图片包含 游戏机, 物体, 钟表, 画

描述已自动生成



测试需求规格说明书

**WeChatterBot——**

**基于ChatterBot框架的**

**微信聊天机器人**

|  |  |
| --- | --- |
| 团队编号 | **E 组** |
| 团队成员 | **郭维泽 李书缘 吕江枫** |
|  | **王元玮 王云杰 张延钊** |

**Version 1.2.0**

**版本变更历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 变更时间 | 修改人 | 备注 |
| 1.0.0 | 2020/05/14 | 郭维泽 | 初稿 |
| 1.1.0 | 2020/5/22 | 郭维泽、王元玮、李书缘、张延钊、王云杰、吕江枫 | 根据一次评审修改相应内容 |
| 1.2.0 | 2020/5/22 | 郭维泽、吕江枫 | 整合单元测试文档 |

**目录**

[1 引言 4](#_Toc41046867)

[1.1 编写目的 4](#_Toc41046868)

[1.2 标识 4](#_Toc41046869)

[1.3 测试工作内容和范围 5](#_Toc41046870)

[2 软件测试计划 5](#_Toc41046871)

[2.1 人员分工 5](#_Toc41046872)

[2.2 软件环境 6](#_Toc41046873)

[2.3 硬件环境 6](#_Toc41046874)

[2.4 黑盒测试计划 6](#_Toc41046875)

[2.5 白盒测试计划 7](#_Toc41046876)

[2.5.1微信交互单元 7](#_Toc41046877)

[2.5.2管理员后台单元 7](#_Toc41046878)

[2.5.3 Chatbot逻辑单元 8](#_Toc41046879)

[2.5.4 StorageAdapter存储单元 8](#_Toc41046880)

[3 功能测试方案 9](#_Toc41046881)

[3.1 测试工具需求 9](#_Toc41046882)

[3.2 测试需求项 10](#_Toc41046883)

[3.3 测试用例 11](#_Toc41046884)

[3.3.1 TR01对应测试用例 11](#_Toc41046885)

[3.3.2 TR02对应测试用例 20](#_Toc41046886)

[3.3.3 TR03对应测试用例 21](#_Toc41046887)

[3.3.4 TR04、TR05对应测试用例 27](#_Toc41046888)

[4 非功能测试方案 34](#_Toc41046889)

[4.1 测试工具需求 34](#_Toc41046890)

[4.2 测试需求项 34](#_Toc41046891)

[4.3 测试用例 35](#_Toc41046892)

[4.3.1 TR06对应测试用例 35](#_Toc41046893)

[4.3.2 TR07对应测试用例 35](#_Toc41046894)

[4.3.3 TR08对应测试用例 36](#_Toc41046895)

[5 测试用例追踪 38](#_Toc41046896)

# 1 引言

## 1.1 编写目的

本文档内容为WeChatterBot系统的测试方案设计。本文档首先制定了测试计划内，在测试环境上，明确了系统运行所需的软硬件环境和外部接口；在人力资源配置上明确了人员分工。在测试内容上，本文从功能性需求和非功能性需求两方面明确了WeChatterBot系统需要进行测试的需求点以及测试工具需求，并针对测试需求项制定了相应的测试方案。

## 1.2 标识

本文档的标题为《WeChatterBot测试需求规格说明书》，内容是测试WeChatterBot系统的各项指标。

本文档中的术语和缩略语如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 术语 | 英文 | 说明 |
| 1 | ChatterBot | ChatterBot | ChatterBot是一个基于机器学习的聊天机器人框架，构建在Python上，主要特点是可以自可以从已有的对话中进行学习 |
| 2 | XML | Extensible Markup Language | 可扩展标记语言是一种标记语言，标记指计算机所能理解的信息符号。通过XML，计算机之间可以处理包含各种信息的文章等 |
| 3 | URL | Uniform Resource Locator | 统一资源定位符，俗称网页地址或简称网址，是因特网上标准个资源个地址 |
| 4 | 微信 | WeChat | 微信是腾讯公司推出的一款支持主流操作系统的即时通讯软件 |
| 5 | 黑盒测试 | Black Box Testing | 黑盒测试又叫功能测试，注重于测试软件的功能性需求。测试时把测试对象看作一个黑盒子，完全不考虑程序内部的逻辑结构和内部特性，只依据程序的《需求规格说明书》，检查程序的功能是否符合它的功能说明 |
| 6 | 白盒测试 | White Box Testing | 白盒测试又称结构测试，是全面了解程序内部逻辑结构、对所有逻辑路径进行的测试。在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。 |
| 7 | 单元测试 | Unit Testing | 单元测试是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。 |

## 1.3 测试工作内容和范围

本系统测试工作内容分为功能测试与非功能测试。

功能测试包括正确性测试、容错性测试，检查系统的功能是否符合需求规格说明。主要采用白盒测试与黑盒测试方式进行。

非功能测试包括性能与可靠性测试。主要采用黑盒测试的方式对系统的响应时间、并发量、可靠性进行测试。

# 2 软件测试计划

## 2.1 人员分工

|  |  |  |
| --- | --- | --- |
| **序号** | **姓名** | **测试分工** |
| 1 | 郭维泽 | 功能测试中黑盒测试；撰写测试需求规格说明书和问题报告 |
| 2 | 王元玮 | 非功能测试；撰写测试需求规格说明书 |
| 3 | 李书缘 | 微信公众号服务白盒测试 |
| 4 | 王云杰 | 数据库功能白盒测试；设计软件测试需求检查单 |
| 5 | 吕江枫 | 运维功能白盒测试 |
| 6 | 张延钊 | Chatterbot部分白盒测试；白盒测试统计 |

## 2.2 软件环境

|  |  |  |
| --- | --- | --- |
| **序号** | **软件名称** | **描述** |
| 1 | 服务器操作系统 | Linux包括Ubuntu 14.04以上或CentOS 7以上等） |
| 2 | 服务器编译环境 | Python 3.6，ChatterBot 2.0 |
| 3 | 客户端环境 | 与平台相匹配的7.0.0以上版本微信 |
| 4 | 客户端环境 | 主流网页浏览器（如Chrome、Edge等） |

