

接口测试自动化基础

HttpClient

本章大纲

- Get请求
- Post请求

HttpGet方法

Constructor

- public **HttpGet**(String uri)

Throws:

IllegalArgumentException - if the uri is invalid.

extends `HttpRequestBase`

HttpGet方法-发起get请求

- 创建HttpClient对象

```
CloseableHttpClient httpClient = HttpClient.createDefault();
```

- 创建带请求地址的HttpGet对象

```
HttpGet httpGet = new HttpGet("http://xxxxxx");
```

- 设置HttpGet对象属性，如header，cookie等

```
httpGet.setHeader("Content-Type","application/json");
```

- 执行HttpGet请求，获得响应

```
CloseableHttpResponse response = httpClient.execute(httpGet);
```

HttpGet方法-响应解析

- 获取响应实体

```
HttpEntity entity = response.getEntity();
```

- 获取响应内容

```
EntityUtils.toString(entity);
```

- 释放资源

```
EntityUtils.consume(entity);
```

- 断开连接

```
response.close();
```

```
httpClient.close();
```

测试用例回顾

- 请求Header类型: Content-Type=application/json

编号	用例名称	前置条件	请求地址	请求header	请求参数	响应检查点
skulist-1	获取所有商品的sku列表成功	无	GET /common/skuList	Content-Type=application/json	无	code : 200 message : "success" result : 所有商品sku信息列表(略)
skulist-2	获取goodsId=1的商品sku信息成功	无	GET /common/skuList	Content-Type=application/json	goodsId=1	code : 200 message : "success" result : goodsId=1的商品sku信息(略)
skulist-3	获取goodsId=2147483648的商品sku信息失败(超过int最大取值范围)	无	GET /common/skuList	Content-Type=application/json	goodsId=2147483648	code : 400 message : "商品ID不正确"
skulist-4	获取goodsId不存在的商品失败	无	GET /common/skuList	Content-Type=application/json	goodsId=9999	code : 201 message : "商品ID不存在"
skulist-5	goodsId类型不正确	无	GET /common/skuList	Content-Type=application/json	goodsId="1"	code : 400 "message" : "商品ID参数类型不正确"

skulist-1测试用例详解

- 用例： 获取所有商品的sku列表成功

```
try {  
    // 初始化httpClient对象  
    httpClient = HttpClient.createDefault();  
    // 初始化HttpGet请求  
    HttpGet httpGet = new HttpGet("http://study-perf.qa.netease.com/common/skuList");  
    // 执行http请求, 得到response  
    response = httpClient.execute(httpGet);  
    // 打印http返回状态码  
    System.out.println(response.getStatusLine());  
    httpGet.setHeader("Content-Type", "application/json");  
    // 用 HttpEntity得到response返回的实体  
    HttpEntity entity = response.getEntity();  
    if (entity != null) {  
        // 打印body内容  
        System.out.println(EntityUtils.toString(entity, "UTF-8"));  
    }  
    EntityUtils.consume(entity);  
} catch (Exception e) {  
    e.printStackTrace();  
}  
finally {  
    response.close();  
    httpClient.close();  
}
```

skulist-2测试用例详解

- 用例：获取goodsId=1的商品sku信息成功

```
// 初始化httpClient对象
httpClient = HttpClient.createDefault();
// 初始化HttpGet请求
HttpGet httpGet = new HttpGet("http://study-perf.qa.netease.com/common/skuList?goodsId=1")
// 执行http请求, 得到response
response = httpClient.execute(httpGet);
```

```
HTTP/1.1 200 OK
{"message":"success","result":[{"id":3,"searchParamList":[]},
```


skulist-3测试用例详解

- 用例：获取goodsId=2147483648的商品sku信息失败

```
// 初始化httpClient对象
httpClient = HttpClient.createDefault();
// 初始化HttpGet请求
HttpGet httpGet = new HttpGet("http://study-perf.qa.netease.com/common/skuList?goodsId=2147483648");
// 执行http请求, 得到response
response = httpClient.execute(httpGet);
```

HttpGet用法总结

- 注意事项:

- ✓ 请求地址中有多个参数用&符号连接, 例如

`http://aaaaa?id=1&name=tom`

- ✓ 请求参数如包含非英文字符, 需要encode转码,

例如: `URLEncoder.encode("{ \"pId\": \"123456\" }", "UTF-8");`

本章大纲

- Get请求
- Post请求

HTTP请求格式-POST方法

请求 (Request)

