

**本科毕业论文（设计）**

**基于loadrunner框架的web应用自动化测试系统的设计与实现**

|  |  |
| --- | --- |
| **学 院：** | **计算机科学与技术学院** |
| **专 业：** | **计算机科学与技术** |
| **班 级：** | **计科152** |
| **学 号：** | **1500170092** |
| **学生姓名：** | **章红梅** |
| **指导教师：** | **黄瑞章** |

2019年 5 月 19 日

**贵州大学本科毕业论文（设计）**

**诚信责任书**

本人郑重声明：本人所呈交的毕业论文（设计），是在导师的指导下独立进行研究所完成。毕业论文（设计）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。

特此声明。

论文（设计）作者签名：

日 期：

目 录：

目录

**[基于Loadruner的自动化测试的设计与实现](#_Toc6511_WPSOffice_Level1)** **[1](#_Toc6511_WPSOffice_Level1)**

[摘要](#_Toc508_WPSOffice_Level2) [1](#_Toc508_WPSOffice_Level2)

**[Design and Implementation of Automated Test Based on Loadruner](#_Toc508_WPSOffice_Level1)** **[2](#_Toc508_WPSOffice_Level1)**

[Abstract](#_Toc11481_WPSOffice_Level2) [2](#_Toc11481_WPSOffice_Level2)

[第1章 绪论](#_Toc170_WPSOffice_Level2) [3](#_Toc170_WPSOffice_Level2)

**[第二章 相关技术介绍](#_Toc11481_WPSOffice_Level1)** **[5](#_Toc11481_WPSOffice_Level1)**

[2.1 Loadrunner工具介绍](#_Toc12266_WPSOffice_Level2) [5](#_Toc12266_WPSOffice_Level2)

[2.2自动化接口功能测试简介](#_Toc27199_WPSOffice_Level2) [5](#_Toc27199_WPSOffice_Level2)

[2.3自动化性能测试简介](#_Toc19062_WPSOffice_Level2) [6](#_Toc19062_WPSOffice_Level2)

[2.4自动化测试流程简介](#_Toc1500_WPSOffice_Level2) [6](#_Toc1500_WPSOffice_Level2)

[2.5本章小结](#_Toc25206_WPSOffice_Level2) [6](#_Toc25206_WPSOffice_Level2)

**[第三章 系统分析](#_Toc170_WPSOffice_Level1)** **[7](#_Toc170_WPSOffice_Level1)**

[3.1测试项目背景](#_Toc30582_WPSOffice_Level2) [7](#_Toc30582_WPSOffice_Level2)

[3.2需求分析](#_Toc29313_WPSOffice_Level2) [7](#_Toc29313_WPSOffice_Level2)

[3.3 小结](#_Toc25578_WPSOffice_Level2) [8](#_Toc25578_WPSOffice_Level2)

**[第四章 详细设计](#_Toc12266_WPSOffice_Level1)** **[9](#_Toc12266_WPSOffice_Level1)**

[4.1 接口测试范围](#_Toc30207_WPSOffice_Level2) [9](#_Toc30207_WPSOffice_Level2)

[4.2 测试流程](#_Toc8960_WPSOffice_Level2) [9](#_Toc8960_WPSOffice_Level2)

[4.3测试用例设计](#_Toc28588_WPSOffice_Level2) [9](#_Toc28588_WPSOffice_Level2)

[4.4 自动化接口功能测试详细设计](#_Toc22095_WPSOffice_Level2) [11](#_Toc22095_WPSOffice_Level2)

**[第五章 测试过程](#_Toc27199_WPSOffice_Level1)** **[13](#_Toc27199_WPSOffice_Level1)**

[5.1搭建测试环境](#_Toc9574_WPSOffice_Level2) [13](#_Toc9574_WPSOffice_Level2)

[5.2测试脚本准备](#_Toc13393_WPSOffice_Level2) [14](#_Toc13393_WPSOffice_Level2)

[5.3脚本运行结果](#_Toc17259_WPSOffice_Level2) [16](#_Toc17259_WPSOffice_Level2)

[5.4性能测试测试过程](#_Toc4411_WPSOffice_Level2) [24](#_Toc4411_WPSOffice_Level2)

[5.5本章小结](#_Toc27237_WPSOffice_Level2) [29](#_Toc27237_WPSOffice_Level2)

**[第六章 测试结果](#_Toc19062_WPSOffice_Level1)** **[29](#_Toc19062_WPSOffice_Level1)**

[6.1测试结果报告](#_Toc15173_WPSOffice_Level2) [29](#_Toc15173_WPSOffice_Level2)

[6.2测试结果分析](#_Toc1786_WPSOffice_Level2) [31](#_Toc1786_WPSOffice_Level2)

[6.3代码质量分析](#_Toc11915_WPSOffice_Level2) [32](#_Toc11915_WPSOffice_Level2)

[6.4本章小结](#_Toc1722_WPSOffice_Level2) [32](#_Toc1722_WPSOffice_Level2)

**[第七章 总结与展望](#_Toc1500_WPSOffice_Level1)** **[32](#_Toc1500_WPSOffice_Level1)**

[参考文献](#_Toc1204_WPSOffice_Level2) [32](#_Toc1204_WPSOffice_Level2)

[致谢](#_Toc23969_WPSOffice_Level2) [33](#_Toc23969_WPSOffice_Level2)

基于Loadruner的自动化测试的设计与实现

## 摘要

随着web端技术的发展，客户对web端应用的质量也要求越来越高,对测试的需求也越来也大；手工测试在某些做某些测试任务时已经逐渐凸显其局限性。例如，当做重复操作测试时，手工测试会耗费大量的人力；当后台系统接口不稳定时，测试人员无法或很难必现，当有性能测试需求时，手工测试很难实现。因此，自动化测试在某种情况下是很有必要的。

针对以上情况，本文介绍了基于Loadrunner的web端应用的自动化测试，全文介绍了整个系统的实现过程，主要包括需求分析、详细设计、系统实现。

系统实现着重介绍了脚本设计及场景设计。

本系统使用面向[开源](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90/20720669" \t "https://baike.baidu.com/item/_blank)及私有[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053" \t "https://baike.baidu.com/item/_blank)项目的托管平台GitHub 来作为代码版本变更的工具，可以很好的管理程序代码及文档的每个历史。

本系统实现了基于Loadruner的接口功能测试和接口性能测试。Loadrunner是一款主流测试工具，同时也能进行简单的功能测试。系统具体实现了用户登录、文件夹的增删该、稿件的增删改、文章关键词提取、文章摘要提取、文章情感分析、稿件查询、文章添加评论、微信库稿件获取这些接口的功能测试和性能测试。为系统后期的冒烟测试节省了大量的人力和时间，对系统性能有了一个整体了解，帮助开发更好的优化系统性能。

关键词：web端应用,自动化测试，功能测试，性能测试

Design and Implementation of Automated Test Based on Loadruner

# Abstract

With the development of web-side technology, customers are increasingly demanding the quality of web-side applications, and the demand for testing is also increasing. Manual testing has gradually highlighted its limitations when doing certain testing tasks. For example, when doing repeated operation tests, manual testing can consume a lot of manpower; when the background system interface is unstable, the tester can't or can't do it. When there is performance testing demand, manual testing is difficult to achieve. Therefore, we can see that automated testing is necessary in some cases.

In view of the above situation, this paper introduces the automated test of the web-based application based on Loadrunner. The whole process introduces the implementation process of the whole system, including requirements analysis, detailed design and system implementation.The system implementation focuses on script design and scene design.

The system implements interface function test and interface performance test based on Loadruner. Loadrunner is a mainstream test tool that also performs simple functional tests. The system specifically implements user login, folder addition and deletion, manuscript addition and deletion, article keyword extraction, article summary extraction, article sentiment analysis, manuscript query, article add comment, WeChat library manuscript to obtain functional test and performance test of these interfaces. It saves a lot of manpower and time for the system's late smoke test, and has a holistic understanding of system performance to help develop better system performance.

Keywords:web-side,applications,automated testing, functional testing, performance testing

# 绪论

## 1.1背景及意义

自动化测试是在互联网蓬勃发展取而产生的测试方式。自动化测试通过模拟用户的真实行为，可以减少手工测试的重复性工作，并测试系统性能。对于开发周期较长的产品适合采用自动化测试。

Loadrunner是惠普公司的一款自动化测试工具，深受测试人员的青睐，Loadrunner做自动化测试采用了ip欺骗的原理更加接近真实用户操作场景。

手工测试能够快速地发现缺陷，而自动化测试和手工测试并不矛盾，自动化测试并不是所有的项目都适用，它适用于性能测试及重复性操作的测试案例，后期做回归测试、冒烟测试。回归测试主要是看系统有没有解决以前老的bug,并且有没有引入新的bug；而冒烟测试主要监测系统的基本功能是否能够正常工作。

因此在产品的上线时间长，版本迭代过于频繁的情况下 ，将会耗费大量的精力在频繁的测试上。当然，随着用户量越来越大，当有一定并发量时，系统容易出错。那么采用自动化测试不仅可以简单地测试应用的基本功能，同时也能测试系统性能，找出系统瓶颈。

## 1.2发展状况

随着web端技术的发展壮大，手工测试已不能完全满足各种web端应用测试需求。首先，若应用后期冒烟测试还进行手工测试，肯定会耗费人力资源，显然自动化测试更适合。宁外，当应用有性能需求时，要求并发大量操作，手动测试模拟并发显得不现实，而且自动化性能测试的优势是可通过设置虚拟用户的方式，对被测应用系统进行并非加压测试。