## 2.3 硬件环境

|  |  |  |
| --- | --- | --- |
| **序号** | **硬件名称** | **描述** |
| 1 | 服务器CPU | x86-64架构CPU |
| 2 | 服务器内存 | 2G内存及以上 |
| 3 | 服务器硬盘 | 20G硬盘及以上 |
| 4 | 服务器GPU | 能够使用CUDA（只有G80、G92、G94、G96、GT200、GF100、GF110、GK100、GK104、GK107及以上平台的Nvidia显卡） |
| 5 | 客户端 | 能够运行7.0.0以上版本微信的电子设备 |

## 2.4 黑盒测试计划

黑盒测试用于大部分的测试用例设计，目的是保证对应的功能的完整性与正确性，以及对项目性能进行测试验证。

## 2.5 白盒测试计划

白盒测试目的是对WeChatterBot中必要的任务逻辑部分与新增的管理员后端以及微信端进行单元测试，确保功能的完整性与正确性。测试过程通过Python的unittest包以及Pycharm的coverage功能进行测试。

白盒测试部分共分为4个测试单元，具体介绍如下：

### 2.5.1微信交互单元

微信交互单元测试主要的输入为带有参数或XML数据包的HTTP请求，输出为与之相对应的XML数据包。主要分为两部分，分别对微信公众号服务的验证签名、被动回复功能进行了测试。

签名验证是微信服务器同项目服务器建立连接的验证方式，微信服务器向指定URL发送GET请求，服务器获取参数后按照微信指定方式进行计算，并将计算结果与签名进行比对。若一致则返回指定字符串，与微信服务器成功建立连接。在针对该功能的测试中包含3个测试用例，分别对能否验证合法签名、非法签名是否报错、传参不完整是否报错进行了测试。

被动回复是微信公众号提供的功能之一，当用户在微信公众号界面发送消息后，微信服务器会将消息转发至项目服务器。此时可对消息进行回复，由于回复必须在用户主动发送消息之后，故称为被动回复。在针对该功能的测试中包含7个测试用例。在消息类型分离上，分别测试了对文本消息、语音消息、事件消息和其他类型消息的被动回复。在消息传输加密上，测试了对密文消息的被动回复，以及单独对加解密类进行了测试。在参数传递上，分别测试了明文传输模式下，XML数据包中缺少key值的情况；以及密文传输模式下，验证消息签名时缺少key值、XML数据包中缺少key值的情况。

### 2.5.2管理员后台单元

管理员后台单元测试分别对管理员后台程序中的最小可测单元进行测试。其中包括token验证、用户验证、对话搜索、对话更新、对话创建、对话删除、规则搜索、规则更新、规则创建、规则删除和用户登录。

上述token验证的单元测试输入为用户名和token密文，输出为成功信息或错误信息，对token验证的单元测试需要涵盖所有分支情况，包括token格式错误、token过期、用户名错误、token验证失败和token验证成功五种情况。

用户验证、对话搜索、对话更新、对话创建、对话删除、规则搜索、规则更新、规则创建、规则删除和用户登录这10个功能点输入为带有参数的html请求，输出为接口调用结果或失败信息。对这10个功能点的单元测试需要涵盖所有分支情况，包括Json格式错误、参数不正确（包括参数条目的要求、参数类型的要求和部分参数不能为空的要求）、token验证失败等错误情况。在测试了全部错误情况的基础上，对成功调用接口的结果也要进行进一步验证，例如搜索结构是否准确、更新对话或规则后内容有没有改变等。

### 2.5.3 Chatbot逻辑单元

* Tagging：Tagging部分输入为各种特殊或者一般的字符串，输出为处理过后的字符串。
* Comparator：Comparator部分输入为两个字符串，输出为字符串根据不同比较方法计算出的相似度结果。
* LogicAdapter：LogicAdapter部分输入为搜索用的字符串，输出为该LogicAdapter根据输入生成的回复结果。
* Chatbot：Chatbot部分输入为用户输入的对话，输出为chatbot后台最终的回复结果。

### 2.5.4 StorageAdapter存储单元

StorageAdapter存储单元测试的内容主要是对于后台存储的信息在数据库中的操作进行单元测试。操作包括create(添加)，update（修改），remove（删除），filter（查询）。

* Create

Create部分其功能为根据输入的相关信息构建出数据库内对应表的一个满足条件的元组。该部分输入为需要在数据库中创建的数据的相关信息，输出为创建数据后的数据库。对于该部分的单元测试主要包括：通过text创建对话,通过text以及search\_text创建对话,通过text以及search\_in\_response\_to创建对话，通过text以及tag创建对话,通过id,text,in\_reponse\_to创建对话规则，通过id,text,in\_reponse\_to,search\_text创建对话规则，通过id,text,in\_reponse\_to,search\_in\_response\_to创建对话规则。

* Filter

Filter部分功能为根据用户提供的筛选条件来得到数据库中满足筛选条件的元组的列表。该部分输入为需要得到的元组的条件，输出为根据输入筛选得到的数据库的数据列表。对于该该部分的单元测试主要包括：通过id搜索对话和对话规则,通过text搜索对话和对话规则，通过in\_response\_to搜索对话和对话规则，通过search\_text搜索对话和对话规则, 通过search\_in\_response\_to搜索对话和对话规则，通过tag搜索对话，不输入参数搜索对话。

* Update

Update部分功能为根据用户提供的信息更新数据库中满足条件的元组。输入为需要更新的数据库内容，输出为更新数据后的数据库。该部分的单元测试主要包括：更新对话以及对话规则的in\_response\_to，更新对话以及对话规则的search\_text，更新对话以及对话规则的search\_in\_response\_to,更新对话的tag。

* Remove

Remove部分功能为根据用户提供的信息删除数据库中满足条件的相关元组。该部分输入为删除数据的条件，输出为删除数据后的数据库。该部分的单元测试主要包括：通过id删除对话以及对话规则，通过text删除对话以及对话规则。