请求行	POST /xinhu/index.php?a=check&m=login HTTP/1.1
请求头	Accept: text/html,application/xhtml+xml Accept-Encoding: gzip, deflate Accept-Language: zh-CN, zh; q=0.8, en-US; q=0.5, en; q=0.3 Connection: keep-alive Host: localhost:8032
请求正文	adminuser=YWRtaW4%3A&=123456&rempass=0&button=&jmpass=false&device=1517376146707&adminpass=MTIzNDU2

响应 (Response)

状态行	HTTP1.1 200 OK
响应头	Connection: Keep-Alive Content-Encoding: gzip Content-Length: 1234 Content-Type: text/html; charset=utf-8 Date: Mon, 05 Feb 2018 02:43:40 GMT
响应正文	{"success":true,"face":"http://localhost:8032/xinhu/upload/face/1.jpg"}

HttpPost方法-发起Post请求

- 创建HttpClient对象

```
CloseableHttpClient httpclient = HttpClients.createDefault();
```

- 创建带请求地址的HttpPost 对象

```
HttpPost httpPost = new HttpPost("https://www.xxx.com");
```

- 设置HttpPost对象属性 httpPost.setHeader("Content-Type", "application/json");

- 设置HttpPost 参数

```
StringEntity entity = new StringEntity("para", "utf-8");  
httpPost.setEntity(entity);
```

- 执行HttpPost请求，获取post请求的响应

```
httpclient.execute(httpPost);
```

HttpPost方法-响应解析

- 获取响应实体

```
HttpEntity entity = response.getEntity();
```

- 获取响应内容

```
EntityUtils.toString(entity);
```

- 释放资源

```
EntityUtils.consume(entity);
```

- 断开连接

```
response.close();
```

```
httpClient.close();
```

测试用例回顾

- 请求Header类型: Content-Type=application/json

编号	用例名称	前置条件	请求地址	请求header	请求参数	响应检查点
login-1	登录成功	无	POST /common/fgadmin/login	Content-Type=application/json	phoneArea="86" phoneNumber="20000000000" password="netease123"	code : 200 message : success
login-2	登录失败, phoneArea参数类型不正确	无	POST /common/fgadmin/login	Content-Type=application/json	phoneArea=86 phoneNumber="20000000000" password="netease123"	code : 400 message : 用户名或者密码错误
login-3	登录失败, phoneNumber参数类型不正确	无	POST /common/fgadmin/login	Content-Type=application/json	phoneArea="86" phoneNumber=20000000000 password="netease123"	code : 400 message : 用户名或者密码错误
login-4	登录失败, password参数类型不正确	无	POST /common/fgadmin/login	Content-Type=application/json	phoneArea="86" phoneNumber="20000000000" password=123456	code : 400 message : 用户名或者密码错误
login-5	登录失败, 缺少phoneArea参数	无	POST /common/fgadmin/login	Content-Type=application/json	phoneNumber="20000000000" password="netease123"	code : 400 message : 用户名或者密码错误
login-6	登录失败, 缺少phoneNumber参数	无	POST /common/fgadmin/login	Content-Type=application/json	phoneArea="86" password="netease123"	code : 400 message : 用户名为空
login-7	登录失败, 缺少password参数	无	POST /common/fgadmin/login	Content-Type=application/json	phoneArea="86" phoneNumber="20000000000"	code : 400 message : 用户名或者密码错误

login-1测试用例详解

- 用例：登录成功

```
CloseableHttpClient httpClient = null;
CloseableHttpResponse response = null;
try {
    httpClient = HttpClients.createDefault();
    //创建登录HttpPost
    HttpPost httpPost = new HttpPost("http://study-perf.qa.netease.com/common/fgadmin/login");
    //指定HttpPost的内容类型
    httpPost.setHeader("Content-Type", "application/json");
    //设置HttpPost的内容，设置登录参数
    StringEntity entity = new StringEntity(
        "{ \"phoneArea\":86, \"phoneNumber\": \"20000000000\", \""
        + \"password\": \"netease123\" }", "utf-8");
    httpPost.setEntity(entity);
    //执行请求，完成登录
    response = httpClient.execute(httpPost);
    HttpEntity httpEntity = response.getEntity();
    System.out.println("执行结果是: " + EntityUtils.toString(httpEntity));
    EntityUtils.consume(httpEntity);
}
```