现在，自动化测试已经在软件行业被广泛应用。未来，自动化测试技术将会愈加成熟，使我们的测试很大程度上趋于自动化。

1.3系统概况

本系统测试的对象为web应用全媒体资源库，全媒体资源库是一个存储稿件的库系统。

本系统对目标测试系统进行自动化功能接口测试和自动化接口性能测试。首先，先对每一个接口模块进行功能测试，并记录接口的响应时间和接口功能测试结果，然后让接口按照一定的顺序做系统测试。接下来对系统进行系统测试，首先用controller打开测试脚本，合理设置运行场景，对系统稳定性测试及负载测试，得出系统性能测试报告并对报告进行分析和预估。

1.4文本组织结构

本文共有7章，第一章主要介绍了系统的背景及意义，以及系统地一个发展概况；第二章介绍了系统的相关技术；第三章分析了系统的系统的测试目标及业务流程。第四章介绍了系统的详细设计，包括测试范围及测试用例；第五章详细介绍了系统的具体实现；第六章则是对测试结果的分析以及对系统的评估。最后是对本文的总结。

# 相关技术介绍

## 2.1 Loadrunner工具介绍

Loadrunner是一款自动化测试工具，一般用于做接口测试，Loadrunner是现阶段一款主流的测试工具，同时也能进行简单的功能测试。

Loadrunner由三大组件构成，分别是Virtual User Generator、Load Generator、Controller和Analysis，其中Vuser相当于是一个编译器，通常用于录制脚本或者编写脚本以及回放脚本；在用Vuser编写脚本的过程中，设计到很多的技术，例如插入事物，插入检查点，关联函数等，而回放脚本主要是一个检查脚本的过程。Load Generator用于产生并发负载，通常被安装在压力机上。Controller用于场景设计，通过设置虚拟用户在压力机上运行脚本的方式，对目标测试系统进行逐步加压，达到检测被测目标系统性能的目的。Analysis是一个分析工具，分析的对象是场景运行的结果，生成直观的图表，进而分析系统的性能。

为什么要使用Loadrunner？Loadrunner有如下优点：

1. Loadrunner对于新手来说是比较容易并且快速地上手，学习起来也较为容易。
2. Loadrunner采用了代理地模式，运用ip欺骗的原理，能够真实地模拟用户行为，更加接近真实的用户行为。
3. Loadrunner的事物是衡量性能的一个指标，能够帮助我们快速地定位系统瓶颈。
4. Loadrunner不仅可以用于做简单的功能测试，更为强大的功能还可以进一步做性能检测

2.2自动化接口功能测试简介

接口测试是指根据被测应用系统接口文档进而对被测应用接口进行测试，首先要得到接口地址，其次要弄清楚接口的请求参数，对接口发起请求，然后要看接口的响应结果是否正确，若接口的响应结果和预期结果一致，则验证通过，反之不通过。最后生成接口测试结果报告。

2.3自动化性能测试简介

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测，性能测试包括狭义的性能测试、负载测试，压力测试，稳定性测试，并发测试。性能测试的目的是评估被测应用的整体性能，定位应用瓶颈和预估。

2.4自动化测试流程简介

自动化测试只有按照一定的流程进行才能将自动化测试做好，首先我们需要熟悉系统需求和系统，设计测试用例，搭建测试环境，实施测试，测试分析，问题反馈与解决，最后生成测试报告。测试流程如下：

熟悉系统需求

设计测试用例

实时监测

测试分析

实施测试

性能测试报告

功能测试报告

2.5本章小结

本章主要介绍了Loadrunner的三大组件以及三大组件各自的作用，使用Loadrunner的优点，以及自动化功能、性能测试简介和自动化测试流程简介。

# 第三章 系统分析

## 3.1测试项目背景

本系统的测试项目是web端应用全媒体资源库，全媒体资源库是一个数据存储网站，主要面向媒体单位用户。实现了的业务包括登陆、文件夹的增删该、稿件的增删改、文章关键词提取、文章摘要提取、文章情感分析、稿件查询、文章添加评论、微信库稿件获取。

该项目开发周期长，上线时间长，在之前通过手工方式做过功能测试，但从未做过任何自动化测试及性能测试。

系统虽然做过功能测试，但系统在后期的优化过程中常常会引入新的bug，导致原本可用的基本功能不可用，因此，系统需要不断作冒烟测试。宁一方面，一些bug在前期的开发过程中，由于任务的优先级有所不同，低优先级的缺陷会被放到后期解决，那么系统在后期还需要做回归测试。自动化测试适用于冒烟测试和回归测试，因此我们需要对部分模块做回归测试和冒烟测试。最后系统存在不稳定的现象，我们需要系统性能进行性能测试，提出解决方案，为出现严重性问题做好预防。

3.2需求分析

需求分析是测试的第一步，只有明确需求，才能完成后续测试工作，以下为需求描述及详细信息。

|  |  |
| --- | --- |
| 需求描述 | 详细信息 |
| 应用信息 | 全媒体资源库是个基于B/S的web端应用，通信方式采用https协议，实现库存储的基本功能。应用现在已经上线了，使用的用户约50人。 |
| 测试环境 | 测试环境已经按照开发的要求进行配置。测试数据在需编写脚本的过程中新建。测试的用户开发已注册到数据库。 |
| 测试目标 | 需要进行简单的自动化接口功能测试，对于接口功能测试，除了登陆接口采用正向测试和反向测试，其余 接口采用只进行正向测试。宁还需要进行性能测试，测试的并发用户为20-30个用户进行并发,并作长时间的稳定测试,时间设定为10分钟。 |
| 测试历史 | 该项目手动做过功能测试，系统的功能已全部可用，但从未做过性能测试。 |
| 测试时间 | 2019/05开始 |
| 应用接口文档 | 已有应用的接口文档作为参考。 |

## 3.3 小结

本章主要介绍了被测应用的背景，介绍了被测项目的用户和被测应用实现的功能以及其测试情况。宁外介绍了测试过程的测试需求。

# 第四章 详细设计

4.1 接口测试范围

测试目标系统：web端应用全媒体资源库

测试接口：用户登陆平台、更新用户信息、用户登陆全媒体应用系统、文件夹的新建、修改文件名、删除文件夹、获取文章关键词、获取文章摘要、文章情感分析、稿件增删改、稿件查询、为稿件添加评论、获取微信库稿件。

4.2 测试流程

站在系统的测试流程共有如下两步：

第一步：对系统的接口进行功能自动化测试。生成测试报告。

第二步:在功能测试的基础上加进一步根据需求做性能测试，生成测试报告，提出解决方案。

4.3测试用例设计

4.3.1登陆接口测试测试用例

（1）登陆接口正向测试

步骤：输入正确的手机号，正确的密码

预期：能够登陆成功，接口返回状态码200，并返回用户的企业信息。

（2）登陆接口用户名错误反向测试

步骤：输入错误的账户名，输入密码点击登陆

预期：提示消息用户不存在。

（3）登陆接口密码错误反向测试

步骤：输入正确的手机号，然后输入错误的密码，点击登陆。

预期：提示消息用户名与密码不匹配。

4.4.2选择企业登陆后更新用户信息接口测试

步骤：输入正确的企业信息，正确的usercode和token

预期：接口返回状态码200，返回用户所在的企业名称。

4.4.3登陆全媒体资源库接口测试

步骤：输入正确的企业信息，appid,正确的usercode和token

预期：接口返回状态码200，返回全媒体资源库的链接

4.4.4新增文件夹接口测试

步骤：请求新增文件夹接口，输入库id，文件夹名字，appid,正确的usercode和token

预期：接口返回状态码200，返回新增文件夹的基本基本信息并在全媒体应用系统中生成对应文件夹。

4.4.5修改文件夹名字接口测试

步骤：请求修改文件夹名字接口，输入库id，需要修改的文件夹id，修改后名字,正确的usercode和token

预期：接口返回状态码200，返回修改后文件夹的基本信息，并在全媒体资源库系统中成功地修改了文件夹的名字。

4.4.6删除文件夹接口测试

步骤：请求删除文件夹接口，输入库id，需要删除的文件夹id，正确的usercode和token

预期：接口返回状态码200，返回修改后文件夹的基本信息，并在全媒体资源库系统中成功地删除了该文件夹

4.4.7新增稿件接口测试

步骤：请求新增稿件接口，输入新增稿件所在的库id，文件夹id，稿件内容，正确的usercode和token 。

预期：接口返回状态码200，返回新建文件夹的基本信息，并相应地在全媒体资源库系统中成功地创建了一篇稿件。

4.4.8获取文章关键词接口测试

步骤：请求获取文章关键词接口，输入全文内容，正确的usercode和token。

预期：接口返回状态码200，返回获取得到的关键词信息。

4.4.9获取文章摘要接口测试

步骤：请求获取文章摘要接口，输入全文内容，正确的usercode和token。

预期：接口返回状态码200，返回根据全文获取得到的摘要信息。

4.4.10对文章进行情感分析接口测试

步骤：请求对文章进行情感分析接口，输入全文内容，正确的usercode和token。

预期：接口返回状态码200，返回根据全文获取得到的全文的情感指数。

4.4.10 修改文件夹接口测试

步骤：请求请求对文章进行情感分析接口,传入文件夹id，修改后的用户名，正确的usercode和token。

预期：口返回状态码200，修改文件夹名字称成功

4.4.10 修改文件夹接口测试

步骤：请求请求对文章进行情感分析接口,传入文件夹id，修改后的用户名，正确的usercode和token。

预期：口返回状态码200，修改文件夹名字称成功

4.4 自动化接口功能测试详细设计

4.4.1测试结果表设计

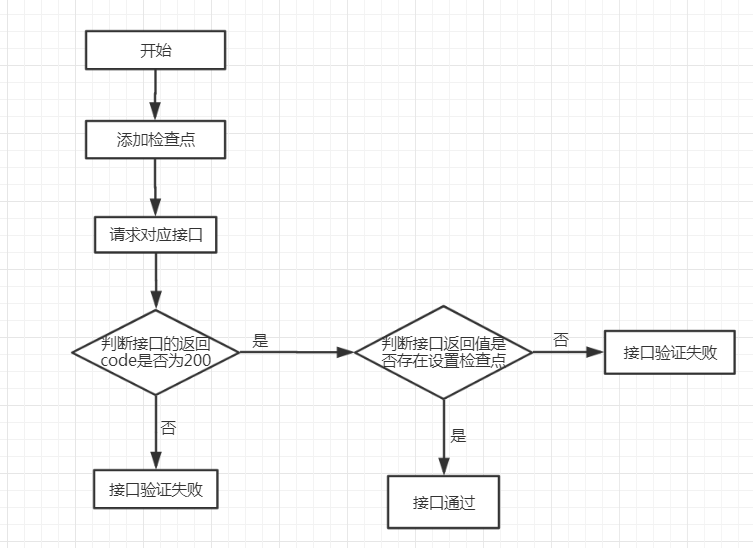
测试结果的表设计如下所示，其中描述信息表示对接口测试的描述；预期结果表示期望系统返回的结果；而Y/N则表示对应一个接口测试是否通过，当实际结果与预期结果相一致时，结果为通过，不一致时，结果为失败。

|  |  |  |  |
| --- | --- | --- | --- |
| 描述信息 | 预期结果 | 实际结果 | Y/N |
| 登陆 | 注册成功 | 注册失败 | 失败 |
| 登陆 | 登陆成功 | 登陆成功 | 通过 |

