 body 对象转换为 json 字符串传入 map 里对数据 body 进行签名。并将签名结果放在请求头中 **X-JEECG-SIGN**

签名示例代码:

```
String key="26F72780372E84B6CFAED6F7B19139CC47B1912B6CAED753";
JSONObject jsonObject=new JSONObject();
jsonObject.put("id","40281381537e969401537eb9902d0006");
jsonObject.put("tableName","jform_contact");
JSONObject data=new JSONObject();
data.put("name","张山丰");
data.put("sex","0");
jsonObject.put("data",data);
String body=jsonObject.toJSONString();
Map param=new HashMap();
param.put("body",body);

String sign=SignatureUtil.sign(param,key);
```

代码示例如下:

```
package test;

import com.alibaba.fastjson.JSONObject;
import org.jeecgframework.web.cgform.util.SignatureUtil;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.ConnectException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.HashMap;
import java.util.Map;

public class HttpUtil {

    /**
     * 发起 https 请求并获取结果
     */
}
```

```

* @param requestUrl
*         请求地址
* @param requestMethod
*         请求方式（GET、POST）
* @param outputStr
*         提交的数据
* @return JSONObject(通过 JSONObject.get(key)的方式获取 json 对象的属性值)
*/
public static JSONObject httpRequest(String requestUrl,
    String requestMethod, String outputStr,String sign) {
    JSONObject jsonObject = null;
    StringBuffer buffer = new StringBuffer();
    HttpURLConnection httpUrlConn = null;
    try {
        // 创建 SSLContext 对象，并使用我们指定的信任管理器初始化
        URL url = new URL(requestUrl);
        httpUrlConn = (HttpURLConnection) url.openConnection();
        httpUrlConn.setDoOutput(true);
        httpUrlConn.setDoInput(true);
        httpUrlConn.setUseCaches(false);
        httpUrlConn.setRequestProperty("X-JEECG-SIGN",sign);
// httpUrlConn.setRequestProperty("content-type", "text/html");
        // 设置请求方式（GET/POST）
        httpUrlConn.setRequestMethod(requestMethod);
        if ("GET".equalsIgnoreCase(requestMethod))
            httpUrlConn.connect();

        // 当有数据需要提交时
        if (null != outputStr) {
            OutputStream outputStream = httpUrlConn.getOutputStream();
            // 注意编码格式，防止中文乱码
            outputStream.write(outputStr.getBytes("UTF-8"));
            outputStream.close();
        }

        // 将返回的输入流转换成字符串
        InputStream inputStream = httpUrlConn.getInputStream();
        InputStreamReader inputStreamReader = new InputStreamReader(
            inputStream, "utf-8");
        BufferedReader bufferedReader = new BufferedReader(
            inputStreamReader);

        String str = null;
        while ((str = bufferedReader.readLine()) != null) {

```

```

        buffer.append(str);
    }
    bufferedReader.close();
    inputStreamReader.close();
    // 释放资源
    inputStream.close();
    inputStream = null;
    httpUrlConn.disconnect();
    jsonObject = JSONObject.parseObject(buffer.toString());
    // jsonObject = JSONObject.fromObject(buffer.toString());
} catch (ConnectException ce) {
    org.jeecgframework.core.util.LogUtil
        .info("Weixin server connection timed out.");
} catch (Exception e) {
    //e.printStackTrace();
    org.jeecgframework.core.util.LogUtil.info("https request error:{"
        + e.getMessage());
}finally{
    try {
        httpUrlConn.disconnect();
    }catch (Exception e) {
        e.printStackTrace();
        org.jeecgframework.core.util.LogUtil.info("http close error:{" + e.getMessage());
    }
}
return jsonObject;
}

public static void main(String args[]){
    String key="26F72780372E84B6CFAED6F7B19139CC47B1912B6CAED753";
    JSONObject jsonObject=new JSONObject();
    jsonObject.put("id","40281381537e969401537eb9902d0006");
    jsonObject.put("tableName","jform_contact");
    JSONObject data=new JSONObject();
    data.put("name","張山丰");
    data.put("sex","0");
    jsonObject.put("data",data);
    String body=jsonObject.toJSONString();
    Map param=new HashMap();
    param.put("body",body);

    String sign=SignatureUtil.sign(param,key);

    JSONObject
resp=HttpUtil.httpRequest("http://localhost:8080/jeecg/api/cgFormDataController.do?addFor

