南 阳 理 工 学 院

本科生毕业设计(论文)

学院(系)： 软件学院

专 业： 软件工程

学 生： 张三

指导教师： 李四

完成日期 2015 年 04 月

南阳理工学院本科生毕业设计（论文）

XXXX系统的设计与实现

Design and Implementation of the  
XXXX System

总 计：毕业设计(论文) 5页

表 格： 1个

图 片： 1个

南 阳 理 工 学 院 本 科 毕 业 设 计(论文)

XXXX系统的设计与实现

Design and Implementation of the  
XXXX System

学 院(系)： 软件学院

专 业： 软件工程

学 生 姓 名： 张三

学 号： 12345678

指导教师(职称)： 李四 副教授

评 阅 教 师： 王五

完 成 日 期： 2015年04月02日

南阳理工学院

Nanyang Institute of Technology

XXXX系统的设计与实现

软件工程 张三

1. 自动化测试指的是无需过多人工干预，从而实现测试的目的，减少人为误差。目前互联网行业的测试能力远远跟不上开发的进度和技术深度，往往是开发工程师完成开发后，测试工程师才开始挨个测试，效率低下。针对这样的问题，我们需要开发一种高效的便捷的测试方式。本系统后端使用Python的Flask Web框架，Python有利于提高开发速度和提高代码可读性。前端使用Vue和element-UI实现，大大降低了前端开发的难度，降低开发时间成本。数据库使用MySQL数据库，开源免费的，降低开发成本，简单好用。本系统不但能实现自动化测试，还在可视化方面提高用户交互。通过简单的操作就能实现复杂的测试。本系统支持批量执行测试，大大方便了大型后端系统的测试流程，提高研发效率。本系统有着操作简单、方便快捷的优点，代码量小，容易进行第三方开发特点。
2. MySql;Python;自动化测试

Design and Implementation of the  
XXXX System

Software Engineering Major Zhang San

1. Automated testing means that there is no need for too much manual intervention to achieve the purpose of testing and reduce human error. At present, the testing capability of the Internet industry can not keep up with the progress and technical depth of development. Often, after the development engineer completes the development, the test engineer starts testing one by one, which is inefficient. In response to such problems, we need to develop an efficient and convenient testing method. The back end of this system uses Python's Flask Web framework. Python is conducive to improving development speed and improving code readability. The front end is implemented using Vue and element-UI, which greatly reduces the difficulty of front-end development and reduces the cost of development time. The database uses MySQL database, which is open source and free, which reduces development costs and is easy to use. This system not only enables automated testing, but also improves user interaction in terms of visualization. Complex operations can be achieved with simple operations. This system supports batch execution testing, which greatly facilitates the testing process of large back-end systems and improves research and development efficiency. This system has the advantages of simple operation, convenience and quickness, small amount of code, and easy third-party development.
2. MySql;Python; [automated](javascript:;) [testing](javascript:;)

目 录

[1 绪论 1](#_Toc36042624)

[1.1 背景 1](#_Toc36042625)

[1.2 国内外研究现状分析 2](#_Toc36042626)

[1.3 可行性分析 2](#_Toc36042627)

[1.3.1 经济可行性 2](#_Toc36042628)

[1.3.2 技术可行性 3](#_Toc36042629)

[1.3.3 运行可行性 3](#_Toc36042630)

[1.4 实验内容 3](#_Toc36042631)

[1.5 实验目标 3](#_Toc36042632)

[1.6 论文整体结构 3](#_Toc36042633)

[2 相关工具及技术介绍 4](#_Toc36042634)

[2.1 开发环境 4](#_Toc36042635)

[2.2 编程语言 4](#_Toc36042636)

[2.2.1 Python 4](#_Toc36042637)

[2.2.2 JavaScript 5](#_Toc36042638)

[2.3 数据库存储 5](#_Toc36042639)

[2.4 前端展示技术 6](#_Toc36042640)

[2.5 Web技术 6](#_Toc36042641)

[3 测试方案及测试用例设计 7](#_Toc36042642)

[3.1 测试方案 7](#_Toc36042643)

[3.2 测试用例设计 7](#_Toc36042644)