图4.5.1测试结果表设计

4.4.2 接口功能测试模块流程设计

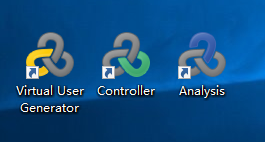
首先,在请求见接口之前注册检查点,之后请求对应接口,然后根据接口的API文档,当接口的返回值code为200时，判断为请求接口成功，若code为200，则判断检查点是否存在，若存在，则接口验证通过



# 第五章 测试过程

## 5.1搭建测试环境

运行环境为Loadrunner12.3，12.3版本可支持50个虚拟用户的并发量，因测试需求只要求并发用户数没有超过50个，所以测试环境满足需求。安装过程全部都是下一步即可安装成功，然后就会在在我们的电脑上出现三个组件。点开Virtual User Generator 即可开始编写脚本。



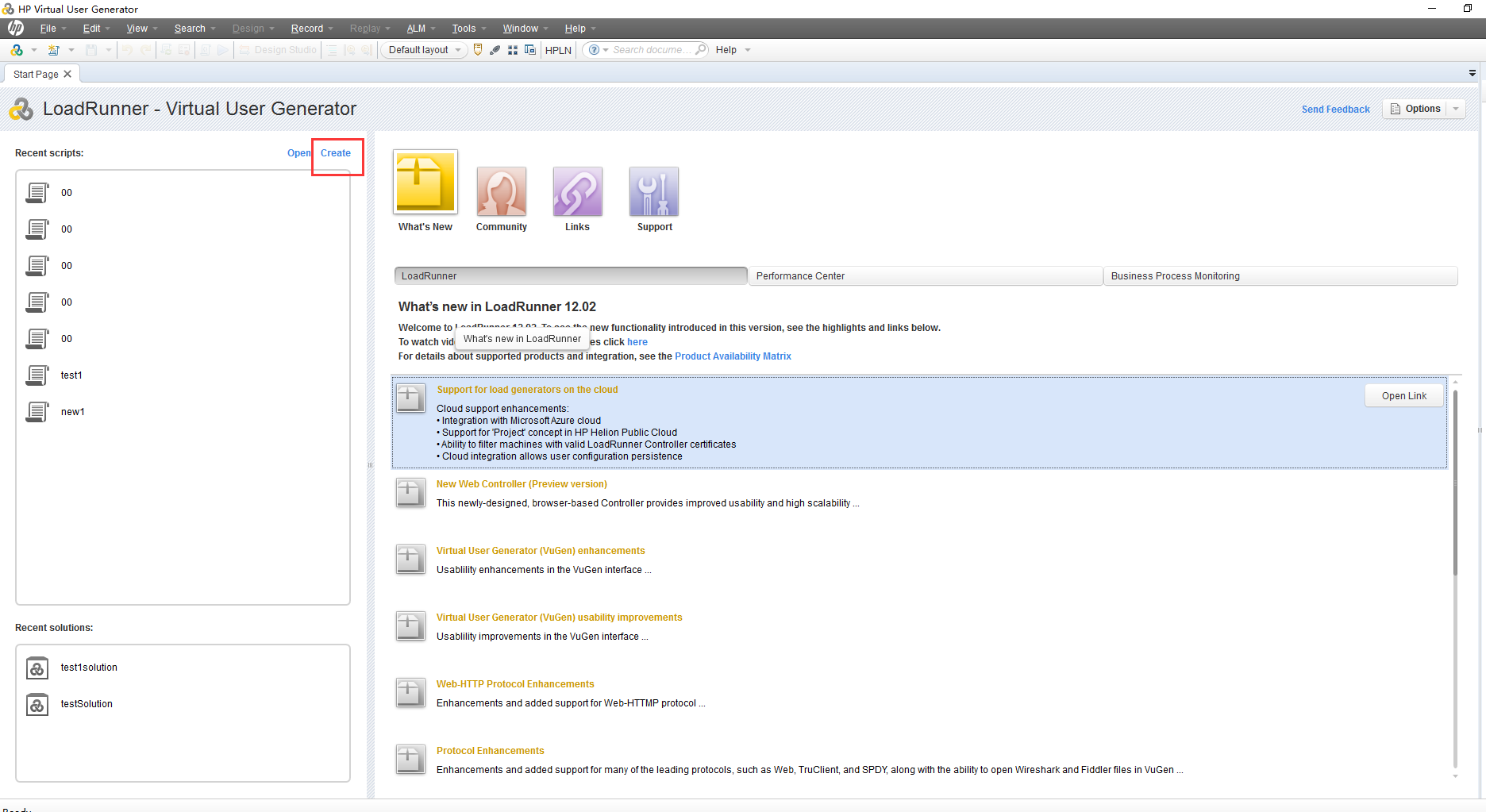
## 5.2测试脚本准备

5.2.1脚本定义

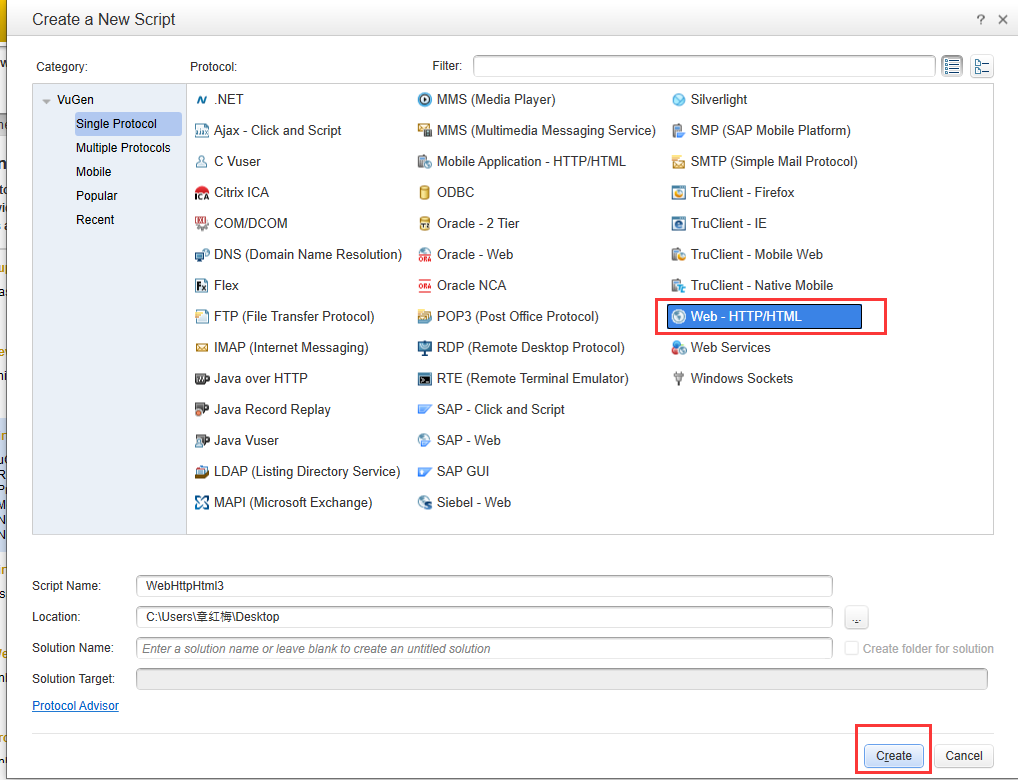
Testing script（测试脚本），一般指的是一个特定测试的一系列指令，这些指令可以被自动化测试工具执行[1]。测试脚本可通过自动化测试工具自动生成，也可通过手动方式编写脚本。自动生成的脚本相当于就是对操作翻译，例如Loadrunner测试脚本脚本可通过录制的方式生成脚本。

5.2.2脚本编写

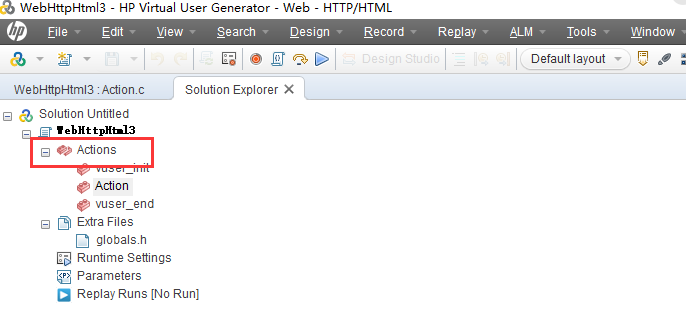
编写脚本是执行自动化测试的第一步，本测试采用手动编写脚本的方式。首先点开Virtual User Generator，进入如下界面，点击图中的“**+**”号创建脚本