```

```

mlInfo", "POST", "body="+body, sign);
    System.out.println(resp.toJSONString());
}
}

```

3. 表单信息查询接口

3.1. 请求地址

http://地址:端口/项目名称/api/cgFormDataController.do?getFormInfo

具体参考请求参数说明

请求方式: get/post 请求

编码: utf-8

3.2. 求参数说明

| 参数名 | 是否必须 | 类型 | 描述 | 示例 |
|-----------|------|--------|-----------------------|----------------------------------|
| tableName | 是 | String | Online 表名 | jfrom_le_demo |
| id | 是 | String | 数据 id (单表 id/主表 id) | 40281381537e969401537eb9902d0005 |

3.3. 返回结果

3.3.1. 单表模型

单表返回 json 格式:

```

{
  "success": true,
  "tableData": {
    "id": "40281381537e969401537eb9902d0005",
    "create_name": "管理员",
    "create_by": "admin",
    "create_date": 1458112595000,
    "update_name": null,
    "update_by": null,
    "update_date": null,
    "sys_org_code": "A02",
    "sys_company_code": "A0",
    "name": "1",
    "age": 1,
    "job": "",
    "job_desc": "1",
  }
}

```

```

        "bpm_status": null,
        "sex": ""
    },
    "msg": "操作成功",
    "subTableDate": null,
    "tableType": 1
}

```

3.3.2. 主子表模型

主子表返回 josn 格式:

```

{
  "success": true,
  "tableData": {
    "id": "402813815384ba04015384c5c3e00001",
    "create_name": null,
    "create_by": null,
    "create_date": null,
    "update_name": "管理员",
    "update_by": "admin",
    "update_date": 1458576000000,
    "sys_org_code": null,
    "sys_company_code": null,
    "name": "1",
    "sex": "1",
    "remark": "1",
    "bpm_status": null
  },
  "msg": "操作成功",
  "subTableDate": {
    "jform_le_submany": [
      {
        "id": "402813815384e2df015384e39a270001",
        "create_name": "管理员",
        "create_by": "admin",
        "create_date": 1458144000000,
        "update_name": "管理员",
        "update_by": "admin",
        "update_date": 1458576000000,
        "sys_org_code": "A02",
        "sys_company_code": "A0",
        "name": "1",
        "sex": "1",
        "remark": "1",

```



```

        "bpm_status": "1",
        "main_id": "402813815384ba04015384c5c3e00001"
    }
],
"jform_le_subone": [
    {
        "id": "402813815384ba04015384c5c4320002",
        "create_name": "管理员",
        "create_by": "admin",
        "create_date": 1458144000000,
        "update_name": "管理员",
        "update_by": "admin",
        "update_date": 1458576000000,
        "sys_org_code": "A02",
        "sys_company_code": "A0",
        "name": "1",
        "sex": "1",
        "remark": "1",
        "bpm_status": "1",
        "main_id": "402813815384ba04015384c5c3e00001"
    }
]
},
"tableType": 2
}

```

3.3.3. 返回结果JSON参数说明

| 参数名 | 是否必须 | 描述 | 示例 |
|-------------------------|------|------------------------|--------------------|
| success | 是 | 处理状态 true:成功; false:失败 | true |
| msg | 是 | 返回的信息 | 操作成功 |
| tableType | 是 | 表单类型 | 1 单表; 2 主表以及其关联的附表 |
| 以下字段 success 为 true 时返回 | | | |
| tableData | 是 | 单表/主表表数据 | Json 格式 |
| subTableDate | 是 | 附表数据 | Json 格式 |

