|  |
| --- |
| **UserSupport接口自动化设计与应用** |

|  |  |  |  |
| --- | --- | --- | --- |
| Date  日期 | Version  版本 | Reviser  修订人 | Comments  备注 |
| 2020-05-19 | V0.1 | 陈乙高 | Creation of Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Department部门 | Function  职位 | Name  姓名 | Date  日期 | Signature  签名 |
| Written by  拟定 | **HZST2** | **测试工程师** | **陈乙高** | 2020-05-19 |  |
| Verified by  审核 |  |  |  |  |  |
| Verified by  审核 |  |  |  |  |  |
| Approved by  批准 |  |  |  |  |  |

目录

[1. UserSupport接口介绍与自动化设计的意义 3](#_Toc41314864)

[1.1 UserSupport功能和接口简介 3](#_Toc41314865)

[1.2 UserSupport接口自动化设计的意义 3](#_Toc41314866)

[2. UserSupport接口自动化的设计 3](#_Toc41314867)

[2.1 UserSupport接口自动化技术选型 3](#_Toc41314868)

[2.2 UserSupport接口自动化项目结构 4](#_Toc41314869)

[2.3 开发准备与环境的配置 4](#_Toc41314870)

[2.4 项目开发流程与步骤 5](#_Toc41314871)

[2.4.1 在IDEA中新建maven项目 5](#_Toc41314872)

[2.4.2 创建项目结构 5](#_Toc41314873)

[2.4.3 引入相关依赖 5](#_Toc41314874)

[2.4.4 编写第一个简单的接口自动化例子 5](#_Toc41314875)

[2.4.5 接口的封装 5](#_Toc41314876)

[2.4.6 Test测试报告 5](#_Toc41314877)

[2.5 持续集成与自动构建 5](#_Toc41314878)

[3. 常见的问题与总结 5](#_Toc41314879)

# 1. UserSupport接口介绍与自动化设计的意义

## 1.1 UserSupport功能和接口简介

UserSupport是一款只针对内部员工的bug提交工具。通过这个工具，用户可进行手机端的问题提交并上传至后台操作。这款软件开发的目的是为了提高内测群体提交问题的便携性和及时性，从而快速提交，快速响应，这对内测来说，是非常好用的工具。

UserSupport的接口分成四大类，分别是：用户注册登录接口、用户反馈接口、上传文件接口、Simplex相关接口。这些接口都是restful风格的。

## 1.2 UserSupport接口自动化设计的意义

（解决测试痛点、质量反馈、回归测试、测试策略、重构测试）

接口自动化是将我们已开发完成并上线的功能，进行维护与检测，第一，维护我们现有的功能可用性，第二， 检测新开发的接口，部署之后，是否对于原有功能有影响，造成原有功能不可用。

随着测试技术的发展，接口自动化测试逐渐成为各大公司投入产出比最高的测试技术。介入时间早，执行效率高，稳定性高的优点，让越来越多的公司引入接口自动化测试。

# 2. UserSupport接口自动化的设计

## 2.1 UserSupport接口自动化技术选型

* 开发环境：IntelliJ IDEA 2019.3.4
* 开发语言：Java
* 测试开源框架：Testng+RestAssured
* 项目管理工具：Maven
* 版本控制工具：Git
* 持续集成、自动化构建：Jenkins