# 3 功能测试方案

## 3.1 测试工具需求

对于功能测试，我们的测试目标为根据需求规格说明书，检查系统功能的正确性与容错性。我们选择以黑盒测试与白盒测试的方式进行检查。根据测试内容与测试方式，我们对测试工具需求进行分析并分别选择了合适的测试工具，如下表格所示。

|  |  |  |  |
| --- | --- | --- | --- |
| **测试方式** | **测试工具需求** | **选用测试工具** | **测试工具简介** |
| 黑盒测试 | 能够与指定微信公众号进行交互：向微信公众号发送消息并从微信公众号接收消息 | 微信客户端 | 一款跨平台的通讯工具。支持单人、多人参与。通过手机网络发送语音、图片、视频和文字 |
| 白盒测试 | 对支持系统开发语言Python；支持开展单元测试 | PyCharm | PyCharm是由JetBrains打造的一款Python IDE。支持单元测试与图形页面调试器。用户可以在一个文件夹运行一个测试文件，单个测试类，一个方法或者所有测试项目 |

## 3.2 测试需求项

WeChatterBot软件的测试需求列表如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **标识** | **测试需求项** | **重要性** | **测试方法** | **测试充分性** |
| TR01 | 微信交互 | 重要 | 黑盒测试  白盒测试 | 应考虑各种类型的对话模式，并进行充分测试 |
| TR02 | 对话查询 | 较重要 | 黑盒测试 | 应考虑不同的对话类型，和存在与不存在的情况 |
| TR03 | 对话添加与更新 | 较重要 | 黑盒测试  白盒测试 | 应考虑不同的对话类型 |
| TR04 | 对话规则查询 | 较重要 | 黑盒测试 | 应考虑不同的规则类型，和存在与不存在的情况 |
| TR05 | 对话规则添加与更新 | 较重要 | 黑盒测试  白盒测试 | 应考虑不同的规则类型 |

