

Carpo Test Tool 使用说明

一、工具需要的环境

- JDK 1.8
- Eclipse Luna(4.4.2)
- Junit 4
- 导入工程后需要把根目录下的lib文件夹里的jar包导入到工程

二、配置文件config.properties

在使用此工具前需要先配置好config.properties文件:

CasePath 为测试用例所在文件夹,使用者可以自己定义case文件夹的位置,默认为空,默认位置在工程的src/case文件夹

BaseUrl 为前端页面的网址,需要加http://,如果不是80端口需要加上端口号

LoginApi 为页面登录的API,写对于BaseUrl的相对地址

UserName 为登录的用户名

Password 为登录的密码

AuthToken 为认证用户的一个字符串,可以不填

Cases 可以指定需要单独执行的Case,用文件夹名称表示,多个用例以逗号分隔

三、测试用例存放方式

- 测试用例默认存放在项目的case文件夹下,路径可以在config.properties文件中更改
- 测试用例以文件夹形式放在case文件夹下,每个用例以单独文件夹存放,文件夹内有data文件夹:用来存放创建cdc,cdo等的json,expect文件夹:用来存放预期结果的json文件,saveld文件夹:用来存储各种name/id,以json为后缀的用例文件

四、测试用例字段概览

- case中的字段:
 - **action**: 可以为 create, update, delete, query, update, run, expectdata, result
 - **api**: 相对API地址,如/api/flows
 - **sleepTime**: action执行完毕后休眠的时间
 - **data**: 创建、更新时用的json所在文件路径(相对路径)
 - **status**: 期待的状态码

- `expect`:期待返回结果的文件路径(相对路径)
- 一个完整的case示例

```
{
  "case": [
    {
      "action": "create,scheduler,dataflow",
      "api": "/api/schedulers",
      "sleepTime": "3000",
      "data": "case1/data/scheduler.json",
      "status": "201",
      "expect": ""
    }
  ]
}
```

如果有多个,在case的数组里继续添加json片段

五、字段详细说明

所有的标点符号都是英文状态下的标点

1.创建cdc

创建cdc只需要 `create,cdc` 即可

参考API: `/api/dataClasses`

2.创建cdo

创建cdc只需要 `create,cdo` 即可

参考API: `/api/dataObjects`

3.创建dataflow

创建dataflow时,先编写好创建dataflow的json文件,里面的dataObject和dataClass的值需要用cdo和cdc的name替换,然后把name按顺序(cdo,cdc,cdo,cdc...)放在 `create,dataflow` 后面以逗号分隔

参考API: `/api/flows`

4.创建workflow

创建workflow时,先编写好创建workflow的json文件,里面的\$dataflowId和source_cdo的值需要用dataflow和do的name替换,然后把name按顺序放在 `create,workflow` 后面以逗号分隔

参考API: `/api/flows`

5.创建scheduler

创建scheduler需要将json文件里的name需要自己指定,flowName用相应的Flow的name替换,create时要指明是dataflow还是workflow,eg: `create,scheduler,dataflow`

参考API:

6.创建tag

创建tag只需指定name即可,使用 `create,tag` 就可以了

参考API: /api/tags

7.创建user

创建user需要指定name,loginId,password和roles,使用使用 `create,user` 就可以了

参考API: /api/users

8.更新

更新的关键字是 `update`, 格式为 `update,cdc或cdo等, 对应的name`, eg: `update,tag,mytag`

参考API:

9.查询

预期json可以删去你不想要的字段,但要保留json的整体框架不能变

查询的关键字是 `query`, 查询是通用的统一为 `query,cdc或cdo等, 对应的name`, eg: `query,cdc,cdo_name1`

参考API: 与创建的api相同

10.删除

删除的关键字是 `delete`, 删除也是通用的统一为 `delete,cdc或cdo等, 对应的name`, eg: `delete,cdo_name1`

删除本地的name/id,用法 `delete,all-local`, 删除服务器上创建的条目: `delete,all-server`

参考API: 创建的API/removeList

11.预览

预览的关键字是 `preview`, 格式为 `preview,cdo的name, 预览的行数`, eg: `preview,cdo,100`

参考API: /api/dataObjects

12.运行

运行的关键字是 `run`, 格式为 `run,run的flow的类型, 如dataflow, flow的name`, eg: `run,dataflow,dataflow`

参考API: /api/schedulers

13.监控运行状态

监控运行状态的关键字是 `monitor`, 格式为 `monitor,scheduler的name`, eg: `monitor,scheduler`

参考API: /api/executions

六、case示例

1.创建cdc

```
{
  "action": "create,cdc",
  "api": "/api/dataClasses",
  "sleepTime": "0",
  "data": "case1/data/cdc_1.json",
  "status": "201",
  "expect": ""
}
```

2.创建cdo

```
{
  "action":"create,cdo",
  "api":"/api/dataObjects",
  "sleepTime":"0",
  "data":"case1/data/cdo_1.json",
  "status":"201",
  "expect":""
}
```

3.创建dataflow

```
{
  "action":"create,dataflow,student_cdo,student_cdc,student_out1_cdo,student_out1_cdc,student_out2_cdo,student_out2_cdc",
  "api":"/api/flows",
  "sleepTime":"1000",
  "data":"case1/data/dataflow_1.json",
  "status":"201",
  "expect":""
}
```

4.创建scheduler

```
{
  "action":"create,scheduler,dataflow",
  "api":"/api/schedulers",
  "sleepTime":"1000",
  "data":"case1/data/scheduler.json",
  "status":"201",
  "expect":""
}
```

5.监控

```
{
  "action":"monitor,scheduler1",
  "api":"/api/executions",
  "sleepTime":"1000",
  "data":"","
  "status":"200",
  "expect":""
}
```

6.预览

```
{
  "action":"preview,student_out1_cdo,10",
  "api":"/api/dataObjects",
  "sleepTime":"3000",
  "data":"","
  "status":"200",
  "expect":"case1/expect/result1.json"
}
```

7.删除

```
{
  "action":"delete,cdc,student_cdc",
  "api":"/api/dataClasses/removeList",
  "sleepTime":"0",
  "data":"","
  "status":"204",
  "expect":""
}
```

8.查询

```
{  
  "action": "query,cdc,student_cdc",  
  "api": "/api/dataClasses",  
  "sleepTime": "1000",  
  "data": "",  
  "status": "200",  
  "expect": "case1/expect/expect_cdc.json"  
}
```

七、执行并生成测试报告

在项目中右键**build.xml**-->**Run As-->Ant Build**

生成的测试报告在**case**文件夹下的**report**下,在**html**文件夹中有一个**index.html**,打开它就可以看见测试报告了