因为测试系统为web端应用，所以选择http协议，如下图所示，然后再点create创建。

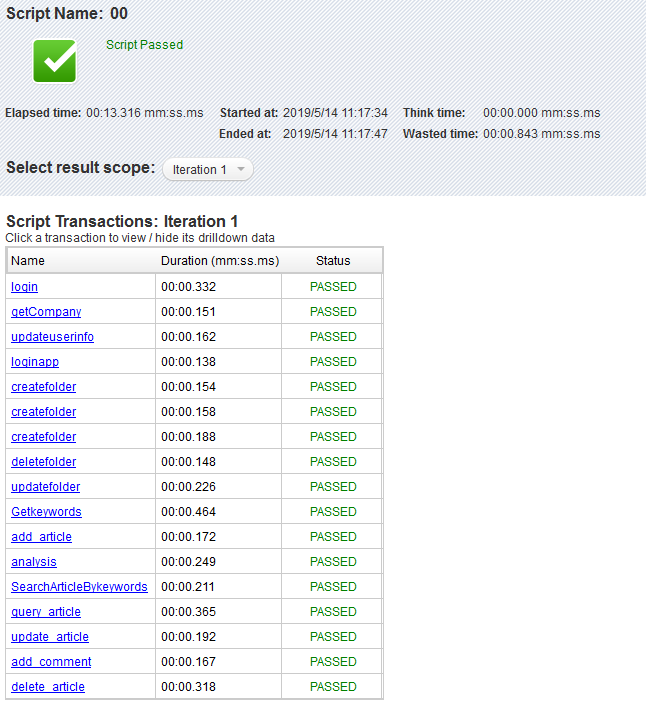


接下来接进入了编写脚本界面，如下图，一个solution就创建成功了，编写脚本在Actions展开目录的三个文件中编写。Vuser\_init编写初始化操作的脚本，如登录；Action编写主要操作逻辑的脚本；而vuser\_end主要编写退出的脚本，如退出登陆。



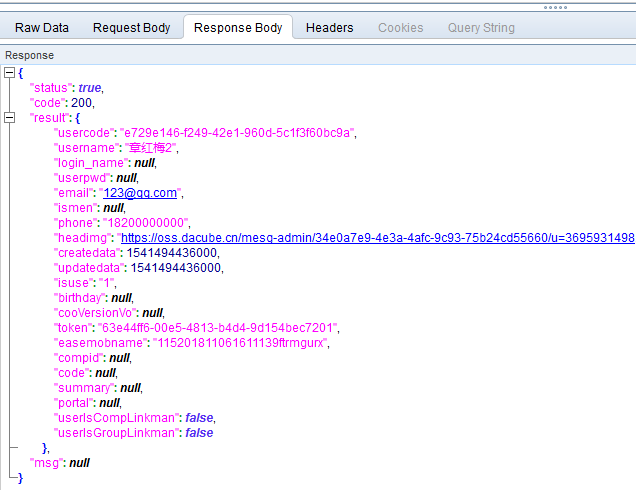
## 5.3脚本运行结果

所有接口的整体运行结果如下，共15个接口，17个事物。所有事物在运行过成中没有出错。

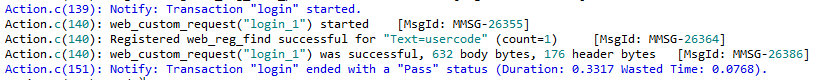


5.1.1登录接口自动化测试

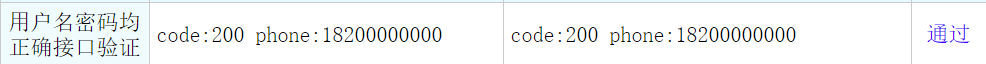
1、登陆接口的测试请求体为正确的用户名和正确的密码，接口成功返回了用户的信息，状态码、用户名、用户id、用户token，状态码为200表示登陆成功。并且系统成功地生成了该接口的测试结果报告。接口返回信息以及测试结果如下图，用时332ms。



用户名密码均正确时登陆接口response

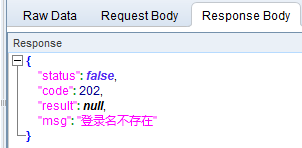


用户名密码均正确请求登陆接口响应时间

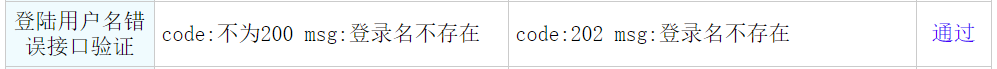


用户名密码均正确登陆接口响测试结果

2、登陆接口的测试请求体为正确的错误用户名和密码，接口成功响应，并提示“登录名不存在”，测试结果通过。接口返回信息以及测试结果如下图。

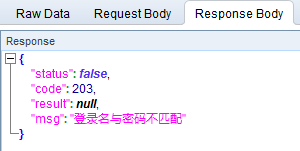


登陆接口用户名错误响应结果



登陆接口用户名错误测试结果

3、登陆接口的测试请求体为正确的正确用户名和错误密码，接口成功响应，并提示“密码错误”测试结果通过。接口返回信息以及测试结果如下图。



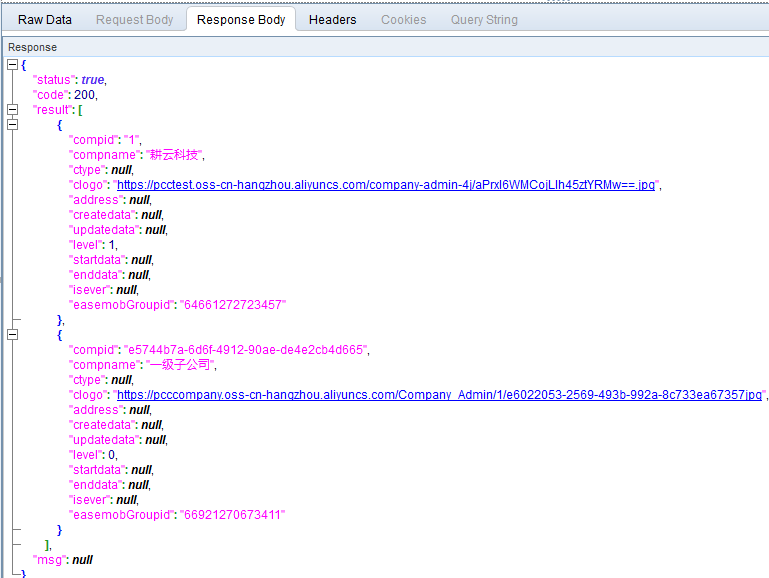
登陆接口用户名密码不匹配响应结果



登陆接口用户名密码不匹配测试结果

5.1.2获取用户企业信息接口测试

获取用户企业信息接口是根据用户的Usercode 获取用户企业信息，接口返回的code为200，并返回用户所在企业信息，接口验证通过。运行结果如下图：



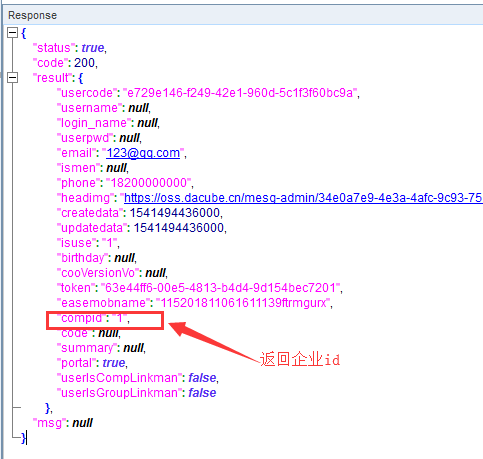
获取用户企业信息接口响应结果



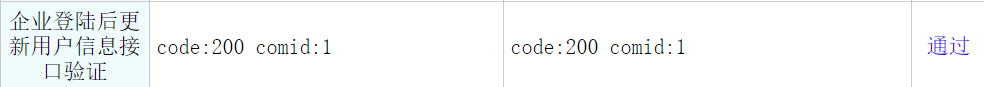
获取用户企业信息接口测试结果

5.1.3登陆企业接口验证

登陆企业接口的功能使用户登陆到对应企业，请求接口时需要传入企业id，接口返回的code为200，并返回用户登陆的企业id，接口验证通过。运行结果如下图



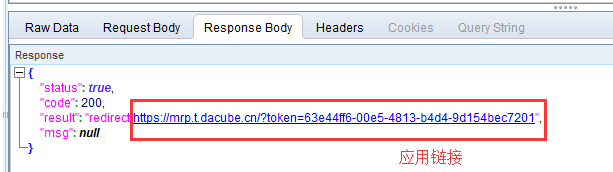
登陆企业接口响应结果

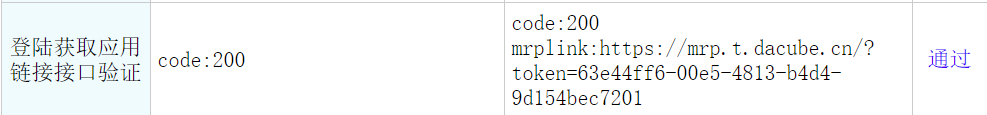


登陆企业接口接口测试结果

5.1.5登陆web应用接口测试

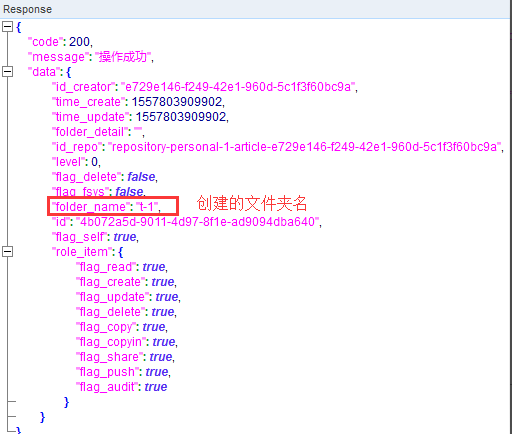
该接口的作用是打开应用，请求接口时需要传入应用id，接口会返回该应用的链接，返回的code为200，为了进一步验证我们将返回浏览器中打开能够成功打开对应的应用界面，接口验证通过。运行结果如下图