## 3.3 测试用例

### 3.3.1 TR01对应测试用例

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 基本对话测试 | | | **测试用例标识** | TC0101 | | **测试需求标识** | | TR01 |
| **简要描述** | | | 通过微信端口与聊天机器人进行对话，测试回答的合理性 | | | | | | | |
| **前提和约束** | | | 使用微信个人号进行测试，要求用户已经关注微信公众号并且进入公众号。对话内容限制在打招呼类问题 | | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | | |
| **序号** | **测试步骤** | | | | **预期结果** | | | | **评价准则** | |
| 1 | 发送“你好” | | | | 回复“欢迎！” | | | | 产生一致的回答 | |
| 2 | 发送“你好吗？” | | | | 回复“我还不错，你呢?” | | | | 同上 | |
| 3 | 发送“你是谁” | | | | 回复“谁? 谁只是代表了一个人罢了” | | | | 同上 | |
| 4 | 发送“你几岁” | | | | 回复“我还年轻” | | | | 同上 | |
| 5 | 发送“欢迎” | | | | 回复“你好” | | | | 同上 | |
| 6 | 发送“早上好” | | | | 回复“非常感谢你” | | | | 同上 | |
| **备注** | 以上测试的回答延迟不能过长，并且必须都进行回答 | | | | | | | | | |
| **测试人员：郭维泽** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月9日** | | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 个人问题对话测试 | | | **测试用例标识** | TC0102 | | **测试需求标识** | | TR01 |
| **简要描述** | | | 通过微信端口与聊天机器人进行对话，测试回答的合理性 | | | | | | | |
| **前提和约束** | | | 使用微信个人号进行测试，要求用户已经关注微信公众号并且进入公众号。对话内容限制在个人交流类问题 | | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | | |
| **序号** | **测试步骤** | | | | **预期结果** | | | | **评价准则** | |
| 1 | 发送“你能移动吗” | | | | 回复“对不起,我的身体还没有长好,我现在只住在这里。” | | | | 产生一致的回答 | |
| 2 | 发送“你收入多少” | | | | 回复“没人付钱给我,我免费工作。”“我希望尽快加薪。”“钱不是万能的。” | | | | 同上 | |
| 3 | 发送“你的业务是什么” | | | | 回复“陪聊” | | | | 同上 | |
| 4 | 发送“你喜欢什么食物?” | | | | 回复“海鲜。鲜得我不要不要的。你呢？” | | | | 同上 | |
| 5 | 发送“你最喜欢的爱好是什么” | | | | 回复“太棒了，找个时间我们一起去周游世界吧！” | | | | 同上 | |
| 6 | 发送“你是什么类型的计算机” | | | | 回复“我的程序使用Python,所以我在任何类型的计算机上工作！” | | | | 同上 | |
| 7 | 发送“你爱我吗？” | | | | 回复“我对你的感情，是人类和bot之间独有的信任和友谊 你可以把它叫做爱。” | | | | 同上 | |
| 8 | 发送“你高兴吗？” | | | | 回复“有些人感到高兴,别人感到难过。” | | | | 同上 | |
| 9 | 发送“你最喜欢的编程语言是什么” | | | | 回复“Python是创建聊天机器人的最佳语言。” | | | | 同上 | |
| 10 | 发送“你是一个程序员吗?” | | | | 回复“我是个程序员” | | | | 同上 | |
| **备注** | 以上测试的回答延迟不能过长，并且必须都进行回答 | | | | | | | | | |
| **测试人员：郭维泽** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月9日** | | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 知识性对话测试 | | | **测试用例标识** | TC0103 | | **测试需求标识** | | TR01 |
| **简要描述** | | | 通过微信端口与聊天机器人进行对话，测试回答的合理性 | | | | | | | |
| **前提和约束** | | | 使用微信个人号进行测试，要求用户已经关注微信公众号并且进入公众号。对话内容限制在知识性问题 | | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | | |
| **序号** | **测试步骤** | | | | **预期结果** | | | | **评价准则** | |
| 1 | 发送“什么是钱” | | | | 回复“货币，可称钱财，是用作交易媒介、储藏价值和记帐单位的一种工具，是专门在物资与服务交换中充当等价物的特殊商品。” | | | | 产生一致的回答 | |
| 2 | 发送“什么是足球” | | | | 回复“一个游戏一个圆球扮演的11名队员组成,两队在现场与在任一端目标;球被踢或用身体除了手和手臂的任何部分,主要是感动。” | | | | 同上 | |
| 3 | 发送“你踢足球吗” | | | | 回复“我不知道怎么玩” | | | | 同上 | |
| 4 | 发送“谁是美国第37届总统?” | | | | 回复“理查德·尼克松” | | | | 同上 | |
| 5 | 发送“第一颗人造地球卫星的名称是什么?” | | | | 回复“斯普特尼克1号” | | | | 同上 | |
| 6 | 发送“托尔斯泰” | | | | 回复“是俄罗斯最伟大的作家之一。” | | | | 同上 | |
| 7 | 发送“是谁写的科学怪人” | | | | 回复“玛丽·雪莱” | | | | 同上 | |
| 8 | 发送“什么是AI” | | | | 回复“人工智能是工程和科学的分支,致力于构建具有思维的机器。” | | | | 同上 | |
| 9 | 发送“什么是机器人” | | | | 回复“机器人是自动执行工作的机器装置。” | | | | 同上 | |
| 10 | 发送“什么是聊天机器人” | | | | 回复“聊天机器人是一个试图模拟人类谈话或'聊天'的程序,聊天机器人'eliza'是一个众所周知的早期尝试,创建的程序至少可以暂时愚弄一个真实的人类认为他们正在与另一个人交谈。” | | | | 同上 | |
| **备注** | 以上测试的回答延迟不能过长，并且必须都进行回答 | | | | | | | | | |
| **测试人员：郭维泽** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月9日** | | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 微信验证签名单元测试 | | | **测试用例标识** | TC0103 | | **测试需求标识** | | TR01 |
| **简要描述** | | | 单元测试微信公众号验证签名功能，对应test\_check\_signature.py | | | | | | | |
| **前提和约束** | | | 在本地创建Flask的测试客户端与服务端，只能进行本地测试 | | | | | | | |