1、IDEA是java编程在智能提示、快速编程上具有很大优势，相比Eclipse，更具有优势。

目前自动化常用的语言有java和python，之所以使用java语言，因为Java是一门面向对象编程语言,可以跨平台的，应用十分广泛，速度快的同时又十分稳定。

同时，公司对apk开发人员使用的语言是java,接口自动化使用java更方便和开发人员的沟通和互相协助。

2、Maven工具,只要在pom.xml中引入依赖，就可以快速下载所需要的对应版本的包。

3、Git有助于协助团队协助与分工。

4、Jenkins:TODO

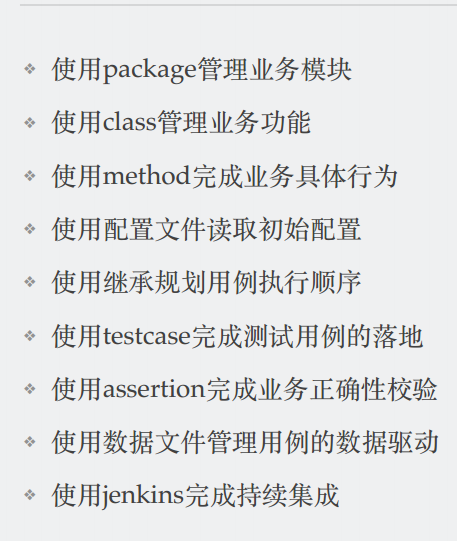
5、测试开源框架使用的是Testng和RestAssured。

TestNG的好处：TODO

RestAssured介绍：TODO

## 2.2 UserSupport接口自动化项目结构

在项目的设计上，为了更好地对代码的管理和维护，遵循下面几个原则。



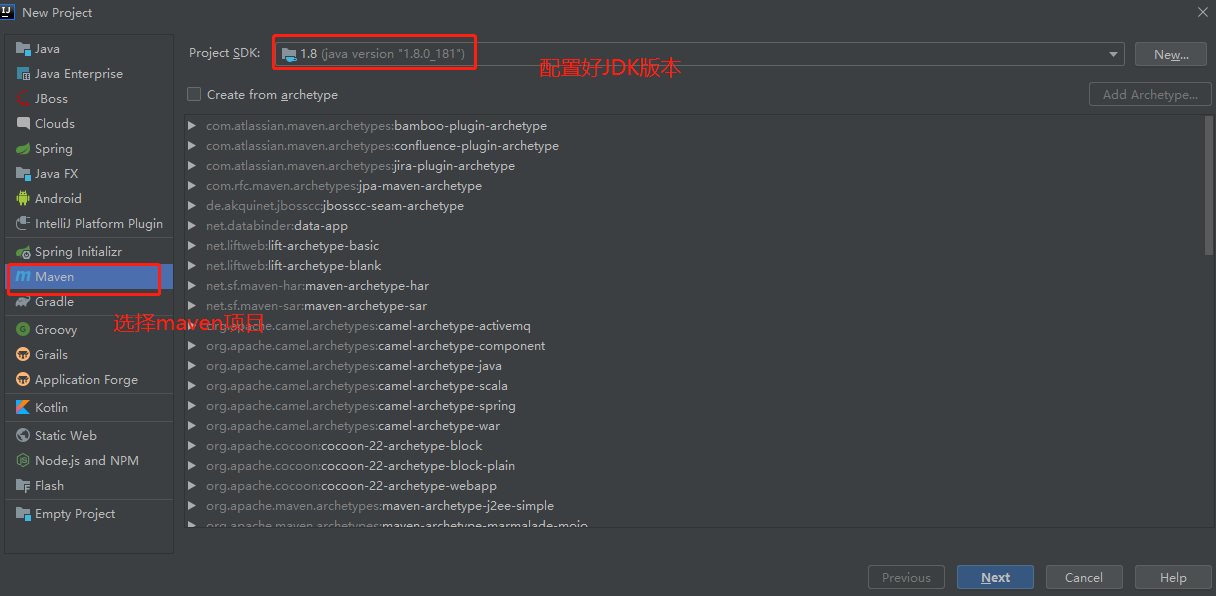
## 2.3 开发准备与环境的配置

JDK1.8的安装、IDEA的安装、Maven的环境变量配置、Git的环境变量配置、Jenkins的安装与配置。

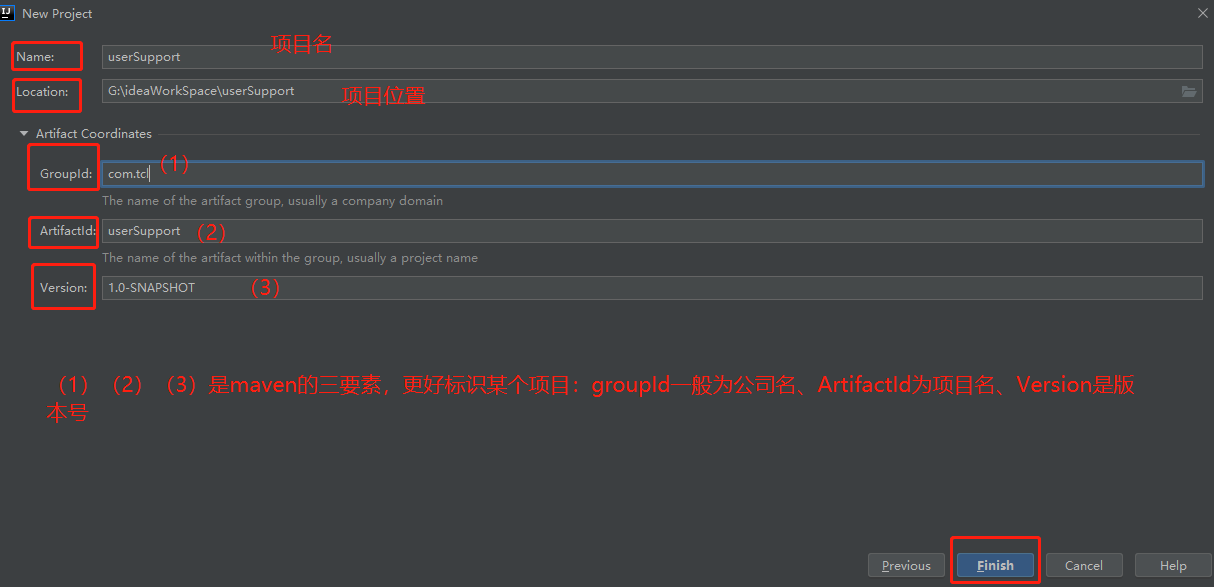
## 2.4 项目开发流程与步骤

### 2.4.1 在IDEA中新建maven项目

File -> New ->Project

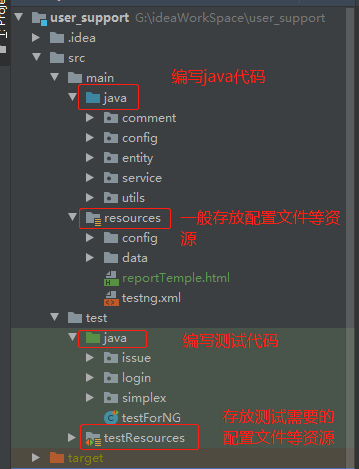


点击Next,填写项目信息



注：groupid和artifactId被统称为“坐标”是为了保证项目唯一性而提出的，如果你要把你项目弄到maven本地仓库去，你想要找到你的项目就必须根据这两个id去查找。