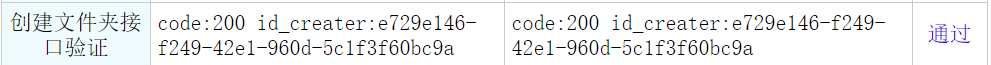


5.15文件夹的增删改

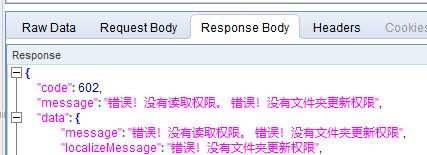
1. 文件夹新增接口返回消息为操作成功，code为200，对应的在被测系统中创建一个文件夹，验证通过，结果如下：

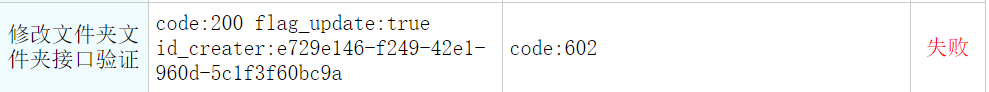






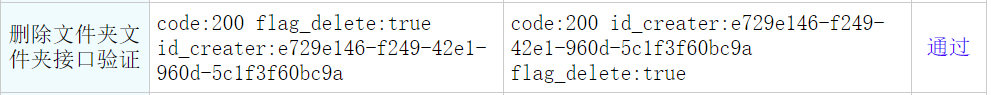
1. 修改文件夹名字接口返回消息为没有权限，code为602，进一步分析该文件夹的所属库为个人库，个人库没有权限限制，不符合逻辑，验证失败





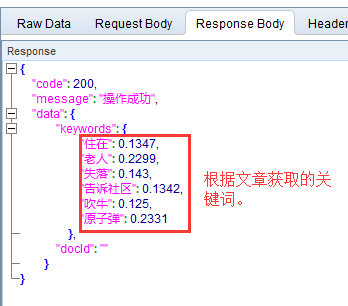
1. 删除文件夹名接口返回消息为操作成功，code为200，在被测系统中查看修改对应文件夹名字成功，验证通过。结果如下：

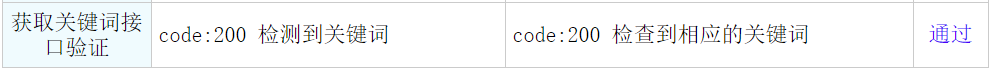




5.1.5获取文章关键词接口测试

获取文章关键词，响应结果及测试结果如下图，测试结果为通过。





5.1.6文章情感分析接口测试

获取文章关键词，响应结果及测试结果如下图，测试结果为通过。

5.1.7 文章增删改接口测试

获取文章关键词，响应结果及测试结果如下图，测试结果为通过。

5.1.8文章查询接口测试

获取文章关键词，响应结果及测试结果如下图，测试结果为通过。

5.1.9文章添加评论接口测试

获取文章关键词，响应结果及测试结果如下图，测试结果为通过。

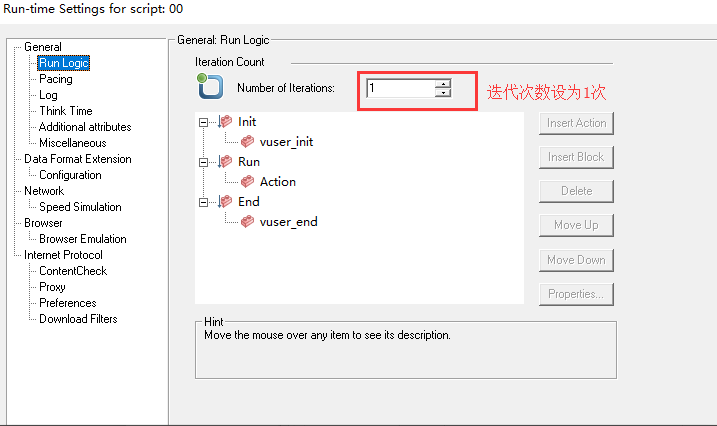
5.1.10 获取微信库稿件接口测试

获取文章关键词，响应结果及测试结果如下图，测试结果为通过。

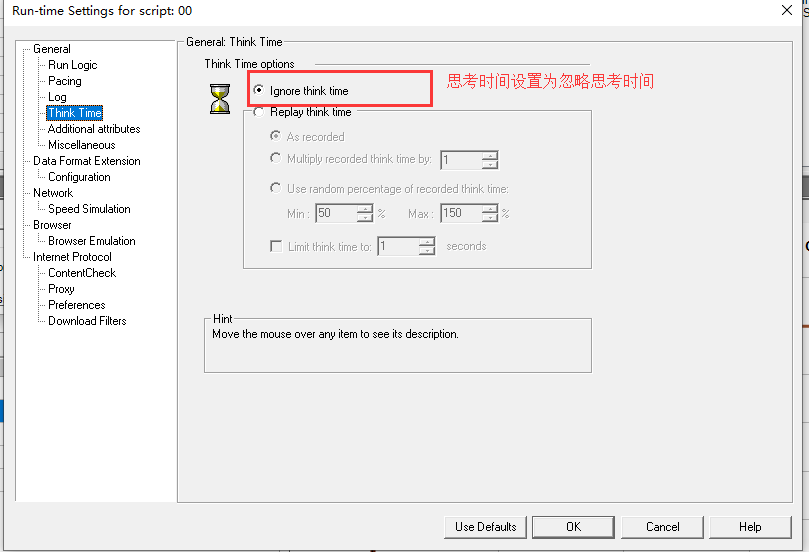
### 5.4性能测试测试过程

#### 5.4.1运行时设置

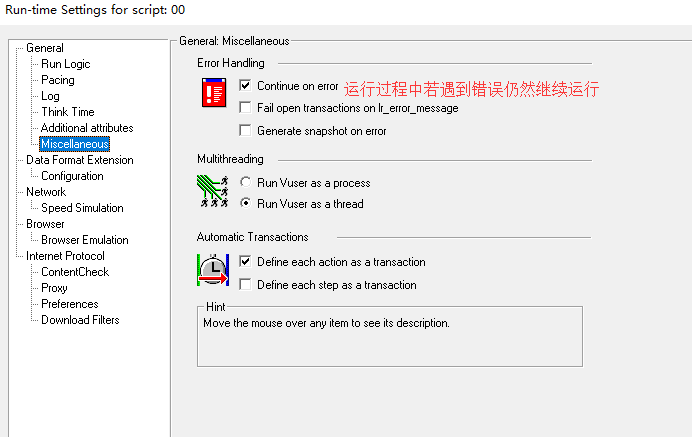
运行时设置主要设置四个地方，一是迭代次数设为1次，为了给系统加压，二是思考时间设置为忽略思考时间，第三，若运行过程中遇到错误设置为仍然选择继续。



迭代次数设置



思考时间设置图



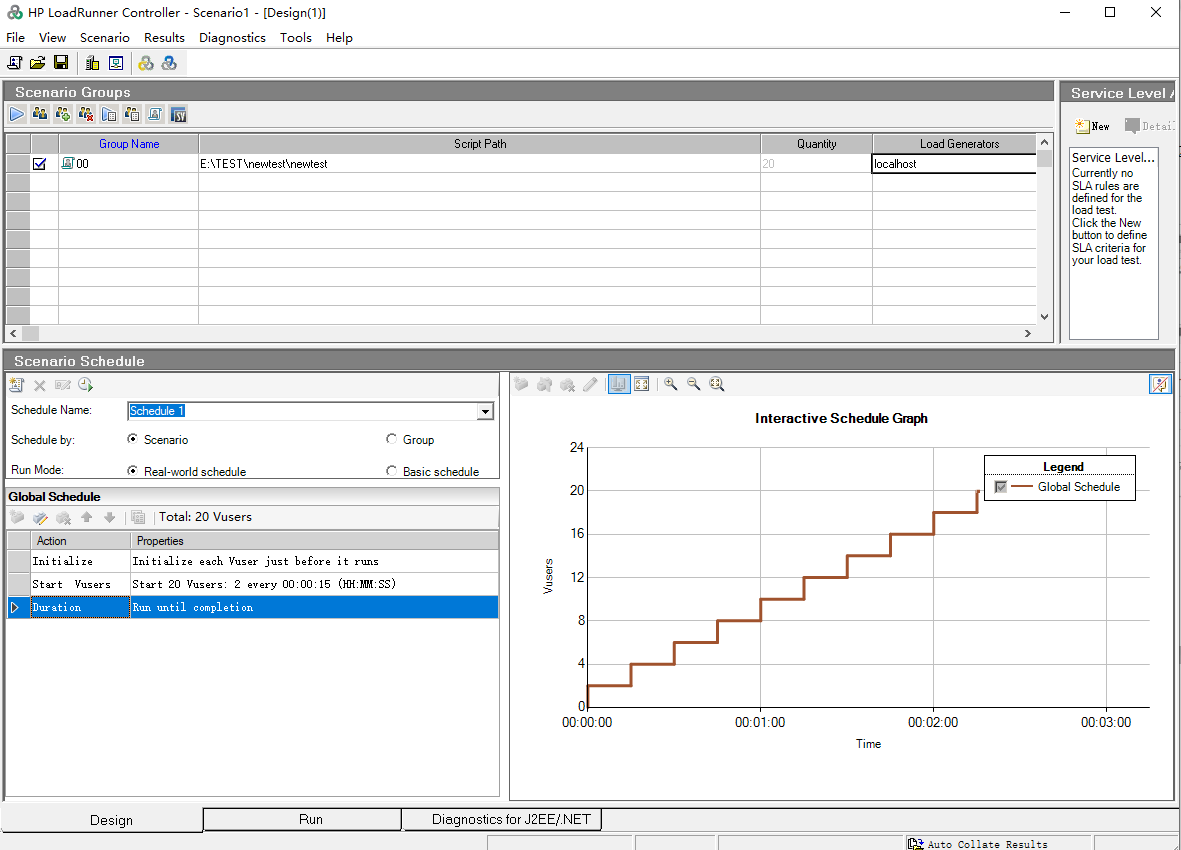
#### **5.4.2场景设计**

**5.4.2.1 场景定义**

场景是组织脚本的方式，即压力测试的策略【1】。Loadrunner中的场景是通过conroller组件生成的，场景中定义了并发用户数、用户的增长方式、虚拟用户启动方式，虚拟用户的迭代次数、思考时间等。 这些可更改设置可模拟出与生产环境基本一致的用户行为【1】。