- 响应结果

{"message":"success","code":200}

login-2测试用例详解

- 用例：登录失败，phoneNumber参数类型不正确

```
CloseableHttpClient httpClient = null;
CloseableHttpResponse response = null;
try {
    httpClient = HttpClients.createDefault();
    //创建登录HttpPost
    HttpPost httpPost = new HttpPost("http://study-perf.qa.netease.com/common/fgadmin/login");
    //指定HttpPost的内容类型
    httpPost.setHeader("Content-Type", "application/json");
    //设置HttpPost的内容，设置登录参数
    StringEntity entity = new StringEntity(
        "{ \"phoneArea\":86, \"phoneNumber\": \"20000000\", \""
        + \"password\": \"netease123\" }", "utf-8");
    httpPost.setEntity(entity);
    //执行请求，完成登录
    response = httpClient.execute(httpPost);
    HttpEntity httpEntity = response.getEntity();
    System.out.println("执行结果是: " + EntityUtils.toString(httpEntity));
    EntityUtils.consume(httpEntity);
}
```

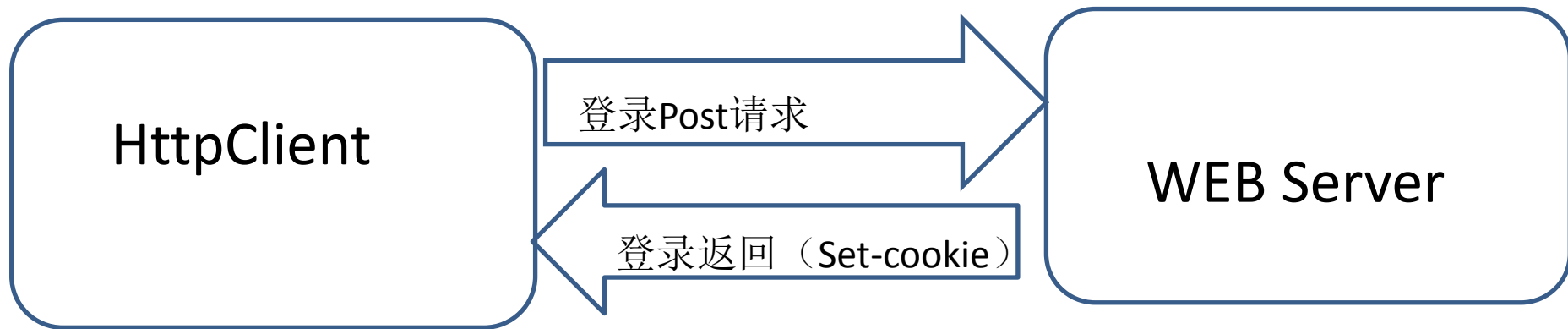
- 响应结果：

{"message":"用户名或者密码错误","code":400}

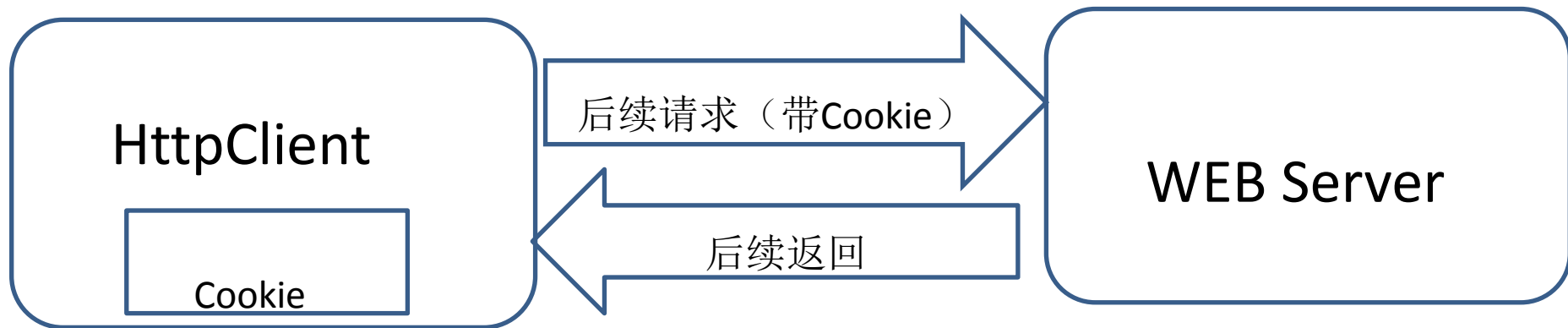
HttpPost用法总结

- 注意事项:
 - ✓请求参数的格式
 - ✓不同httpclient版本对cookie的处理方式

Web登录技术——自动化视角



Web登录技术——自动化视角



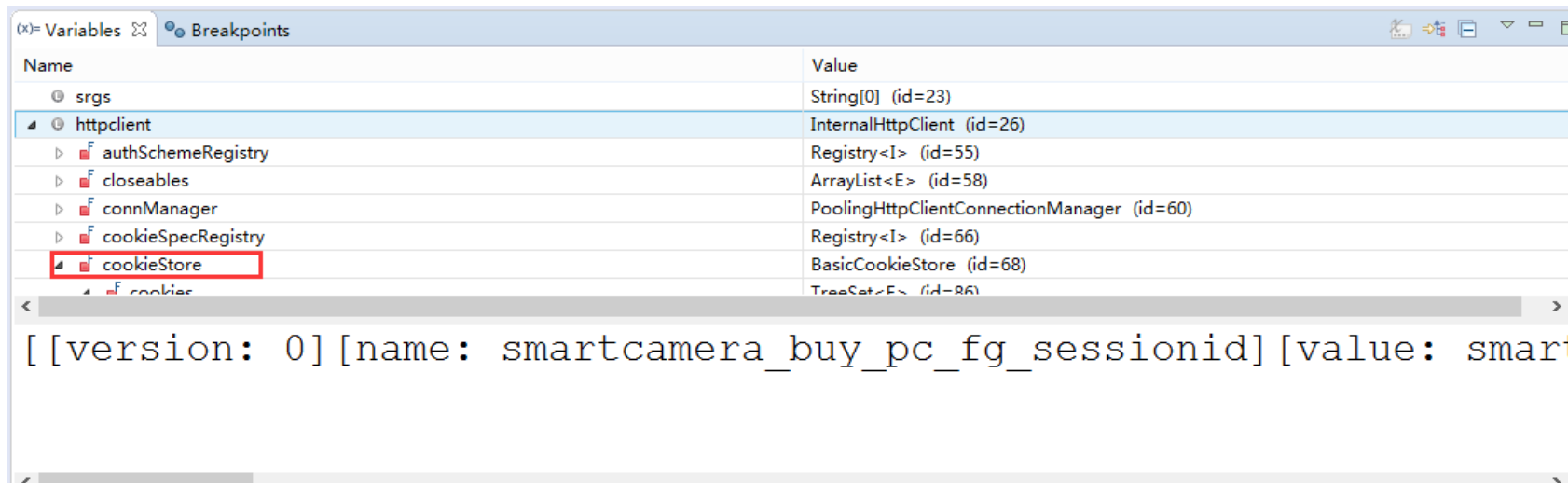
使用HttpClient模拟登录

- 登录过程
- 登录成功后获取到的认证信息： Cookie

```
CloseableHttpClient httpClient=HttpClients.createDefault();
HttpPost httpPost =new HttpPost("http://study-perf.qa.netease.com//common/fgadmin/login");