[3.2.1 系统整体设计 7](#_Toc36042645)

[3.2.2 数据库结构设计 7](#_Toc36042646)

[3.2.3 日志模块设计 7](#_Toc36042647)

[3.2.4 定时任务设计 7](#_Toc36042648)

[4 测试实现 7](#_Toc36042649)

[4.1 API实现 7](#_Toc36042650)

[4.2 UI实现 7](#_Toc36042651)

[4.3 系统UI展示 8](#_Toc36042652)

[5 测试结论 8](#_Toc36042653)

[结束语 9](#_Toc36042654)

[参考文献 10](#_Toc36042655)

[附录 12](#_Toc36042656)

[致谢 13](#_Toc36042657)

# 绪论

随着互联网行业发展，公司系统Web API越来越多。一个小的销售系统都有近100个API，每次版本迭代测试都是严峻的挑战，需要专门的测试团队进行测试。人力测试的时间成本和人力成本很高，而且人为存在失误，导致用户体验差的情况，投诉率居高不下。

本系统很好的解决了这个问题，系统只需要在每次系统API开发完成后，录入测试用例，即可一键批量测试API。完美解决时间成本和人力成本的问题，避免人为操作失误的问题。

## 背景

API测试用于确定API是否针对各种可行请求返回正确的响应（以预期的格式），对诸如失败和意外等极端输入之类的极端情况做出正确反应，在可接受的时间内交付响应，以及安全地应对潜在的安全攻击[1][4]。服务虚拟化与API测试结合使用，以隔离被测服务，并通过模拟无法进行测试的API /服务来扩展测试环境访问[6]。

API测试通常包括测试REST API或SOAP Web服务以及通过HTTP，HTTPS，JMS和MQ发送的JSON或XML 消息有效负载[2][7]。它还可以包括消息格式，例如SWIFT，FIX，EDI和类似的固定长度格式，CSV，ISO 8583和通过传输协议（例如TCP / IP，ISO 8583）发送的 协议缓冲区，MQTT，FIX，RMI，SMTP，TIBCO集合点和FIX[8][9]。

API测试通常涉及以下实践：

1、单元测试：测试单个操作的功能。

2、功能测试：测试更广泛场景的功能，通常使用单元测试作为端到端测试的构建块。包括测试用例定义，执行，验证和回归测试。

3、负载测试：常通过重用功能测试用例来验证负载下的功能和性能。

4、运行时错误检测：监视应用程序的自动或手动测试的执行，以暴露诸如争用条件，异常和资源泄漏之类的问题。

5、安全测试：包括渗透测试和模糊测试以及验证身份验证，加密和访问控制。

6、Web UI测试：作为端到端集成测试的一部分执行，该测试还涵盖API，使团队可以在较大事务的上下文中验证GUI项目。

7、互操作性测试：（SOAP only）检查是否符合Web服务互操作性概要文件。

8、WS-\*符合性测试：（SOAP only）检查是否符合WS- \*标准，例如WS-Addressing，WS-Discovery，WS-Federation，WS-Policy，WS-Security和WS-Trust。

9、渗透测试：测试计算机系统，网络或Web应用程序，以发现攻击者可以利用的漏洞。

10、模糊测试：强制将大量纯随机数据（有时称为"noise"或"fuzz"）输入系统，以尝试强制崩溃，溢出或其他负面行为。这样做是为了在绝对极限下测试API，并且在某种程度上充当"worst case scenario"。

## 国内外研究现状分析

近年来Web服务领域发生着翻天覆地的变化，继传统的XML-RPC风格的Web Service之后，一种新的风格REST被应用于Web Service。

REST是Representational State Transfer(表述性状态转移)的缩写，它最初是Roy Thomas Fielding 于2000 年在他的博士论文中提出的，是针对分布式系统的软件架构风格。

目前很多公司采用传统的网络应用软件测试技术对REST架构软件进行测试，这种测试技术主要针对网络软件的User Interface层，由测试人员根据经验编写测试用例，测试方式以手动测试为主。通常情况下这种测试方式并不能够保证完全覆盖到底层的所有接口，不易发现软件深层次的逻辑问题，在发现问题时不易进行准确定位，繁琐和重复的手动测试也会降低网络应用软件的开发效率，导致网络应用软件的测试完备性和测试充分性不高，软件的性能得不到很好的保证。