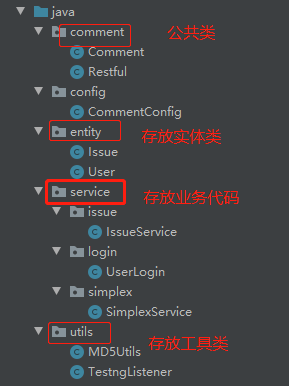
### 2.4.2 创建项目结构



Maven项目的结构一般分成4大部分。

src/main/java :存放不同业务的Java代码。

该java目录下根据不同业务需要创建不同的包，通过不同的包管理不同的业务模块，如下图所示



src/main/resources：存放项目配置文件等资源。

src/test/java:存放测试代码，根据不同接口测试业务需要创建不同的package去管理业务模块，然后具体的class再去实现业务功能。

src/test/resources:存放测试资源。

### 2.4.3 引入相关依赖

Testng和restAssured是开源的框架，要使用他们，必须引入相关的包。

引入方式：通过maven项目中pom.xml文件填写相关依赖导入相关的包。

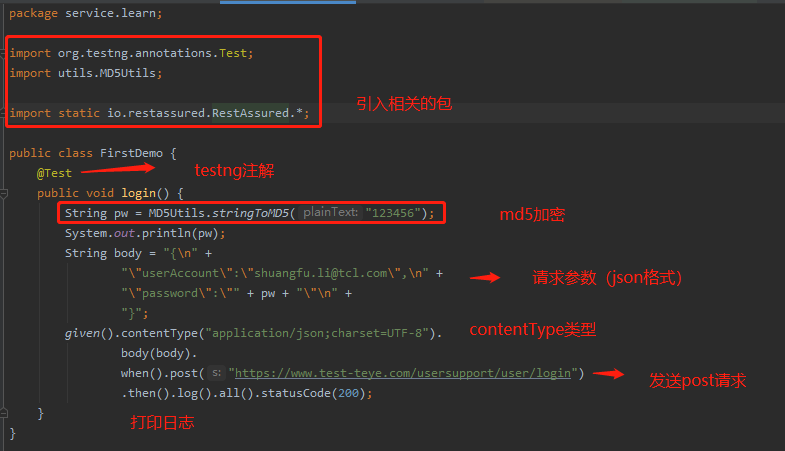
<dependency>  
 <groupId>org.testng</groupId>  
 <artifactId>testng</artifactId>  
 <version>6.10</version>  
 <scope>compile</scope>  
</dependency>

<dependency>  
 <groupId>io.rest-assured</groupId>  
 <artifactId>rest-assured</artifactId>  
 <version>3.2.0</version>  
</dependency>

后续使用到其他包也可以通过这种方法添加依赖。

### 2.4.4 编写第一个简单的接口自动化例子

我们通过userSupport的登录接口进行简单的接口请求编写。



详细分析：上述login方法，是传入请求json格式参数（用户名和密码）请求登录接口，并对响应状态码进行断言。

1）由于密码是使用md5加密的，所以我们先要对密码进行加密处理。

2)变量body的值为请求参数，即用户名和md5加密后的密码，且为JSON格式，。

3)@Test 是testng的注解，标注在测试方法之前，是单元测试的必要。

4)given().contentType(“application/json;charset=UTF-8”).post().then().log().all.statusCode(200)

这里引入了RestAssured的包，调用其api，传入参数与请求类型并发送接口请求，得出响应结果，其中log().all()表示打印所有日志。

这里需要用到RestAssured的知识，请看我总结的文档。（TODO）

### 2.4.5 接口请求方法的封装

封装的意义：

1. 提高代码复用性：从2.4.4的例子可以看出，每次编写接口请求都得去调用RestAssured的api,传入请求参数，根据请求方法请求接口，会造成代码重复冗余，这就需要将其封装起来，提高代码的复用。
2. 优化代码。可以利用配置文件（TODO）

### 2.4.6 Testng测试报告

一、使用testng.xml配置文件管理测试用例的执行顺序。

Testng.xml的配置文件如下：

二、Testng报告的美化：

## 2.5 持续集成与自动构建

### 2.5.1 Jenkins的环境搭建

### 2.5.2 Jenkins的配置以及插件的使用

### 2.5.3 持续集成，自动化构建

# 3. 常见的问题与总结