StringEntity entity = new StringEntity("{\"phoneArea\":\"86\",\"phoneNumber\":\"20000000000\",\"password\":\"123456\"}");
entity.setContentType("application/json");
httpPost.setEntity(entity);
CloseableHttpResponse response = httpClient.execute(httpPost);
HttpEntity httpEntity=response.getEntity();
System.out.println(EntityUtils.toString(httpEntity));
EntityUtils.consume(httpEntity);
```

HttpClient模拟登录

- 登录过程
- 登录成功后获取到的认证信息：Cookie



测试用例回顾

- 请求地址: POST /fgadmin/address/new
- 公共请求参数:
{"receiverName":"张三",
"cellPhone":"12345678901",
"addressDetail":"河北师范大学",
"province":"河北省",
"city":"石家庄市",

编号	用例名称	前置条件	请求参数	响应检查点
addressnew-1	添加收货地址成功	已登录	同公共参数	code : 200 message : "success"
addressnew-11	添加收货地址失败, 缺少province参数	已登录	province为空, 其他为公共参数	code : 400 message : "province参数不存在"
addressnew-20	添加超过6个收货地址失败	已登录 已添加6个收货地址	同公共参数	code : 400 message : "最多只能添加6个地址"

HttpClient登录模拟总结

- 注意事项:
 - ✓ 根据具体登录请求选择HttpEntity具体类型（UrlEncodedFormEntity、StringEntity等等）
 - ✓ 登录请求的Content-Type需要正确设置
 - ✓ 如果不想使用同一个HttpClient对象传递登录信息，可以考虑对需要登录信息请求分别设置Cookie:

```
httpPost.setHeader("Cookie"," mindsparktb_232530392=true;  
mindsparktbsupport_232530392=true");
```


HttpClient设置代理

```
HttpHost proxy = new HttpHost("127.0.0.1",8888);  
RequestConfig requestConfig =  
RequestConfig.custom().setProxy(proxy).build();httpClient=  
HttpClientBuilder.create().setDefaultRequestConfig(requestConfi  
g).build();
```