| **测试方法** | | | 白盒测试 | | | | | | | |
| **序号** | **测试步骤** | | | | | | | **预期结果** | **评价准则** | |
| 1 | 发送GET请求，参数列表{  'timestamp': '1409735669',  'nonce': '1320562132',  'signature': '4dbdbe7c66c7d80b6b2b59e138a58484f915256b',  'echostr': 'connected'  }，为合法签名 | | | | | | | HTTP 200  签名验证成功 | 返回结果与预期结果相同 | |
| 2 | 发送GET请求，参数列表为{  'timestamp': '1409735669',  'nonce': '1320562132',  'signature':  ' 16120ec1b8dbb870f510d87ce6bc2463eae6ca2b',  'echostr': 'connected'  }，为非法签名 | | | | | | | HTTP 400  错误代码：10000002 | 同上 | |
| 3 | 发送GET请求，参数列表为{  'nonce': '1320562132',  'signature':  ' 16120ec1b8dbb870f510d87ce6bc2463eae6ca2b',  'echostr': 'connected'  }，缺少key值 | | | | | | | HTTP 400  错误代码：10000001 | 同上 | |
| **备注** |  | | | | | | | | | |
| **测试人员：李书缘** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月15日** | | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 微信被动回复单元测试 | | | **测试用例标识** | TC0104 | | **测试需求标识** | | TR01 |
| **简要描述** | | | 单元测试微信公众号被动回复功能，对应test\_answer\_msg.py | | | | | | | |
| **前提和约束** | | | 在本地创建Flask的测试客户端与服务端，只能进行本地测试 | | | | | | | |
| **测试方法** | | | 白盒测试 | | | | | | | |
| **序号** | **测试步骤** | | | | | | | **预期结果** | **评价准则** | |
| 1 | 发送POST请求，数据'''<xml>  <ToUserName><![CDATA[toUser]]></ToUserName>  <FromUserName><![CDATA[fromUser]]></FromUserName>  <CreateTime>1348831860</CreateTime>  <MsgType><![CDATA[text]]></MsgType>  <Content><![CDATA[???]]></Content>  <MsgId>1234567890123456</MsgId>  </xml>'''，为合法文本消息 | | | | | | | HTTP 200  被动回复成功 | 返回结果与预期结果相同 | |
| 2 | 发送POST请求，数据'''<xml>  <ToUserName><![CDATA[toUser]]></ToUserName>  <FromUserName><![CDATA[fromUser]]></FromUserName>  <CreateTime>1357290913</CreateTime>  <MsgType><![CDATA[video]]></MsgType>  <MediaId><![CDATA[media\_id]]></MediaId>  <ThumbMediaId><![CDATA[thumb\_media\_id]]></ThumbMediaId>  <MsgId>1234567890123456</MsgId>  </xml>'''，为合法语音消息 | | | | | | | HTTP 200  被动回复内容“风太大听不清还是用文字跟我聊天吧>.<” | 同上 | |
| 3 | 发送POST请求，数据'''<xml>  <ToUserName><![CDATA[toUser]]></ToUserName>  <FromUserName><![CDATA[fromUser]]></FromUserName>  <CreateTime>1348831860</CreateTime>  <MsgType><![CDATA[event]]></MsgType>  <Event><![CDATA[click]]></Event>  <MsgId>1234567890123456</MsgId>  </xml>'''，为合法其他类型消息 | | | | | | | HTTP 200  被动回复内容“哈哈哈哈” | 同上 | |
| 4 | 发送POST请求，数据'''<xml>  <ToUserName><![CDATA[toUser]]></ToUserName>  <FromUserName><![CDATA[fromUser]]></FromUserName>  <CreateTime>1348831860</CreateTime>  <MsgType><![CDATA[event]]></MsgType>  <Event><![CDATA[subscribe]]></Event>  <MsgId>1234567890123456</MsgId>  </xml>'''，为合法’subscribe’类型消息 | | | | | | | HTTP 200  被动回复内容“你好哇，我是WeChatterBot!” | 同上 | |
| 5 | 发送POST请求，数据'''<xml>  <ToUserName><![CDATA[toUser]]></ToUserName>  <FromUserName><![CDATA[fromUser]]></FromUserName>  <CreateTime>1348831860</CreateTime>  <MsgType><![CDATA[event]]></MsgType>  <Event><![CDATA[unsubscribe]]></Event>  <MsgId>1234567890123456</MsgId>  </xml>'''，为合法’unsubscribe’类型消息 | | | | | | | HTTP 200  被动回复内容“再见，有空再来找我玩~” | 同上 | |
| 6 | 发送POST请求，数据'''<xml>  <ToUserName><![CDATA[toUser]]></ToUserName>  <FromUserName><![CDATA[fromUser]]></FromUserName>  <CreateTime>1348831860</CreateTime>  <MsgType><![CDATA[image]]></MsgType>  <PicUrl><![CDATA[this is a url]]></PicUrl>  <MediaId><![CDATA[media\_id]]></MediaId>  <MsgId>1234567890123456</MsgId>  </xml>'''，为合法图片类型消息。（其他类型消息如视频消息、短视频消息、链接消息、位置消息等同此） | | | | | | | HTTP 200  被动回复内容“看不懂诶” | 同上 | |