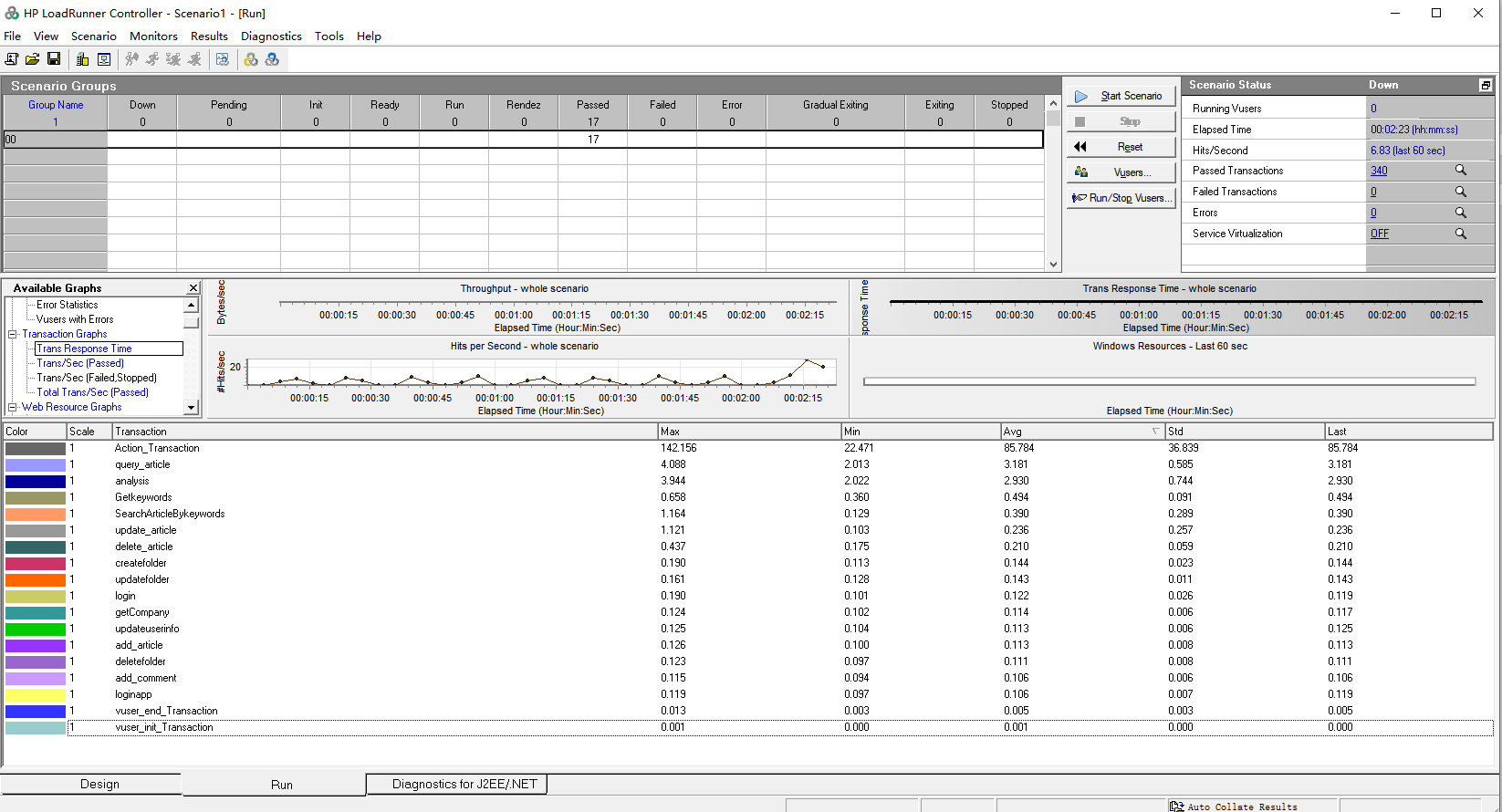
**5.4.2.2场景设计**

场景设置采用手动场景的设置方式，根据需求第一次设置20个虚拟用户，第二次设置30个，第三次设置40个。结合运行时设置，每个用户只迭代一次，场景采用慢增长的设计方式，每15秒启动两个用户，达到设置点后开始运行。运行时间设置为直到所有的虚拟用户运行结束。下图为第一次设置的场景，之后两次只用改变虚拟用户的数目，其它设置保持不变。

