接口请求功能

一些参数的对应信息：

`REQ_TYPE`：1-GET；2-POST；3-PUT；4-DELETE（请求类型）

`IF_SAVED`：1-true；0-false（是否保存到本地）

`SAVE_TYPE`：1-CSV；2-TXT；3-XLS；4-JSON（文件保存格式）

展示页查询所有记录

接口url：`/req/queryAll`

请求类型：GET

Input: null

更新记录

接口url：`/req/updateCfg?req_oid=x`

请求类型：GET

Input：REQ_OID

查询对应OID的记录

接口url：`/req/updateCfg`

请求类型：POST

Input:

```
{
    "req_oid": 22,
    "req_name": "test_name",
    "req_desc": "test_desc",
    "req_url": "https://xxx",
    "req_type": 1,
    "req_param": "key1=val1&key2=val2",
    "if_saved": 1,
    "save_path": "d://",
    "save_type": 2,
    "enable": 1
}
```

先通过oid查询记录信息，更新到界面中，点击提交按钮后，才进行更新的操作

更新到数据库表中还差 REQ_OID、LAST_UPD_DATE、LAST_UPD_USER 三个字段

REQ_OID：使用 TC_CFG_REQ_SEQ.NEXTVAL更新

LAST_UPD_DATE：使用系统时间set进去

LAST_UPD_User：使用session里的user，set进去

插入记录

SQL使用merge语法

接口url：`/req/insertCfg`

请求类型：POST

Input:

```
{
    "req_name": "test_name",
    "req_desc": "test_desc",
    "req_url": "https://xxx",
    "req_type": 1,
    "req_param": "key1=val1&key2=val2",
    "if_saved": 1,
    "save_path": "d://",
    "save_type": 2,
    "enable": 1
}
```

更新到数据库表中还差 REQ_OID、LAST_UPD_DATE、LAST_UPD_USER 三个字段

REQ_OID：使用 TC_CFG_REQ_SEQ.NEXTVAL更新

LAST_UPD_DATE：使用系统时间set进去

LAST_UPD_User：使用session里的user，set进去

删除一条记录

接口url：`/req/deleteCfg`

请求类型：delete

Input:

```
{
    "req_oid": 22
}
```

Return:

True or False

批量删除

接口url： `/req/deleteCfg`

请求类型：Post

Input:

```
{
    "oid_list": [12, 15, 33]
}
```

手工执行

前端输入格式为`key1=val1&key2=val2&key2=val3`

```
{
    "key1": "val1",
    "key2": ["val2", "val3"]
}
```

```
如果一个键对应多个值，那么这个键会对应的值会是一个list
如果一个键只对应一个值，那么这个键对应的是一个值
注意：所有的键值对都是字符串
```

接口url： `/req/manual`

请求类型：POST

Input：手工执行，只需要url，请求类型和请求参数即可

```
{
    "req_url": "https://xxx",
    "req_type": 2,
    "req_param": "key1=val1&key2=val2",
    "if_saved": 1,
    "save_path": "D://",
    "save_type": 1
}
```

req_type:

- 1——GET
- 2——POST
- 3——PUT
- 4——DELETE