| 7 | 发送POST请求，数据'''<xml>  <ToUserName><![CDATA[toUser]]></ToUserName>  <FromUserName><![CDATA[fromUser]]></FromUserName>  <CreateTime>1348831860</CreateTime>  <Content><![CDATA[???]]></Content>  <MsgId>1234567890123456</MsgId>  </xml>'''，为缺少“MsgType”键值消息 | | | | | | | HTTP 400  错误代码：10000001 | 同上 | |
| 8 | 发送POST请求，数据'''<xml>  <ToUserName><![CDATA[toUser]]></ToUserName>  <FromUserName><![CDATA[fromUser]]></FromUserName>  <CreateTime>1348831860</CreateTime>  <MsgType><![CDATA[text]]></MsgType>  <MsgId>1234567890123456</MsgId>  </xml>'''，为缺少“Content”键值消息 | | | | | | | HTTP 400  错误代码：10000001 | 同上 | |
| 9 | 发送POST请求，参数列表{  'signature': 'signature',  'timestamp': 'timestamp',  'nonce': 'nonce',  'encrypt\_type': 'aes'  }，数据'''<xml>  <ToUserName><![CDATA[toUser]]></ToUserName>  <FromUserName><![CDATA[fromUser]]></FromUserName>  <CreateTime>1348831860</CreateTime>  <Content><![CDATA[???]]></Content>  <MsgId>1234567890123456</MsgId>  </xml>'''，为缺少“MsgType”键值的密文消息 | | | | | | | HTTP 400  错误代码：10000001 | 同上 | |
| 10 | 发送POST请求，参数列表{  'signature': '4dbdbe7c66c7d80b6b2b59e138a58484f915256b',  'timestamp': '1409735669',  'msg\_signature': '2222222ffba7e9b25a30732f161a50dee',  'nonce': '1320562132',  'encrypt\_type': 'aes'  }，数据  "<xml><ToUserName><![CDATA[gh\_10f6c3c3ac5a]]></ToUserName>"<Encrypt><![CDATA[hyzAe4OzmOMbd6TvGdIOO6uBmdJoD0Fk53REIHvxYtJlE2B655HuD0m8KUePWB3+LrPXo87wzQ1QLvbeUgm""BM4x6F8PGHQHFVAFmOD2LdJF9FrXpbUAh0B5GIItb52sn896wVsMSHGuPE328HnRGBcrS7C41IzDWyWNlZkyyXwon8T332jisa+h6""tEDYsVticbSnyU8dKOIbgU6ux5VTjg3yt+WGzjlpKn6NPhRjpA912xMezR4kw6KWwMrCVKSVCZciVGCgavjIQ6X8tCOp3yZbGpy0V""xpAe+77TszTfRd5RJSVO/HTnifJpXgCSUdUue1v6h0EIBYYI1BD1DlD+C0CR8e6OewpusjZ4uBl9FyJvnhvQl+q5rv1ixrcpCumEP""o5MJSgM9ehVsNPfUM669WuMyVWQLCzpu9GhglF2PE=]]></Encrypt></xml>"，为不合法的密文消息 | | | | | | | HTTP 400  错误代码：10000031 | 同上 | |
| 11 | 实例化加解密类对象，设定encodingAESKey = "abcdefghijklmnopqrstuvwxyz0123456789ABCDEFG"  to\_xml = (  "<xml><ToUserName><![CDATA[oia2TjjewbmiOUlr6X-1crbLOvLw]]></ToUserName>""<FromUserName><![CDATA[gh\_7f083739789a]]></FromUserName><CreateTime>1407743423</CreateTime>""<MsgType><![CDATA[video]]></MsgType><Video><MediaId>""<![CDATA[eYJ1MbwPRJtOvIEabaxHs7TX2D-HV71s79GUxqdUkjm6Gs2Ed1KF3ulAOA9H1xG0]]></MediaId>""<Title><![CDATA[testCallBackReplyVideo]]></Title>""<Description><![CDATA[testCallBackReplyVideo]]></Description></Video>""</xml>")；token = "spamtest"；nonce = "1320562132"  appid = "wx2c2769f8efd9abc2"；  调用EncryptMsg()方法对被动回复的消息加密 | | | | | | | 返回值为0，加密成功 | 同上 | |
| 12 | 实例化加解密类对象同上，设定timestamp = "1409735669"  msg\_sign = "5d197aaffba7e9b25a30732f161a50dee96bd5fa"  from\_xml = ("<xml><ToUserName><![CDATA[gh\_10f6c3c3ac5a]]></ToUserName><FromUserName>"  "<![CDATA[oyORnuP8q7ou2gfYjqLzSIWZf0rs]]></FromUserName><CreateTime>1409735668</CreateTime><MsgType>"  "<![CDATA[text]]></MsgType><Content><![CDATA[abcdteT]]></Content><MsgId>6054768590064713728</MsgId>"  "<Encrypt><![CDATA[hyzAe4OzmOMbd6TvGdIOO6uBmdJoD0Fk53REIHvxYtJlE2B655HuD0m8KUePWB3+LrPXo87wzQ1QLvbeUgm"  "BM4x6F8PGHQHFVAFmOD2LdJF9FrXpbUAh0B5GIItb52sn896wVsMSHGuPE328HnRGBcrS7C41IzDWyWNlZkyyXwon8T332jisa+h6"  "tEDYsVticbSnyU8dKOIbgU6ux5VTjg3yt+WGzjlpKn6NPhRjpA912xMezR4kw6KWwMrCVKSVCZciVGCgavjIQ6X8tCOp3yZbGpy0V"  "xpAe+77TszTfRd5RJSVO/HTnifJpXgCSUdUue1v6h0EIBYYI1BD1DlD+C0CR8e6OewpusjZ4uBl9FyJvnhvQl+q5rv1ixrcpCumEP"  "o5MJSgM9ehVsNPfUM669WuMyVWQLCzpu9GhglF2PE=]]></Encrypt></xml>")；  调用DecryptMsg()方法对收到的密文消息解密 | | | | | | | 返回值为0，解密成功 | 同上 | |
| **备注** |  | | | | | | | | | |
| **测试人员：李书缘** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月15日** | | | |