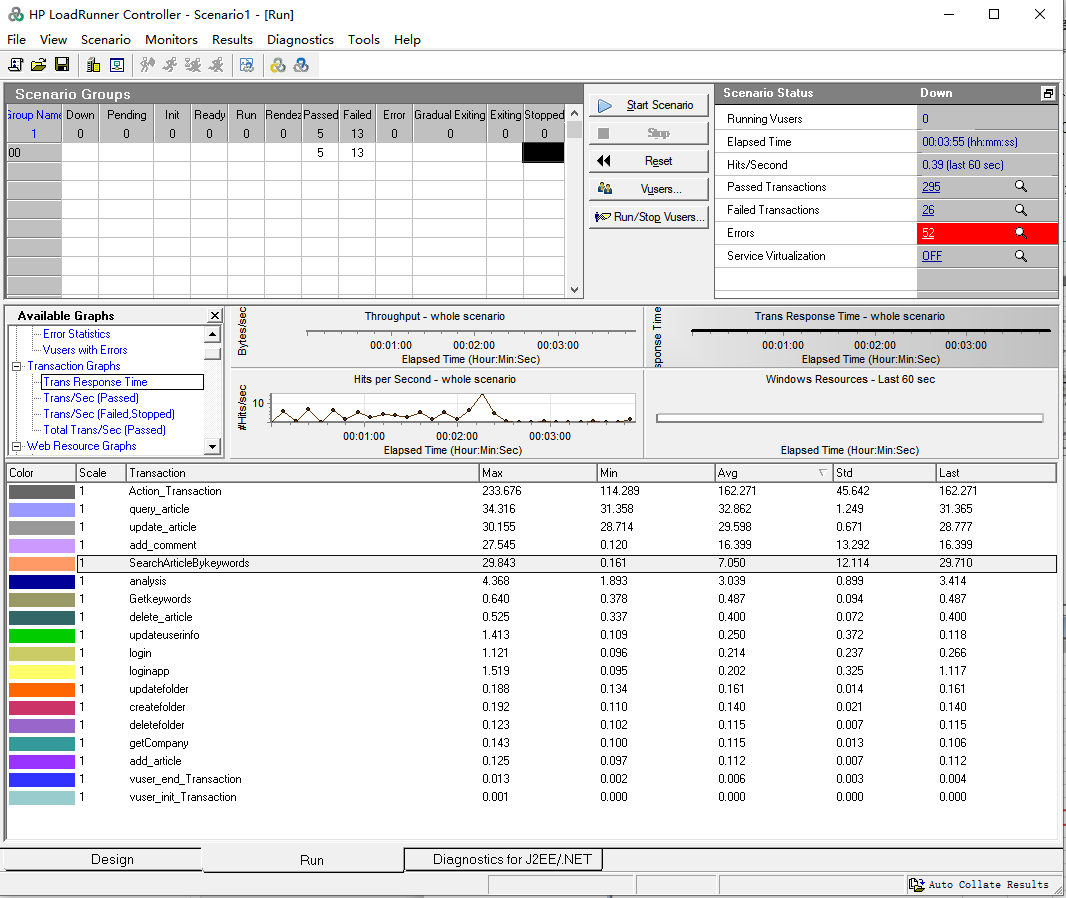


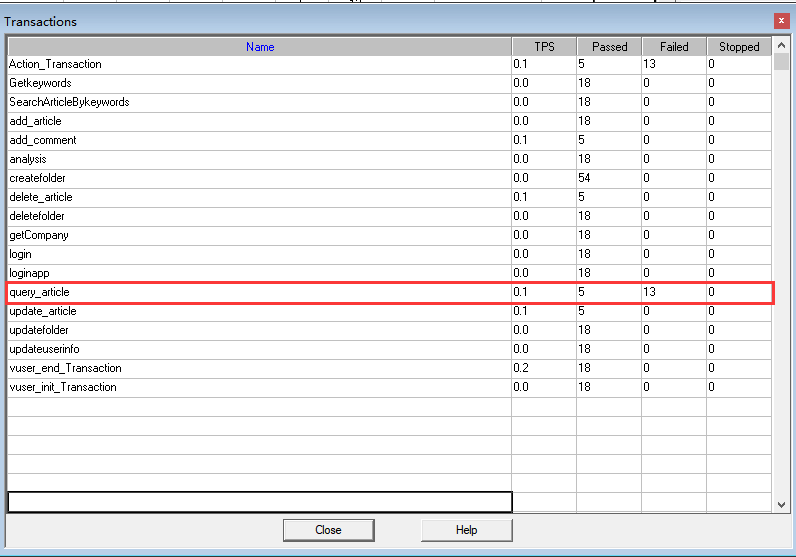
#### **5.4.3运行结果**

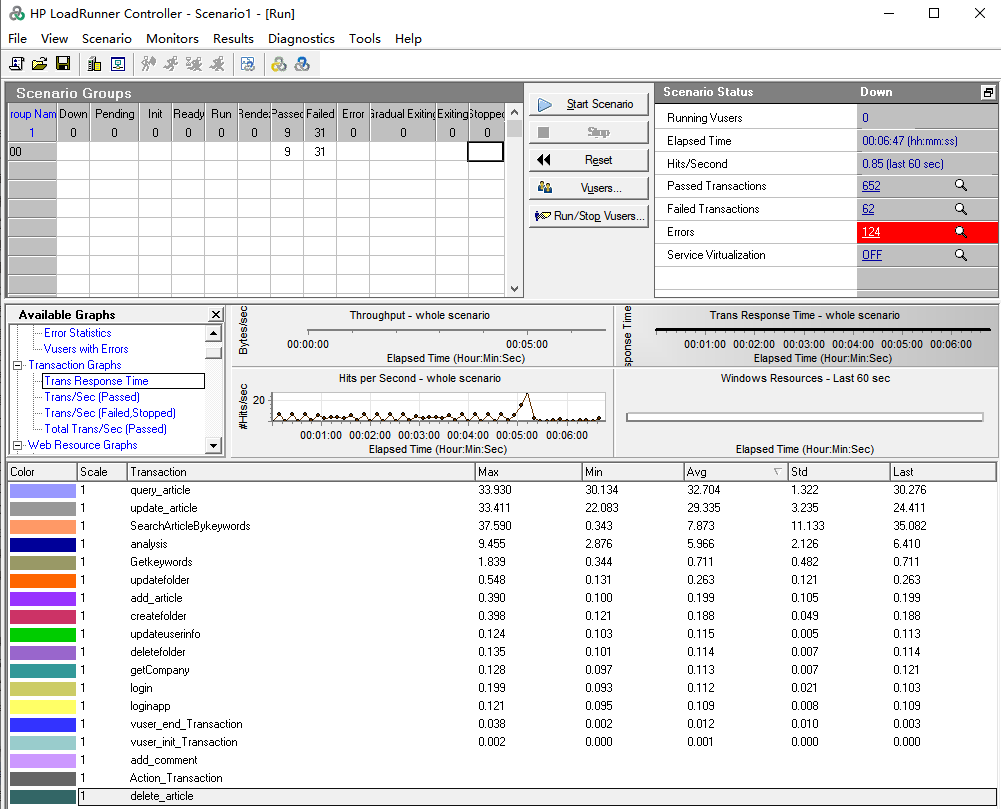
1、15个虚拟用户的运行结果如下，场景运行时间2分，错误数为0，失败的事物数为0，最大事物响应时间为3.18s，该事务为获取微信库稿件事物（图中query-article）。

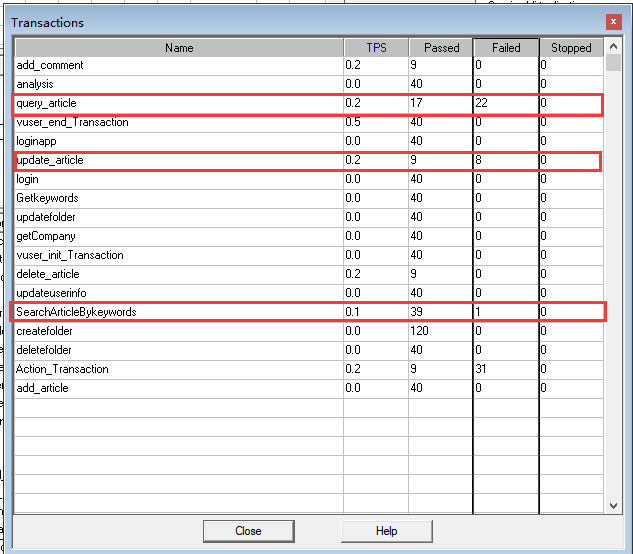


2、18个虚拟用户的运行结果如下，场景运行时间3分26秒，响应时间用时最较长的为获取微信库稿件事物（图中query-article）和关键词查询事物（图中query-article）和情感分 析（图中analysis）,分别用时33s、20s、4s，其余事物响应时间均在3秒内。失败事物数为20个，分别为action事物和query-article事物，而query-article事物是被写在action事物中的，因此失败事物数共20-10=10个，10个失败事物均为query-article事物。





1. 查询事物（图中query-article）和情感分 析（图中analysis）,分别用时33s、20s、4s，其余事物响应时间均在3秒内。失败事物数为20个，分别为action事物和query-article事物，而query-article事物是被写在action事物中的，因此失败事物数共20-10=10个，10个失败事物均为query-article事物。
2. 



