

接口自动化测试介绍

目标

1. 理解接口自动化测试的概念
2. 了解接口自动化测试的实现方式

1. 接口自动化测试

1.1 概念

接口测试：是对系统或组件之间的接口进行测试，主要是校验数据的交换、传递和控制管理过程，以及相互逻辑依赖关系。

自动化测试：是把以人为驱动测试行为转化为机器执行的一种过程。

接口自动化测试：是让程序或工具代替人工自动的完成对接口进行测试的一种过程。

1.2 实现方式

- 使用接口测试工具来实现，比如：JMeter
- 通过编写代码来实现

1.3 接口测试工具的不足

1. 测试数据不好控制
2. 不方便测试加密接口

https://blog.csdn.net/z_202041

Requests库

目标

1. 掌握如何使用Requests库

1. 什么是Requests库？

1.1 介绍

Requests库是用Python编写的，基于urllib，采用Apache2 Licensed开源协议的HTTP库；相比urllib库，Requests库更加方便，可以节约我们大量的工作，完全满足HTTP测试需求；

1.2 安装

```
pip install requests
```

https://blog.csdn.net/z_202041



二、Requests库介绍

2.1 Requests库介绍

1. 使用python语言编写
2. 使用开源协议，基于urllib库做的二次封装
3. Requests库中封装了相应接口测试方法

2.2 Requests库安装及验证

- 安装: `pip install requests`
- 验证: `pip show requests` --> 显示相应的版本信息
- 注意:
 - 安装时，电脑必须连接互联网。

https://blog.csdn.net/z_202041



三、Requests常用请求方法

方法: GET、POST、PUT、DELETE

3.1 GET方法

- 作用: 获取资源 (查询)
- 步骤
 - 导包 `import requests`
 - 调用get方法 `request.get()`
- 示例:
 - `r = request.get("http://www.baidu.com")` # r:为响应对象
- 响应对象:
 - 获取URL: `r.url`
 - 获取响应状态码: `r.status_code`
 - 获取响应信息: `r.text`

https://blog.csdn.net/z_202041

3.1 GET方法

- 作用: 获取资源 (查询)
- 步骤
 - 导包 `import requests`
 - 调用get方法 `request.get()`
- 示例:
 - `r = request.get("http://www.baidu.com")` # r:为响应对象
 - 请求含参数

```
1. http://www.baidu.com?id=1001
2. http://www.baidu.com?id=1001,1002
3. http://www.baidu.com?id=1001&kw=北京
```

- 参数: `params`
 1. 方式1: `params = {"id": 1001}`
 2. 方式2: `params = {"id": "1001,1002"}`
 3. 方式3: `params = {"id": 1001, "kw": "北京"}`

- 响应对象:

- 获取URL: `r.url`

https://blog.csdn.net/z_202041

3.2 POST方法

- 作用：新增资源
- 应用：
 - 导包 `import requests`
 - 调用 `post`方法 `requests.post()`
 - 示例：`request.post(url,json,headers)`
- 参数：
 1. url: 新增接口url地址
 2. json:新增请求报文
 3. Headers: 请求信息头信

响应状态: `r.status_code`

响应信息: `r.json()` # 以json文本形式响应内容

https://blog.csdn.net/z_202041

1. url: 新增接口url地址
2. json:新增请求报文
3. Headers: 请求信息头信

响应状态: `r.status_code`

响应信息: `r.json()` # 以json文本形式响应内容

- 扩展：
 - data与json区别
 - data:字典对象
 - json:json字符串
 - 提示：
 1. 在python中字典对象和json字符串长的一样，但是后台格式是有区别的
 - 如何将字典对象转为json字符串?
 1. 导入 `json`
 2. `json.dumps(字典对象)` # 转换json字符串

https://blog.csdn.net/z_202041

- 扩展：
 - data与json区别
 - data:字典对象
 - json:json字符串
 - 提示：
 1. 在python中字典对象和json字符串长的一样，但是后台格式是有区别的
 - 如何将字典对象转为json字符串?
 1. 导入 `json`
 2. `json.dumps(字典对象)` # 转换json字符串
 - 响应数据 `json()`与`text`区别
 - `json()`:返回类型字典，可以通过键名来获取响应的值
 - `text`:返回的类型为字符串，无法通过键名来获取响应的值
 - 提示：共同点长得都像字典

https://blog.csdn.net/z_202041

3.4 DELETE方法

- 作用：删除资源
- 应用：
 - 导包 `import requests`
 - 调用方法 `delete`
- 响应：
 - 响应状态码：204

六、session对象

6.1 为什么使用session对象

- 说明：session可以自动保存服务器产生的cookies信息，并且自动在下一条请求是附加。

6.2 什么是session

- 说明：一次会话（从客户端和服务器创建请求连接开始，客户端和服务端断开连接结束）

6.3 session获取及应用

- 获取：
- 应用：

https://blog.csdn.net/z_202041