Intel Leap Ahead Logo

HTTP RESTful API Test Framework

**ReadMe**

Revision: 1.0

Owner: Zhang Yi

Contributors:

## Document Revisions

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rev. | Date | | Owner | Contributors | | Details |
| V1.0 | 6/25/2019 | Zhang Yi | |  | The first version | |
|  |  |  | |  |  | |
|  |  |  | |  |  | |
|  |  |  | |  |  | |
|  |  |  | |  |  | |
|  |  |  | |  |  | |

HTTP RESTful API Test Framework ReadMe

[Document Revisions 2](#_Toc12378783)

[1 Introduction 4](#_Toc12378784)

[2 How to execute HTTP RESTful API Tests 5](#_Toc12378785)

[3 How to add new test cases 6](#_Toc12378786)

# Introduction

这篇ReadMe讲述：

1. 如何执行针对Authentication Center 和AMS User Cloud 服务的HTTP RESTful API自动化测试
2. 如何开发和添加新的测试到自动化运行框架中

# How to execute HTTP RESTful API Tests

我们使用Spring的JUnit框架配合Maven完成自动化测试。它还分两种，一种是使用Spring自带的TestRestTemplate类来调用RESTful接口，另一种是使用Apache的HttpClient工具类来调用HTTP RESTful接口。

* Spring的TestRestTemplate实际上并没有真的启动被测试服务的HTTP端口，而是通过Spring内部的映射调用到要测试的RESTful接口，所以这种测试不需要当读启动被测试服务。
* HttpClient工具类是通过HTTP协议来访问HTTP RESTful接口的，所以第二种测试需要先启动被测试的服务，例如AMS User Cloud服务，否则就会报Connect Refused的异常。

TestRestTemplate测试的运行：

* 我们在ibroker的home folder运行“mvn install”或者“mvn package”的时候，实际上是执行了所有子项目的Unit test的，也包括Auth Center和AMS User cloud的TestRestTemplate测试
* 想要单独执行AuthCenter和AMS User cloud的测试，请到该子项目的目录下运行“mvn test”即运行该项目的所有Unit test
* 如果想要指定运行某个测试类，可以运行”mvn test –Dtest=[测试类名]”，例如：”mvn test –Dtest=AuthApiRestTemplateTest”
* 如果想要运行某个测试类中的某个测试函数，可运行”mvn test –Dtest=[测试类名]#[测试函数名]”，例如：”mvn test –Dtest=AuthApiRestTemplateTest#testLogin1”
* 注意，运行TestRestTemplate的测试无需启动AuthCenter或AMS User cloud的服务程序

HttpClient测试的运行：

* 首先，因为HttpClient的测试需要启动服务程序，无法完全自动化，所以我们在这类测试的Class上加了”@Ignore”的annotation，以防运行”mvn install”的时候自动运行所有测试时出现测试错误退出。所以，在运行此类测试前，先将测试类头的”@Ignore”去掉。
* 其次，进入项目目录运行”mvn spring-boot:run”启动服务程序
* 然后，在项目目录下运行” mvn test –Dtest=[HttpClient测试类名]”
  + 如果是运行Auth Center的HttpClient测试，运行” mvn test–Dtest=AuthApiHttpClientTest“
  + 如果是运行AMS User Cloud的HttpClient测试，运行” mvn test–Dtest=AmsApiHttpClientTest“
  + 同理，如果是运行某个指定的函数，在命令尾加上”#[测试函数名]”即可
* 有时候出现Build Failure，可以先运行下”mvn clean”，再运行以上命令

# How to add new test cases

1. 如何使用Spring TestRestTemplate类来测试RESTful API:

首先，在类的field声明部分声明一个TestRestTemplate的实例。

@Autowired

**private** TestRestTemplate restTemplate;

其次,可参照如下例子：

//Query product with product name

ResponseEntity<String> response = **null**; **//声明一个response**

HttpHeaders headers = **new** HttpHeaders();

headers.add(tokenHeader, tokenHead + token); **//添加token到Header中**

HttpEntity<String> requestEntity = **new** HttpEntity<String>(headers**);//构造request**

response = restTemplate.exchange(baseUrl + "ams/v1/product?name=iagent", HttpMethod.***GET***, requestEntity, String.**class**);

其他，如POST,PUT,DELETE的使用，可以参考AMS User Cloud中的AmsApiRestTemplateTest的例子来开发。

2. 如何使用Apache HttpClient类来测试HTTP RESTful API:

//Query product with product name

**//1. 构造httpClient的一个实例**

CloseableHttpClient httpClient = HttpClientBuilder.*create*().build();

CloseableHttpResponse httpResponse = **null**;

RequestConfig requestConfig = RequestConfig.*custom*().setConnectTimeout(200000).setSocketTimeout(200000000).build();

String productName = "iagent";

String queryStr = "name=" + productName;

String productUrl = "http://localhost:8080/ams/v1/product";

String productQueryUrl = String.*format*("%s?%s", productUrl, queryStr);

**//2. 构造一个HTTP GET 请求的实例**

HttpGet getProductInfo = **new** HttpGet(productQueryUrl);

getProductInfo.setConfig(requestConfig);

**//3. 将token设置到Header中**

getProductInfo.setHeader(tokenHeader, tokenHead + token);

**//4. 执行该HTTP GET请求**

httpResponse = httpClient.execute(getProductInfo); *assertThat*(httpResponse.getStatusLine().getStatusCode()).isEqualTo(HttpStatus.***SC\_OK***);

**//5. 特别注意，每次执行完都要close一次response，否则导致程序hang住**

httpResponse.close();

其他，如POST,PUT,DELETE的使用，可以参考AMS User Cloud中的AmsApiHttpClientTest的例子来开发。