API测试被认为比GUI测试更适合于测试自动化和连续测试（尤其是与敏捷软件开发和DevOps结合使用的自动化）[3][4]。引用的原因包括：

系统复杂度： GUI测试无法充分验证与多层体系结构关联的功能路径和后端API /服务。API被认为是与被测系统最稳定的接口。

具有快速反馈循环的短发布周期：敏捷和DevOps团队在进行短迭代和快速反馈循环后发现GUI测试需要大量的返工才能跟上频繁变化的步伐。API层的测试不那么脆弱，更易于维护。

由于这些原因，建议团队在降低对GUI测试的依赖的同时提高其API测试的级别。建议将API测试用于绝大多数的测试自动化工作和尽可能多的边缘测试。然后保留GUI测试，以在系统级别，移动测试和可用性测试中验证典型用例[3][4][10]。

总体来说，需要一套高效的软件测试框架对REST架构软件进行自动化测试，提高软件生命周期中的测试环节的效率，避免低级BUG的产生，提高API可靠性和稳定性。减少软件后期的经济支出，节约软件后期维护成本。

## 可行性分析

### 经济可行性

开发本系统需要购买云服务器作为载体，对服务器资源没有要求，购买最低配即可。本系统实现为纯软件实现，无需购买硬件，对服务器资源没有强依赖，只需要极小的资源即可运行。使用过程中也无多余费用的产生，开发所有使用的技术和工具都是免费的，没有版权，都是开源软件。而且本系统实现简单，开发周期短，人员经济支出基本没有，产生的效益远远大于专人软件测试所需的经济支出。所以从经济方面谈开发此系统是可行的。

### 技术可行性

本系统的实现主要使用Python进行开发，还使用一些Python的第三方支持库来实现一些功能。Python有着免费和容易上手的特点，用于自动化方面有着先天优势。数据库使用MySQL数据库，特点也很明显，稳定和免费。综上所述，本系统在技术方面是可行的。

### 运行可行性

本系统是自动化执行，输入相对应的参数和指定结果即可。没有复杂的操作，操作简单容易上手。

## 实验内容

通过Python编写自动化测试系统，用于Web API的功能性测试，帮助项目迭代时进行的大量接口测试。使用VUE和element-UI建立前端组件，用于更好的可视化体验。

针对不同的Web API进行测试，可查询测试结果，测试结果在前端展示。

## 实验目标

根据用户提供的测试用例，进行批量测试，记录测试结果。用户可查测试结果，可用测试定时任务，定期测试指定API。

每次测试记录用户信息和测试结果，用于报表使用。测试结果可以导出，便于项目汇报。

## 论文整体结构

第一章，绪论。本章节介绍了自动化API测试的研究背景及其意义，然后分析了现阶段国内外学者们在自动化API测试方法上的研究现状，以及现阶段依然存在的问题，并且进一步提出本文课题实验目标。

第二章，

第三章，

第四章，

第五章，

# 相关工具及技术介绍

## 开发环境

开发过程中，搭建开发环境是该系统开发的起步过程，同时标志着该实验开始的标志，是非常重要的一步，这一部分重点讲解如何配置开发使用的环境。

本次操作系统

本次系统开发过程使用的开发环境如表2-1所示：

表2-1 开发环境表

| 开发环境 | 操作系统：Microsoft windows 10 enterprise 64Bit |
| --- | --- |
| 编程语言：Python for Windows 3.6.4 64Bit |
| 开发IDE： PyCharm 2019.1 professional edition 64Bit |
| 三方库： flask、pymysql、json、yaml、time、logging、requests、ast、flask\_cors、flask\_apscheduler、datetime、VUE、element-UI |

## 编程语言

### Python

Python是一种解释，高层次的，通用的 编程语言。Python由Guido van Rossum创建并于1991年首次发布，Python的设计理念通过显着使用大量空白来强调代码的可读性。它的语言构造和面向对象的方法旨在帮助程序员为大型和大型项目编写清晰的逻辑代码[11]。

