**自动化测试**

**实施计划**

|  |  |  |  |
| --- | --- | --- | --- |
| 文档状态 | [√] 初稿 | 当前版本 |  |
| [ ] 评审通过 |
| [ ] 修改 |
| [ ] 发布 |
| [ ] 作废 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **修订内容描述** | **作者** | **审核人** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**修 订 记 录**

**目 录**

[一、引言 4](#_Toc519873018)

[1.1. 编写目的 4](#_Toc519873019)

[1.2. 缩写定义 4](#_Toc519873020)

[二、项目概述 4](#_Toc519873021)

[2.1. 概述 4](#_Toc519873023)

[2.2. 范围 4](#_Toc519873024)

[三、实施计划 5](#_Toc519873025)

[3.1. 框架设计 5](#_Toc519873027)

[3.2. CI搭建 6](#_Toc519873028)

[3.3. 开发计划 6](#_Toc519873029)

[3.4. 脚本样例 7](#_Toc519873030)

[3.4.1. UI自动化脚本 7](#_Toc519873031)

[3.4.2. API自动化脚本 8](#_Toc519873032)

# 一、引言

## 编写目的

根据测试部门业务及创新需要，并结合实际需求，拟订网站自动化测试开发项目实施计划。适用读者为：自动化测试开发经理、自动化测试框架设计人员、自动化测试开发人员、软件测试人员以及相关项目组成员。

## 缩写定义

文档中使用到了一些英文缩写，定义如下：

* IDE：集成开发环境（Integrated Development Environment ）
* CI：持续集成（Continuous integration）
* UI：用户界面（User Interface）
* API：应用程序接口（Application Programming Interface）

# 二、项目概述



## 概述

有别于传统桌面应用开发，Web应用因为其迭代周期短，需求变化快的特性，越来越多的Web开发团队在拥抱敏捷。同时随着前端技术JavaScript、Html5等不断被推出和完善，Web应用也变得越来越复杂和绚丽。这都给我们的测试带来了挑战。

如何在有限的资源下，有效地解决目前我们测试团队所面临的挑战呢？最好的方式即是通过工具手段实现。

* 借助selenium + testNG实现UI自动化测试
* 借助httpClient/REST Assured + testNG实现API自动化测试

## 范围

* + 框架设计
  + 脚本开发
    - 主业务流程ui自动化脚本编写
    - 1级测试用例ui自动化脚本编写
    - API测试框架搭建

# 三、实施计划



* 实施人员

负责人：潘永军

实施人员：潘永军

* 实施环境
* OS：win10/win7
* 浏览器：chrome v67
* 开发语言：Java
* IDE工具：Eclipse
* 测试工具：Selenium v3.5.3、autoIt v3
* 版本控制工具：Git、TortoiseGit
* 项目管理工具：Maven v3.39
* 测试框架：Testng 6.14
* CI：Jenkins v2.46
* Servlet容器：tomcat v8.5

## 框架设计

* UI框架内设计：（2018-07-09~2018-07-27）

Maven

Java

TestNG

XML

Auto It

SQL DB

Test Result Report

Selenium WebDriver

Jenkins

1. 概要设计方案
2. 详细代码说明
3. Jenkins部署

* API框架设计：（2018-07-20~2018-08-24）

Maven

Java/Python

TestNG

Excel（API Test File）

SQL DB

Test Result Report

HttpClient/REST Assured

Jenkins

1. 概要设计方案
2. 详细代码说明
3. Jenkins部署

* API测试流程

测试用例  
（Excel）

测试报告（html）

测试数据

API 测试框架（Driver）

## CI搭建

时间：2018-07-16~2018-07-20

参考“网站自动化测试Jenkins部署与迁移.docx”

## 开发计划

时间：（2018-07-02~）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **模块** | **用例数** | **开发时间(周)** | **开发人** | **计划日期** | **完成日期** |
| UI自动化测试框架设计方案编写 |  | 1 | 潘永军. | 2018-07-02 至 2018-07-07 | 2018-07-02 至 2018-07-07 |
| UI自动化测试框架设计 |  | 4 | 潘永军. | 2018-07-02 至 2018-07-27 | 2018-07-02 至 2018-07-20 |
| 业务流程1级UI用例 | 24 | 2 | 潘永军. | 2018-07-02 至 2018-07-13 | 2018-07-02 至 2018-07-13 |
| API自动化测试设计方案编写 |  | 1 | 潘永军. | 2018-07-16 至 2018-07-20 | 2018-07-16 至 2018-07-20 |
| API自动化测试框架设计 |  | 4 | 潘永军. | 2018-07-20 至 2018-08-24 |  |
| 其他1级UI测试用例 | 100 | 4 | 潘永军. | 2018-08-02 至 2018-08-27 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## 脚本样例

