Mock简介

什么是接口Mock测试?

mock是在测试过程中，对于一些不容易构造/获取的对象，创建一个mock对象来模拟对象的行为。

使用场景

1、接口功能尚未开发完成，此时如何开展接口测试工作?

2、项目开发时涉及到第三方的支付接口时，开发阶段如何进行调试?

3、前端开发已写好页面，后台接口还未实现，前端人员想要调试页面的效果，没有数据

Moco

国人开发基于Java的开源项目，能够快速搭建Mock服务

https://github.com/dreamhead/moco

使用方式 ：
1、单独jar包使用
2、将其框架移植到java项目中

使用方式：jar包运行

https://repo1.maven.org/maven2/com/github/dreamhead/moco-runner/1.1.0/

启动Moco服务

Step1：编写json配置文件

```
[
    {
        "description": "第一个请求响应",
        "response": {
            "text": "hello world"
        }
    }
]
```

Step2：加载jar包启动服务

```
java -jar moco-runner-1.1.0-standalone.jar http -p 9999 -c test.json
```

Step3：打开浏览器输入http://127.0.0.1:9999访问

Moco常用配置参数

命令解释 ：
http -p 9999 指定端口号
-c test.json 指定是哪个json文件

请求信息设置

uri

指定请求资源地址

1、get请求
2、http://127.0.0.1:9999/login

```
[
    {
        "description": "登录请求",
        "request": {
            "uri": "/login"
        },
        "response": {
            "text": "登录成功"
        }
    }
]
```

method

指定请求方法

1、get请求
2、http://127.0.0.1:9999/login

```
[
    {
        "description": "登录请求",
        "request": {
            "uri": "/login",
            "method": "get"
        },
        "response": {
            "text": "登录成功"
        }
    }
]
```

queries

指定查询参数

1、get请求
2、http://127.0.0.1:9999/login?

```
[
    {
        "description": "登录请求",
        "request": {
            "uri": "/login",
            "method": "get",
            "queries": {
                "mobile_phone": "13323234545",
                "pwd": "123456"
            }
        },
        "response": {
            "text": "登录成功"
        }
    }
]
```

forms

指定表单参数

```
[
    {
        "description": "登录请求",
        "request": {
            "uri": "/login",
            "method": "post",
            "forms": {
```

```
                "mobile_phone": "13323234545",
                "pwd": "123456"
            }
        },
        "response": {
            "text": "登录成功"
        }
    }
]
```

json

指定json请求体参数

```
[
    {
        "description": "登录请求",
        "request": {
            "uri": "/login",
            "method": "post",
            "json": {
                "mobile_phone": "13323234545",
                "pwd": "123456"
            }
        },
        "response": {
            "text": "登录成功"
        }
    }
]
```

headers

指定请求头

```
[
    {
        "description": "登录请求",
        "request": {
            "uri": "/login",
            "method": "post",
            "headers": {
                "X-Lemonban-Media-Type": "lemonban.v1"
            }
        },
        "response": {
            "text": "登录成功"
        }
    }
]
```

响应信息设置

status

指定响应状态码

```
[
    {
        "description": "登录请求",
        "request": {
            "uri": "/login",
            "method": "post"
        },
        "response": {
            "status": 200
        }
    }
]
```

headers

指定响应头

```
[
    {
        "description": "登录请求",
        "request": {
            "uri": "/login",
            "method": "post"
        },
        "response": {
            "status": 200,
            "headers" :{
                "content-type" : "application/json"
            }
        }
    }
]
```

cookies

指定响应cookie

```
[
    {
        "description": "登录请求",
        "request": {
            "uri": "/login",
            "method": "post"
        },
        "response": {
            "status": 200,
            "headers": {
                "content-type" : "application/json"
            },
            "cookies": {
                "jsessionid": "1234567"
            }
        }
    }
]
```

json

指定响应json数据

```
[
```

```
    {
        "description": "登录请求",
        "request": {
            "uri": "/login",
            "method": "post"
        },
        "response": {
            "status": 200,
            "headers": {
                "Content-Type" : "application/json"
            },
            "cookies": {
                "jsessionid": "1234567"
            },
            "json":{
                "code": 0,
                "msg": "OK"
            }
        }
    }
]
```

moco中文返回乱码问题：

启动moco服务加上参数　1、接口返回解决乱码问题

```
java -Dfile.encoding=UTF-8 -jar moco-runner-1.1.0-standalone.jar http -p 9090 -c
test.json
```

2、cmd控制台乱码解决：见文档

json字符串使用中括号包裹的所以：一个json字符串中可以同时写请求和注册接口信息
做好分别保存

mock不需要做复杂的逻辑判断，只是简单模拟接口返回

★★★mock需要根据接口文档进行设置请求和响应结果，
只需跑正常测试用例即可，进而提前介入测试和跑通自动化测试