Python是动态类型化并被垃圾回收的。它支持多种编程范例，包括过程，面向对象和 函数式编程。由于Python具有完善的标准库，因此通常被称为“batteries included”的语言[13]。

Python在1980年代后期被构想为ABC语言的继承者。2000年发布的Python 2.0引入了诸如列表推导和能够收集参考周期的垃圾收集系统之类的功能。Python 3.0于2008年发布，是该语言的主要修订版，并不完全向后兼容，并且许多Python 2代码不会在未经修改的Python 3上运行[13]。

Python 解释器可用于许多操作系统。全球程序员社区开发和维护CPython，这是一个开源参考实现。一个非盈利性组织，在Python软件基金会，管理和指导资源，Python和CPython的发展[14]。

Python是一种多范式编程语言。完全支持面向对象的程序设计和结构化程序设计，并且其许多功能都支持功能性程序设计和面向方面的程序设计。通过扩展支持许多其他范例，包括按合同设计和逻辑编程[14]。

Python使用动态类型并将引用计数与循环检测垃圾回收器结合使用以进行内存管理。它还具有动态名称解析（后期绑定）功能，可以在程序执行期间绑定方法和变量名称[14]。

### JavaScript

JavaScript通常缩写为JS，是一种编程语言符合的ECMAScript的规范。

JavaScript是高级的，通常是即时编译的，并且是多范例的。它具有花括号语法，动态类型，基于原型 的面向对象和一流的功能。

除了HTML和CSS，JavaScript是万维网的核心技术之一。JavaScript支持交互式网页，并且是Web应用程序的重要组成部分。绝大多数网站将其用于客户端页面行为，并且所有主要的网络浏览器都有专用的JavaScript引擎来执行它[15]。

作为一种多范式语言，JavaScript支持事件驱动，功能和命令式 编程风格。它具有用于处理文本，日期，正则表达式，标准数据结构和文档对象模型（DOM）的应用程序编程接口（API ）。但是，该语言本身不包含任何输入输出，例如网络，存储或图形功能，因为主机环境（通常是Web浏览器）提供了这些API[16]。

JavaScript引擎最初仅在Web浏览器中使用，但现在通常通过Node.js嵌入到服务器端网站部署中。它们还嵌入在使用诸如Electron和Cordova之类的框架创建的各种应用程序中[16]。

尽管JavaScript和Java之间存在相似之处，包括语言名称，语法和各自的标准库，但这两种语言却截然不同，并且在设计上也有很大差异。

## 数据库存储

数据库是一个有组织的集合的数据，通常存储和从计算机系统电子地访问。在数据库更复杂的地方，通常使用正式的设计和建模技术来开发它们[17]。

该数据库管理系统（DBMS）是软件与交互的最终用户，应用程序和数据库本身来捕获和分析数据。DBMS软件还包含为管理数据库而提供的核心功能。数据库，DBMS和关联的应用程序的总和可以称为“数据库系统”。通常，术语“数据库”还用于宽松地指代任何DBMS，数据库系统或与数据库关联的应用程序[18]。

MySQL是开源 关系数据库管理系统（RDBMS）。根据GNU通用公共许可证的条款，MySQL是免费的开源软件，还可以根据各种专有许可证使用。MySQL由瑞典MySQL AB拥有和赞助，该公司由Sun Microsystems（现为Oracle Corporation）收购[18]。

2010年，当Oracle收购Sun时，Widenius 派遣了开源 MySQL项目来创建MariaDB。

## 前端展示技术

Vue是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

## Web技术

万维网（WWW），俗称的网络，是一个信息系统，其中的文件和其它网络资源被确定统一资源定位符，这可能是通过超文本相互链接，并且可以通过Internet访问。WWW的资源是通过超文本传输​​协议（HTTP）进行传输的，并且用户可以通过称为Web浏览器的软件应用程序对其进行访问。并由称为Web服务器的软件应用程序发布[19]。