### UI自动化脚本

/\*\*

\* case：供应链（减员）-供应链对实时反馈审核通过

\* @author panyongjun

\*/

@Test(alwaysRun=true, groups = {"microseer.scm.employee.workfollow"},

timeOut =MAX\_EXCUTE\_TIME,

dependsOnMethods={"caseOfEmployeeCancelOrderTakeFeedback"})

public void caseOfEmployeeCancelScmApprovePass()throws

FrameworkException{

try{

caseName = "接单中心（减员）-供应链对实时反馈审核通过";

whichCaseIsRun(caseName);

boolean result = false;

//在确保上一条用例执行通过的情况下，才执行

if(caseOfEmployeeCancelOrderTakeFeedbackResult){

//登陆并打开指定模块

loginAndBizModuleOpen(microseerUser, microseerPwd, "供应链

");

secLevelPwdInput(microseerSecPwd);

//雇员管理，实操审核

SCMEmployeeManangePage pageAction = new

SCMEmployeeManangePage(handler);

rootMenuOpen("雇员管理");

queryWithIdOrName("输入姓名、证件号码搜索", userId);

pageAction.employeeManage(userName);

handler.close();

//转入客服中心，社保状态为在保中

handler = newBrowserOpen(url,

TestNGCaseBaseUI.driverBrowserType);

loginAndBizModuleOpen(microseerUser, microseerPwd,

"客服中心");

rootMenuOpen("雇员管理");

queryWithIdOrName("可输入姓名，身份证，电话号", userId);

String statusId = gridColumnIdGet("社保状态");

List<String> statusList =

eleTxtListsGet(By.xpath("//td[@data-columnid='" +

statusId + "']"));

result = strIsSameInList("已停保", statusList);

}

caseOfScmApprovePassResult = result;// 为下调用例记录测试结果

ValidationUtil.verifyTrue(result, caseName, handler);

}catch (Throwable e) {

e.printStackTrace();

throw new FrameworkException(e.getMessage());

}

}

### API自动化脚本

/\*\*

\* case：api测试，豆瓣网书籍检索

\* @author panyongjun

\*/

@Test(groups = {"microseer.scm.api.workflow"}, timeOut =MAX\_EXCUTE\_TIME)//testNG标签

public void apiTest()throws FrameworkException{

try{

caseName = " api测试，豆瓣网书籍检索";//用例名

whichCaseIsRun(caseName);//执行日志分割

//检索关键字，及返回结果记录数

String keySearchTxt = "自动化测试";

String resultCount = "2";

HttpUtils httpApi = new HttpUtils();//实例化HttpUtils

String reponseResult = httpApi.sendGet(

"https://api.douban.com/v2/book/search?q=" +

keySearchTxt + "&start=0&count=" +

resultCount);//以Get的形式获取接口返回值

// 使用返回的字符串直接构造一个JSONObject

JSONObject jsonobj =

JSONObject.fromObject(reponseResult);

//返回books标签下所有的值，并查找标签tags下的name的值

JSONArray jsonArray = jsonobj.getJSONArray("books");

JSONObject row = null;

List<String> tagsNameList = new ArrayList<String>();

for (int i = 0; i < jsonArray.size(); i++) {

row = JSONObject.

fromObject(jsonArray.getJSONObject(i));

JSONArray jsonArray2 = row.getJSONArray("tags");

JSONObject row1 = null;

for(int j=0; j<jsonArray2.size(); j++){

row1 = jsonArray2.getJSONObject(j);

System.out.println(row1.get("name"));

tagsNameList.add(row1.get("name").toString());

}

}

boolean result = strIsIncludedInList(

keySearchTxt, tagsNameList) &&

jsonArray.size() == 2;

ValidationUtil.verifyTrue(result, caseName, handler);

}catch (Throwable e) {

e.printStackTrace();

throw new FrameworkException(e.getMessage());

}

}