

# api automation testing

AnthonyGao

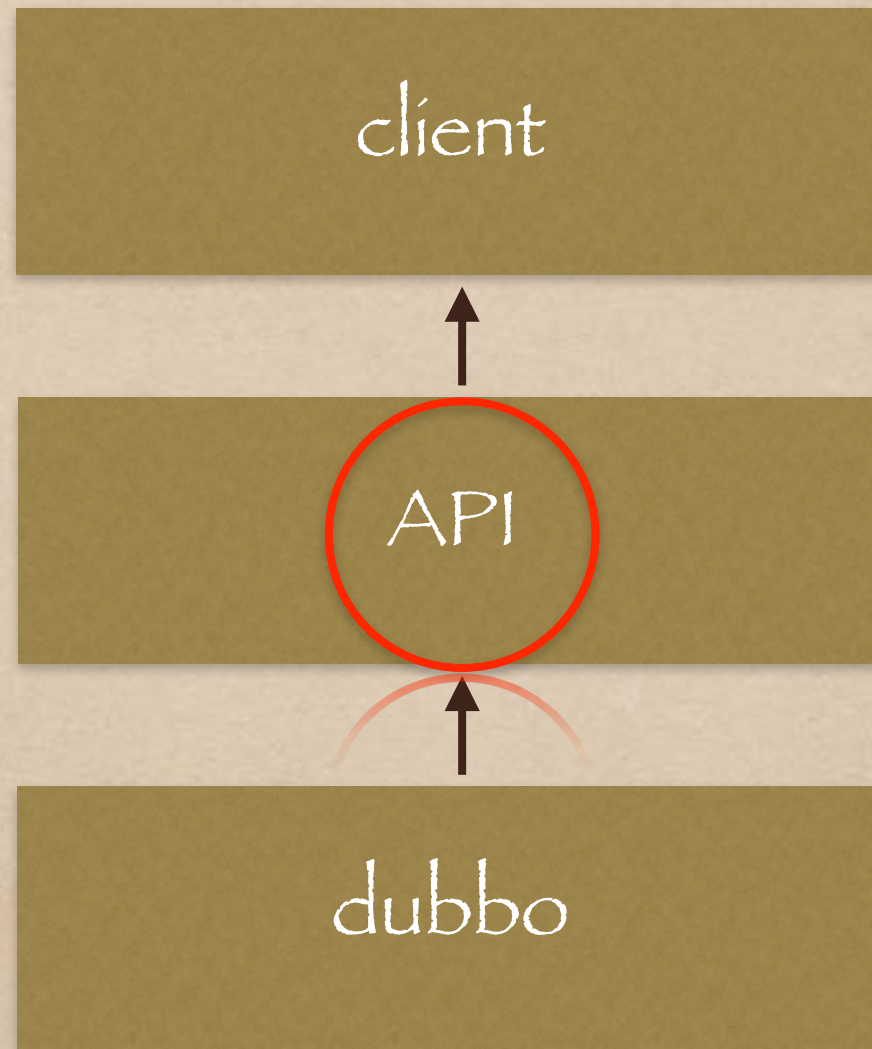


# 什么是API测试

- ◆ API测试又称为接口测试，接口测试是功能测试的一种。它主要借助于单元测试技术，通过模拟上层应用或者系统上层调用接口的应用场景，是对系统接口功能进行测试的一种手段。



# 为什么要API测试?



目前我们的底层用的都是dubbo服务，各个服务系统都不直接对客户端提供服务，api实际上是包装了dubbo接口服务，对客户端提供服务，即使覆盖了dubbo服务测试，却保证不了客户端调用接口的正确性



# 相关工具

- ◆ postman
- ◆ jmeter
- ◆ poster
- ◆ swagger
- ◆ .....