英国科学家Tim Berners-Lee于1989年发明了万维网。他于1990年在瑞士日内瓦附近的欧洲核子研究组织（CERN）工作时编写了第一个网络浏览器。该浏览器于1991年在CERN外部发布，从1991年1月开始首先发布给其他研究机构，然后在1991年8月发布给公众。万维网一直是信息时代发展的中心，并且是数十亿人在互联网上进行交互的主要工具[20]。

Web资源可以是任何类型的下载媒体，但是网页是已经用超文本标记语言（HTML）格式化的超文本媒体。这种格式允许包含URL的嵌入式超链接，并允许用户导航到其他Web资源。除文本外，网页还可以包含对图像，视频，音频和软件组件的引用，这些信息在用户的 Web浏览器中显示为多媒体内容的连贯页面[21]。

一个具有共同主题，共同域名或同时具有两者的多个Web资源组成一个网站。网站存储在运行名为Web服务器的程序的计算机中，该程序响应用户计算机上运行的Web浏览器通过Internet发出的请求。网站内容可以主要由发布者提供，或在用户贡献内容或内容取决于用户或其行为的情况下以交互方式提供。出于各种信息，娱乐，商业，政府或非政府原因，可能会提供网站[21]。

# 测试方案及测试用例设计

文字说明……………………….

## 测试方案

## 测试用例设计

### 系统整体设计

### 数据库结构设计

### 日志模块设计

### 定时任务设计

# 测试实现

## API实现

## UI实现

## 系统UI展示

# 测试结论

结束语

参考文献

1. Testing APIs protects applications and reputations, by Amy Reichert, SearchSoftwareQuality March 2015
2. All About API Testing: An Interview with Jonathan Cooper, by Cameron Philipp-Edmonds, Stickyminds August 19, 2014
3. The Forrester Wave™ Evaluation Of Functional Test Automation (FTA) Is Out And It's All About Going Beyond GUI Testing Archived 2015-05-28 at the Wayback Machine, by Diego Lo Giudice, Forrester April 23, 2015
4. Produce Better Software by Using a Layered Testing Strategy, by SEAN Kenefick, Gartner January 7, 2014
5. Onus for third-party APIs is on enterprise developers, by Amy Reichert, SearchSoftwareQuality July 2014
6. Accelerate Development with Automated Testing, by Nathan Wilson, Gartner December 30, 2013
7. A Guidance Framework for Designing a Great Web API, by Eric Knipp and Gary Olliffe , Gartner August 20, 2014
8. The Fight Against Brittle Scripts and Software Defects, by Adrian Bridgwater, Dr. Dobb's Journal October 26, 2011
9. How Do We Learn Composite App Testing-Speak?, by Adrian Bridgwater, Dr. Dobb's Journal February 14, 2012
10. Cohn, Mike (2009). Succeeding with Agile: Software Development Using Scrum. Addison-Wesley Professional. p. 312. ISBN 978-0321579362.
11. Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.
12. Oliphant, Travis E. Python for Scientific Computing[J]. Computing in Science & Engineering, 9(3):10-20.
13. Rossum G V. The Python Language Reference Manual[M]. 2011.
14. Summerfield M. Programming in Python 3[M]. 2010.
15. Flanagan B D. JavaScript: The Definitive Guide[M]. 2011.
16. Musch J, Reips U D. A brief history of Web experimenting.[M]// false. 2000.
17. Widenius M , Axmark D , Dubois P . Mysql Reference Manual[M]. O'Reilly & Associates, Inc. 2002.
18. Vaswani V. MySQL Database Usage and Administration[M]// MySQL Database Usage & Administration. 2009.
19. Tobin, James (12 June 2012). Great Projects: The Epic Story of the Building of America, from the Taming of the Mississippi to the Invention of the Internet. Simon and Schuster. ISBN 978-0-7432-1476-6.
20. McPherson, Stephanie Sammartino (2009). Tim Berners-Lee: Inventor of the World Wide Web. Twenty-First Century Books. ISBN 978-0-8225-7273-2.
21. Dewey, Caitlin (12 March 2014). "36 Ways the Web Has Changed Us". The Washington Post. Archived from the original on 9 September 2015. Retrieved 1 August 2015.

附录

致谢

自己写