4. 表单信息删除接口

4.1. 请求地址

http://地址:端口/项目名称/api/ cgFormDataController.do?deleteFormInfo

请求方式: get/post 请求

编码: utf-8

4.2. 请求参数说明

| 参数名 | 是否必须 | 类型 | 描述 | 示例 |
|-----------|------|--------|---------------------|--------------------------------------|
| tableName | 是 | String | Online 表名 | jfrom_le_demo |
| id | 是 | String | 数据 id (单表 id/主表 id) | 40281381537e9694015 37eb9902d0005 |

4.3. 返回结果

返回 json 格式

```
{
  "success": true,
  "msg": "删除成功"
}
```

json 参数说明

| 参数名 | 是否必须 | 描述 | 示例 |
|---------|------|------------------------|------|
| success | 是 | 删除状态 true:成功; false:失败 | true |
| msg | 是 | 返回的信息 | 删除成功 |

5. 表单信息添加接口

5.1. 请求地址

http://地址:端口/项目名称/api/cgFormDataController.do?addFormInfo

请求方式: get/post 请求

编码: utf-8

5.2. 请求参数说明

| 参数名 | 是否必须 | 类型 | 描述 | 示例 |
|-----------|------|--------|--------------------------------|--------------------------------------|
| tableName | 是 | String | Online 表名 | jfrom_le_demo |
| id | 是 | String | 数据 id (单表 id/主表 id) | 40281381537e96940153 7eb9902d0005 |
| data | 是 | String | 单表数据 json 格式/主表附表 json 数据格式 | |

5.2.1. 单表DATA数据json格式:

```
{  
  "name": "ceshi",  
  "age": 10,  
  "job": "java developer"  
}
```

5.2.2. 主表附表data 数据json格式

```
{
  jform_le_main:[{id="402813815398698b015398698b710000",
    name:"ceshi111111",
    age:10,
    job:"java developer"
  }],
  jform_le_subone:[{main_id="402813815398698b015398698b710000",
    name:"ceshi111111",
    age:10,
    job:"java developer"
  }],
  jform_le_submany:[{main_id="402813815398698b015398698b710000",
    name:"ceshi111111",
    age:10,job:"java developer"
  },
  {name:"ceshi111111",
    age:10,
    job:"java developer"}
  ]
}
```

5.3. 返回结果

返回 json 格式:

```
{
  "success":true,
  "msg":"新增表单数据成功"
}
```

json 参数说明

| 参数名 | 是否必须 | 描述 | 示例 |
|---------|------|-------------------------|----------|
| success | 是 | 处理状态 true:成功 ; false:失败 | true |
| msg | 是 | 返回的信息 | 新增表单数据成功 |

6. 表单信息更新接口

6.1. 请求地址

http://地址:端口/项目名称/api/cgFormDataController.do?updateFormInfo

请求方式: get/post 请求

编码: utf-8

6.2. 请求参数说明

| 参数名 | 是否必须 | 类型 | 描述 | 示例 |
|-----------|------|--------|--------------------------------|--------------------------------------|
| tableName | 是 | String | Online 表名 | jfrom_le_demo |
| id | 是 | String | 数据 id (单表 id/主表 id) | 40281381537e969401 537eb9902d0005 |
| data | 是 | String | 单表数据 json 格式/主表附表 json 数据格式 | |

6.2.1. 单表DATA数据json格式

```
{  
  "name": "ceshi",  
  "age": 10,  
  "job": "java developer"  
}
```

6.3. 返回结果

返回 json 格式:

```
{  
  "success": true,  
  "msg": "更新表单数据成功"  
}
```

json 参数说明

| 参数名 | 是否必须 | 描述 | 示例 |
|---------|------|------------------------|----------|
| success | 是 | 处理状态 true:成功; false:失败 | true |
| msg | 是 | 返回的信息 | 更新表单数据成功 |