# http请求类型

- ◆ post
- ◆ get
- ◆ delete
- ◆ put
- ◆ file
- ◆ head



# http请求要素

## 1、请求行

请求行包含请求方法(Method)、请求统一资源标识符(URI)、HTTP版本号等

请求方法就是我们所熟悉的POST、GET、HEAD、PUT等

URI就是URL中排除掉Host剩下的部分，也就是资源在服务器本地上的路径

### ▼ General

Request URL: http://admin.daxiaodai.com/api/pub/1.0.0/admin/token?user=gaojiang&password=1234qwer

Request Method: POST

Status Code: ● 200

Remote Address: 10.100.14.76:80

Referrer Policy: no-referrer-when-downgrade



## 2、请求头

请求头主要存放对客户端想给服务端的附加信息，下图框框的部分就是请求头：

```
▼ Request Headers view source  
Accept: application/json, text/plain, */*  
Accept-Encoding: gzip, deflate  
Accept-Language: zh-CN,zh;q=0.9  
Authorization:  
Cache-Control: no-cache  
Connection: keep-alive  
Content-Length: 2  
Content-Type: application/json;charset=UTF-8
```



### 3、请求体

真正需要发给服务端的数据

▼ Query String Parameters

[view source](#)

[view URL encoded](#)

user: gaojiang

password: 1234qwer



# http 请求返回

Headers	Preview	Response	Cookies	Timing
<pre>{code: "0", content: {,...}}   code: "0"   ▼ content: {,...}     expiresIn: 86400     ► params: {userName: "高江",...}       scope: "app"       token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50b3R5IiwiaWF0Ij0y</pre>				



# RestClient介绍

## 建立http请求

```
public RestClient(String endpoint, String service) {  
    this.request = RestAssured.given().contentType("application/json")  
        .config(RestAssuredConfig.newConfig()  
            .encoderConfig(EncoderConfig.encoderConfig().defaultContentCharset("utf-8")))  
        .baseUrl(endpoint + service);  
}
```



# 传header参数

```
public void setHeader(String key, String value) {  
    Header header = new Header(key, value);  
    this.request.header(header);  
}
```



# 传Body参数

## 1. 直接传json

```
public RestClient body(Object value) {  
    this.request.body(value);  
    return this;  
}
```

## 2. 表单格式

```
public RestClient params(String name, Object value) {  
    this.request.param(name, value);  
    return this;  
}
```



# 测试用例

```
/**
 * 发送验证码
 *
 * @throws Exception
 */
@Test(priority = 0)
public void sendPhoneCode() throws Exception {
    RestClient rc = new RestClient(apiDomain, "/pub/1.0.0/code/verification");
    Map<String, String> map = new HashMap<String, String>();
    map.put("cellphone", phone);
    rc.body(JSONArray.toJSONString(map));
    JSONObject jsonObj = rc.post();
    Assert.assertEquals(jsonObj.getString("code"), "0");
}

/**
 * 利用验证码登录并产生token
 *
 * @throws Exception
 */
@Test(priority = 1)
public void login() throws Exception {
    RestClient rc = new RestClient(apiDomain, "/pub/1.0.0/account/token-by-captcha");
    Map<String, String> map = new HashMap<String, String>();
    map.put("phone", phone);
    map.put("captcha", gmc.getCode(msgDB));
    rc.body(JSONArray.toJSONString(map));
    JSONObject jsonObj = rc.post();
    Assert.assertEquals(jsonObj.getString("code"), "0");
    tk = jsonObj.getJSONObject("content").getString("tokenType") + " "
        + jsonObj.getJSONObject("content").getString("token");
}

/**
 * 获取银行卡的银行信息
 *
 * @throws Exception
 */
@Test(priority = 2)
public void getBankInfo() throws Exception {
    RestClient rc = new RestClient(apiDomain, getBankInfoService);
    rc.setHeader("Authorization", tk);
    rc.params("bank_card_no", bankCard);
    JSONObject jsonObj = rc.get();
    Assert.assertEquals(jsonObj.getString("code"), "0");
    Assert.assertEquals(jsonObj.getJSONObject("content").getString("bankName"), "招商银行");
}
```



# Testng管理测试用例

```
<?xml version="1.0" encoding="UTF-8"?>  
<suite name="Rest api test" parallel="true" thread-count="1">  
  <test name="rest api test" preserve_order="true">  
    <classes>  
      <class name="com.lc.app.api.BankCardTest" />  
      <class name="com.lc.app.api.BannerTest" />  
      <class name="com.sdx.d.app.api.DxdMainTest" />  
      <class name="com.lc.app.api.BigWheelTest" />  
    </classes>  
  </test>  
</suite>
```