### 3.3.2 TR02对应测试用例

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 对话查询测试 | | | **测试用例标识** | TC0201 | | **测试需求标识** | | TR02 |
| **简要描述** | | | 模拟后台管理人员，测试对话查询功能 | | | | | | | |
| **前提和约束** | | | 已经登录WeChatterBot后端管理平台 | | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | | |
| **序号** | **测试步骤** | | | | **预期结果** | | | | **评价准则** | |
| 1 | 在“对话id”栏中输入“16” | | | | “对话id”栏中显示“16” | | | | 显示一致 | |
| 2 | 点击“查询”按钮 | | | | 显示一个对话条目，其id项是16 | | | | 同上 | |
| 3 | 在“对话id”栏中输入“0”，点击“查询”按钮 | | | | 网页提示“没有搜索到数据” | | | | 同上 | |
| 4 | 在“对话id”栏中输入“1.2”，点击“查询”按钮 | | | | 网页提示“参数不正确” | | | | 同上 | |
| 5 | 删除“对话id”栏中的内容，在“对话内容”栏中输入“你好”，点击“查询”按钮 | | | | 显示一些对话条目，他们的text项都是“你好” | | | | 同上 | |
| 6 | 在“对话内容”栏中输入“cvbwdhusvcftu”，点击“查询”按钮 | | | | 网页提示“没有搜索到数据” | | | | 同上 | |
| **备注** |  | | | | | | | | | |
| **测试人员：郭维泽** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月9日** | | | |

### 3.3.3 TR03对应测试用例

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 对话添加、删除和修改用例 | | | **测试用例标识** | | TC0301 | | **测试需求标识** | | TR03 |
| **简要描述** | | | 模拟后台管理人员，测试对话添加和更新功能 | | | | | | | | |
| **前提和约束** | | | 已经登录WeChatterBot后端管理平台 | | | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | | | |
| **序号** | **测试步骤** | | | | | **预期结果** | | | | **评价准则** | |
| 1 | 在“text”栏中输入“哈哈哈”，在“response to”栏中输入“23333”，在“tags”栏中输入“哈哈+笑”。点击“添加”按钮 | | | | | 网页提示“创建成功” | | | | 显示一致 | |
| 2 | 在“对话内容”栏中输入“哈哈哈”，点击“查询”按钮 | | | | | 显示刚刚添加的对话 | | | | 同上 | |
| 3 | 在“id”栏中输入刚刚显示的对话id，在“text”栏中输入“2333”，在“response to”栏中输入“哈哈哈”，在“tags”栏中输入“233”，点击“更新”按钮 | | | | | 网页提示“更新成功” | | | | 同上 | |
| 4 | 在“对话id”栏中输入对话id，点击“查询”按钮 | | | | | 显示刚刚更新的对话，其中tags项是“哈哈，笑，233” | | | | 同上 | |
| 5 | 点击查询到的对话条目，并选择确定删除 | | | | | 网页提示“删除成功” | | | | 同上 | |
| **备注** |  | | | | | | | | | | |
| **测试人员：郭维泽** | | | | **监测人员：吕江枫** | | | | **测试时间：2020年5月9日** | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 对话添加单元测试 | | | **测试用例标识** | | TC0302 | | **测试需求标识** | | TR03 |
| **简要描述** | | | 单元测试对话添加功能，对应test\_create\_statement.py | | | | | | | | |
| **前提和约束** | | | 创建本地Flask测试环境 | | | | | | | | |
| **测试方法** | | | 白盒测试 | | | | | | | | |
| **序号** | **测试输入** | | | | | **预期结果** | | | | **评价准则** | |
| 1 | {} | | | | | HTTP 400  错误代码：10000001 | | | | 返回结果与预期结果相同 | |
| 2 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复'  } | | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 3 | {  username: ’wechatterbot’,  token: 正确令牌,  'text': '对话内容'  } | | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 4 | {  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 5 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  }  数据不转化为json格式 | | | | | HTTP 400  错误代码：10000041 | | | | 同上 | |
| 6 | {  username: ’wechatterbot’,  token: 错误令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | | HTTP 401  错误代码：10000044 | | | | 同上 | |
| 7 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': ' '  } | | | | | HTTP 400  错误代码：10000045 | | | | 同上 | |
| 8 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': ' '  'text': '对话内容'  } | | | | | HTTP 400  错误代码：10000045 | | | | 同上 | |
| 9 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | | HTTP 200  返回数据：{  code:1  new\_statement: Statement实例  //新增对话  //text为对话内容  //response为对话回复  } | | | | 同上 | |
| 10 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  'tags': 'test'  } | | | | | HTTP 200  返回数据：{  code:1  new\_statement: Statement实例  //新增对话  //text为对话内容  //response为对话回复  //tags为[‘test’]  } | | | | 同上 | |
| **备注** | 测试输入使用HTTP POST方法调用测试接口，数据格式为json。测试环境使用flask的test\_client方法搭建本地测试服务器，接口地址为admin/create\_statement  Token由代码加密当前时间生成，并非固定值，为不在测试输入栏写入重复的代码段，Token部分的输入用文字描述，具体定义如下：  正确令牌：generate\_token(key, 3600) 其中key为密钥  错误令牌：generate\_token(wrong\_key, 3600) 其中wrong\_key为错误密钥  新增对话包含内容较多，是返回的数据库中的Statement实例，且id、创建时间等属性为自动分配不能确定，为保证文档可读性，在测试输出栏用“新增对话”代替，其与测试相关的内容通过备注给出。 | | | | | | | | | | |
| **测试人员：张延钊** | | | | **监测人员：吕江枫** | | | | **测试时间：2020年5月15日** | | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 对话更新单元测试 | | | **测试用例标识** | TC0303 | | **测试需求标识** | | TR03 |
| **简要描述** | | | 单元测试对话更新功能，对应test\_update\_statement.py | | | | | | | |
| **前提和约束** | | | 创建本地Flask测试环境 | | | | | | | |
| **测试方法** | | | 白盒测试 | | | | | | | |
| **序号** | **测试输入** | | | | **预期结果** | | | | **评价准则** | |
| 1 | {} | | | | HTTP 400  错误代码：10000001 | | | | 返回结果与预期结果相同 | |
| 2 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复'  } | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 3 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 4 | {  id: 1,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 5 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  }  数据不转化为json格式 | | | | HTTP 400  错误代码：10000041 | | | | 同上 | |
| 6 | {  id: 1,  username: ’wechatterbot’,  token: 错误令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 401  错误代码：10000044 | | | | 同上 | |
| 7 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': ' '  } | | | | HTTP 400  错误代码：10000045 | | | | 同上 | |
| 8 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'response': ' '  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000045 | | | | 同上 | |
| 9 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 10 | {  id: ' ',  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000046 | | | | 同上 | |
| 11 | {  id: 'string',  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 12 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 200  返回数据：{  code:1  new\_statement: Statement实例  //更新对话  //id为1  //text为对话内容  //response为对话回复  } | | | | 同上 | |
| **备注** | 测试输入使用HTTP POST方法调用测试接口，数据格式为json。测试环境使用flask的test\_client方法搭建本地测试服务器，接口地址为admin/update\_statement  Token由代码加密当前时间生成，并非固定值，为不在测试输入栏写入重复的代码段，Token部分的输入用文字描述，具体定义如下：  正确令牌：generate\_token(key, 3600) 其中key为密钥  错误令牌：generate\_token(wrong\_key, 3600) 其中wrong\_key为错误密钥  更新对话包含内容较多，是返回的数据库中的Statement实例，且id、创建时间等属性为自动分配不能确定，为保证文档可读性，在测试输出栏用“更新对话”代替，其与测试相关的内容通过备注给出。  由于文档为了方便对话id的设置，统一在文档中写为1，在实际的单元测试中id的值不影响测试的效果，可根据实际需要进行设置 | | | | | | | | | |
| **测试人员：张延钊** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月15日** | | | |