#### **5.5本章小结**

本章讲了自动化的测试过程，从环境搭建到到脚本编写到功能测试再到性能测试过程都做了详细的说明。

# 第六章 测试结果

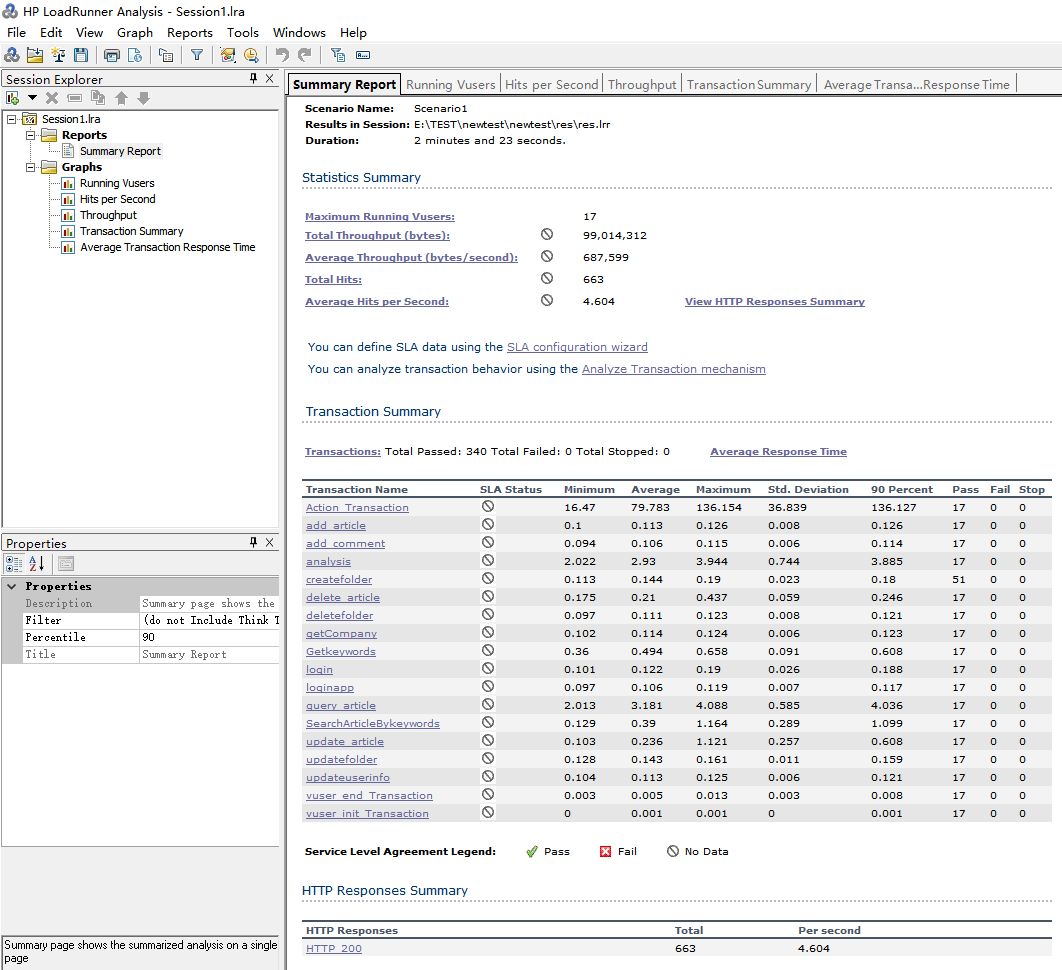
6.1测试结果报告

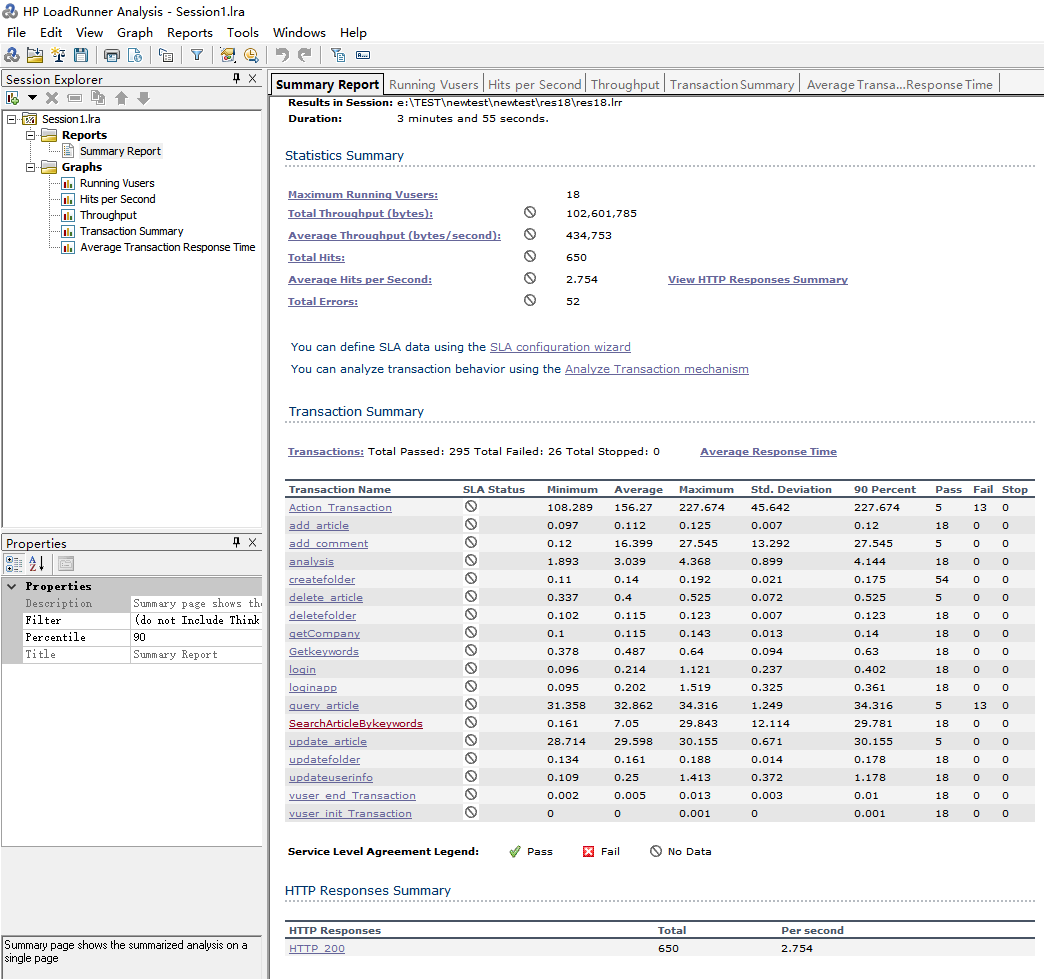
6.1.1 功能测试报告

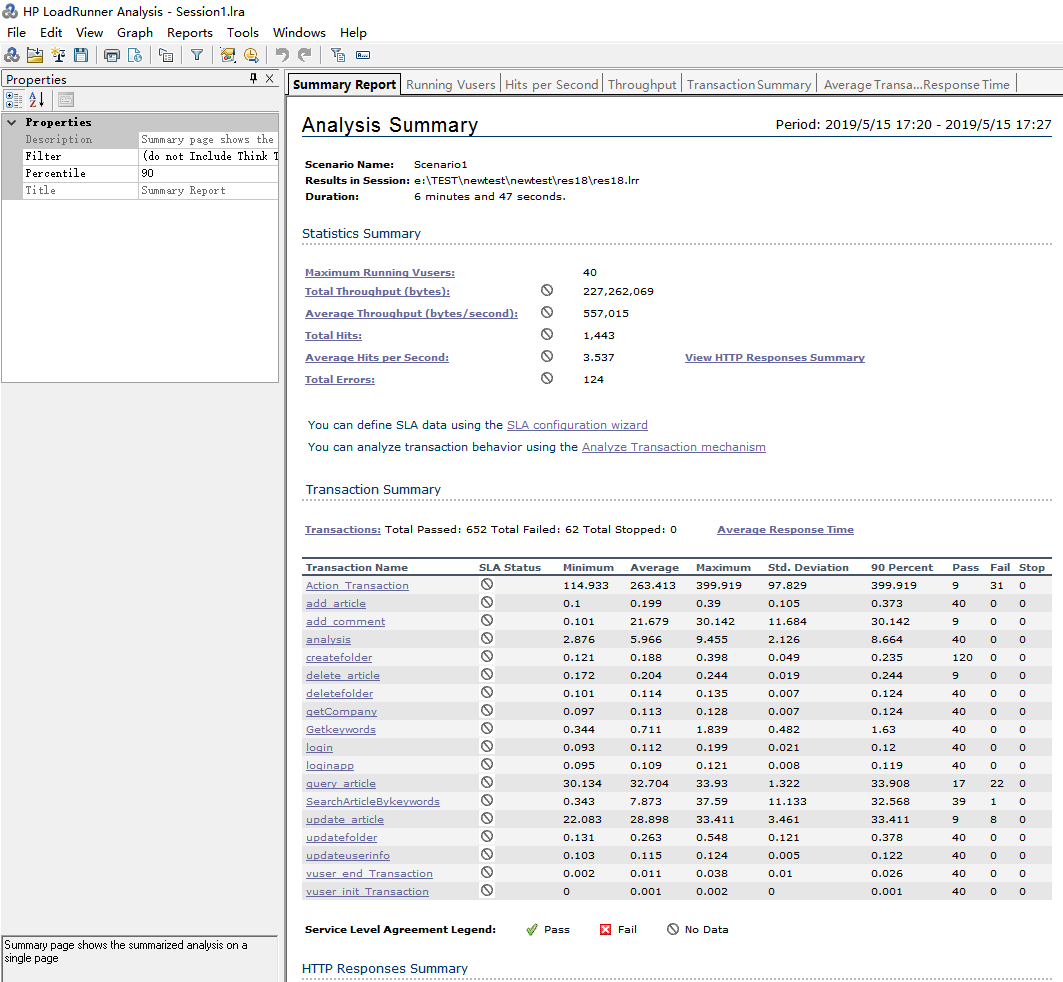
功能测试完成了测试范围中全部接口的功能测试，共17接口测试用例，其16个接口测试用例通过，1个接口测试不通过。该测试报告由系统自动生成，当预期结果与实际结果相一致时，测试通过，否则，测试不通过，测试结果如下。

6.1.2性能测试报告

在Analysis中打开controller的运行结果，Analysis自动生成摘要报告和各种图表。摘要报告可导出为html报告，方便查看。摘要报告的结果如下：







6.2测试结果分析

由功能测试结果可知，应用虽然只有一个接口验证失败，其余接口均验证过。但失败的却是严重性的bug，与开发确认后，确实存在这样的问题。由此判断系统的功能实现处于良好状态。

6.3代码质量分析

脚本可以采用录制的方式生成代码，但这种方式会产生很多的垃圾代码。而该测试系统的所有代码并没有采用录制的方式，而是全部手动按照逻辑写的，所以全部代码都是有效而不多余的。对于代码逻辑一样和重复使用的代码进行了封装，代码复用率高。

6.4本章小结

本章主要介绍了功能测试结果报告，以及性能测试结果报告，通过对功能测试报告分析，得出测试结果；通过对性能测试结果报告分析，得出应用的性能情况和对应用的预估。

# 总结与展望

web端自动化测试系统是一个基于Loadrunner对被测系统做了简单功能测试和性能测试的测试系统。该系统是基于windows的，在整个过程中，自己学到了很多东西，感悟深刻，总结为以下几点。

1. 测试需要学习的东西很多，需要较强的业务逻辑和不断学习的精神。

# 参考文献

[1] 刘艺.Java程序设计教程[M].北京机械工业出版社2006，2.

[2] 张锦煌. 基于SSM整合框架开发的项目管理系统[J]. 电脑知识与技术，2015年，11期.

[3] 黄秀娟. 基于SpringMVC + Hibernate + Extjs框架的Web系统应用研究[J]. 电脑知识与技术，2015年，16期：3.

[4] 贺军. 基于XML应用的MyBatis技术[J]. 火力与指挥控制，2013年，z1期：3.

[5] 张婧. 下一代WEB开发标准HTML5的研究[J]. 希赛网，2011年,9期.

[6] 刘海舒. 浅谈jQuery的应用[J]. 技术前沿，2010年，02期.

[7] 尹婷. 基于jQuery框架的AJAX网站设计模式的研究[J]，理工科学,2010年，03期.

[8] 薛峰. 基于Spring MVC框架的Web研究与应用[J]，计算机与信息工程,2012年，03期.

[9] 戚琦. 基于敏捷方法的轻量级J2EE架构的应用[J]，计算机系统应用,2007年，02期.

[10] 吴小青. JSP+TOMCAT+MYSQL开源软件整合配置初探[J]，齐齐哈尔大学学报(自然科学版),2012年，04期.

[11] 周建军. 如何分析信息系统的非功能性需求[Z]. 希赛网，2007年.

[12] 王天志. 扩展实体联系图的实现及应用[J]. 云南师范大学学报(自然科学版)，2003年，z1期：4.

[13] 王善发. Java编程语言构造方法研究[J]. 保山学院学报，2012年，2期：43.

[14] 辛凤艳. 软件中出现的缺陷的原因和分析[J]. 河北省科学院学报，2009年，2期：28.

[15] 刘壮飞. 浅谈使用Sonar进行代码质量分析[J]. China New，2015年，20期.

# 致谢

首先要感谢我的指导老师黄瑞章老师，黄老师在我的整个做毕设过程中监督我一步一步的完成毕设，从最开始选题的把关，再到系统实现过程中的问题分析和指导，直到最后完成毕业论文。黄老师用她的智慧、耐心及负责的态度帮助我最终完成了我的毕业设计。在此，我对黄老师表示深深的谢意。

最后，我还要感谢我的室友、同学、老师、师兄的帮助。感谢他们的帮助让我学到了很多知识技能，不断地成长进步，最后才会使我在做毕设时有充分的预备知识，顺利完成毕业设计。

由于本人的技术水平及表达能力有限，论文中若有不当之处还请专家和读者不吝赐教，我将虚心接受，并加以改善。