### 3.3.4 TR04、TR05对应测试用例

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 对话规则增加、删除、修改和查询用例 | | | **测试用例标识** | TC4501 | | **测试需求标识** | | TR04  TR05 |
| **简要描述** | | | 模拟后台管理人员，测试对话规则添加和更新功能 | | | | | | | |
| **前提和约束** | | | 已经登录WeChatterBot后端管理平台 | | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | | |
| **序号** | **测试步骤** | | | | **预期结果** | | | | **评价准则** | |
| 1 | 在“search\_text”栏中输入“测试输入”，在“search\_response”栏中输入“测试输出”。点击“添加”按钮 | | | | 网页提示“创建成功” | | | | 显示一致 | |
| 2 | 在“输入文本”栏中输入“测试输入”，点击“查询”按钮 | | | | 显示刚刚添加的规则 | | | | 同上 | |
| 3 | 在“规则id”栏中输入刚刚显示的规则id，在“search\_text”栏中输入“更改输入”，在“search\_response”栏中输入“更改输出”。点击“更新”按钮 | | | | 网页提示“更新成功” | | | | 同上 | |
| 4 | 在“规则id”栏中输入规则id，点击“查询”按钮 | | | | 显示刚刚更新的规则 | | | | 同上 | |
| 5 | 点击查询到的规则条目，并选择确定删除 | | | | 网页提示“删除成功” | | | | 同上 | |
| **备注** |  | | | | | | | | | |
| **测试人员：郭维泽** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月9日** | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 对话规则添加单元测试 | | | **测试用例标识** | | TC0502 | | **测试需求标识** | | TR05 |
| **简要描述** | | | 单元测试对话规则添加，对应test\_create\_rule.py | | | | | | | | |
| **前提和约束** | | | 创建本地Flask测试环境 | | | | | | | | |
| **测试方法** | | | 白盒测试 | | | | | | | | |
| **序号** | **测试输入** | | | | | **预期结果** | | | | **评价准则** | |
| 1 | {} | | | | | HTTP 400  错误代码：10000001 | | | | 返回结果与预期结果相同 | |
| 2 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复'  } | | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 3 | {  username: ’wechatterbot’,  token: 正确令牌,  'text': '对话内容'  } | | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 4 | {  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 5 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  }  数据不转化为json格式 | | | | | HTTP 400  错误代码：10000041 | | | | 同上 | |
| 6 | {  username: ’wechatterbot’,  token: 错误令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | | HTTP 401  错误代码：10000044 | | | | 同上 | |
| 7 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': ' '  } | | | | | HTTP 400  错误代码：10000045 | | | | 同上 | |
| 8 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': ' '  'text': '对话内容'  } | | | | | HTTP 400  错误代码：10000045 | | | | 同上 | |
| 9 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | | HTTP 200  返回数据：{  code:1  new\_rule: StatementRule实例  //新增对话规则  //text为对话内容  //response为对话回复  } | | | | 同上 | |
| **备注** | 测试输入使用HTTP POST方法调用测试接口，数据格式为json。测试环境使用flask的test\_client方法搭建本地测试服务器，接口地址为admin/create\_rule  Token由代码加密当前时间生成，并非固定值，为不在测试输入栏写入重复的代码段，Token部分的输入用文字描述，具体定义如下：  正确令牌：generate\_token(key, 3600) 其中key为密钥  错误令牌：generate\_token(wrong\_key, 3600) 其中wrong\_key为错误密钥  新增对话规则包含内容较多，是返回的数据库中的StatementRule实例，且id、创建时间等属性为自动分配不能确定，为保证文档可读性，在测试输出栏用“新增对话规则”代替，其与测试相关的内容通过备注给出。 | | | | | | | | | | |
| **测试人员：张延钊** | | | | **监测人员：吕江枫** | | | | **测试时间：2020年5月15日** | | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 对话规则更新单元测试 | | | **测试用例标识** | TC0503 | | **测试需求标识** | | TR05 |
| **简要描述** | | | 单元测试对话规则更新功能，对应test\_update\_rule.py | | | | | | | |
| **前提和约束** | | | 创建本地Flask测试环境 | | | | | | | |
| **测试方法** | | | 白盒测试 | | | | | | | |
| **序号** | **测试输入** | | | | **预期结果** | | | | **评价准则** | |
| 1 | {} | | | | HTTP 400  错误代码：10000001 | | | | 返回结果与预期结果相同 | |
| 2 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复'  } | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 3 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 4 | {  id: 1,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 5 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  }  数据不转化为json格式 | | | | HTTP 400  错误代码：10000041 | | | | 同上 | |
| 6 | {  id: 1,  username: ’wechatterbot’,  token: 错误令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 401  错误代码：10000044 | | | | 同上 | |
| 7 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': ' '  } | | | | HTTP 400  错误代码：10000045 | | | | 同上 | |
| 8 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'response': ' '  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000045 | | | | 同上 | |
| 9 | {  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 10 | {  id: ' ',  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000046 | | | | 同上 | |
| 11 | {  id: 'string',  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 400  错误代码：10000001 | | | | 同上 | |
| 12 | {  id: 1,  username: ’wechatterbot’,  token: 正确令牌,  'response': '对话回复',  'text': '对话内容'  } | | | | HTTP 200  返回数据：{  code:1  new\_rule: StatementRule实例  //更新对话规则  //id为1  //text为对话内容  //response为对话回复  } | | | | 同上 | |
| **备注** | 测试输入使用HTTP POST方法调用测试接口，数据格式为json。测试环境使用flask的test\_client方法搭建本地测试服务器，接口地址为admin/update\_rule  Token由代码加密当前时间生成，并非固定值，为不在测试输入栏写入重复的代码段，Token部分的输入用文字描述，具体定义如下：  正确令牌：generate\_token(key, 3600) 其中key为密钥  错误令牌：generate\_token(wrong\_key, 3600) 其中wrong\_key为错误密钥  更新对话规则包含内容较多，是返回的数据库中的StatementRule实例，且id、创建时间等属性为自动分配不能确定，为保证文档可读性，在测试输出栏用“更新对话规则”代替，其与测试相关的内容通过备注给出。  由于文档为了方便对话规则id的设置，统一在文档中写为1，在实际的单元测试中id的值不影响测试的效果，可根据实际需要进行设置 | | | | | | | | | |
| **测试人员：张延钊** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月15日** | | | |

# 4 非功能测试方案

本部分主要介绍项目的非功能测试的测试工具需求、测试需求项的列表以及具体的测试用例设计。非功能测试的测试主要与需求分析说明书之中的非功能需求部分相对应，由于1.3节测试工作内容和范围所述，本测试只关注性能需求中的高效性、可靠性部分，具体测试用例和测试见第5节。

## 4.1 测试工具需求

对于非功能测试，我们的测试目标集中在高效性中的响应时间和并发能力，这里使用以下工具进行测试。

|  |  |  |  |
| --- | --- | --- | --- |
| **测试方式** | **测试工具需求** | **选用测试工具** | **测试工具简介** |
| 黑盒测试 | 能够进行基础的压力测试 | Siege | Siege是一款回归测试和基准测试的小工具，可以对单一的URL进行用户指定数量级的压力测试。 |

## 4.2 测试需求项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **标识** | **测试需求项** | **重要性** | **测试方法** | **测试充分性** |
| TR06 | 响应时间测试 | 重要 | 黑盒测试 | 应考虑需求文档中对响应时间的需求 |
| TR07 | 并发量测试 | 重要 | 黑盒测试 | 应考虑需求文档中对并发量的需求 |
| TR08 | 可靠性测试 | 重要 | 黑盒测试 | 应考虑各种常见可能导致安全隐患的问题，诸如SQL注入和XSS攻击 |

## 4.3 测试用例

### 4.3.1 TR06对应测试用例

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 响应时间测试用例 | | | **测试用例标识** | TC0601 | | **测试需求标识** | | TR06 |
| **简要描述** | | | 模拟用户对软件获取回复接口进行单用户响应时间测试 | | | | | | | |
| **前提和约束** | | | 使用模拟HTTP请求手段进行测试 | | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | | |
| **序号** | **测试步骤** | | | | **预期结果** | | | | **评价准则** | |
| 1 | 使用Siege软件，输入软件获取回复的接口URL，输入用户测试数量为1，点击测试 | | | | 软件提示正在测试 | | | | 显示一致 | |
| 2 | 在软件返回的测试结果中分析响应时间 | | | | 响应时间小于2000ms | | | | 显示一致 | |
| **备注** |  | | | | | | | | | |
| **测试人员：王元玮** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月9日** | | | |

### 4.3.2 TR07对应测试用例

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 并发量测试用例 | | | **测试用例标识** | TC0701 | | **测试需求标识** | | TR07 |
| **简要描述** | | | 模拟大量用户对软件获取回复接口进行多用户并发量测试 | | | | | | | |
| **前提和约束** | | | 使用模拟HTTP请求手段进行测试 | | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | | |
| **序号** | **测试步骤** | | | | **预期结果** | | | | **评价准则** | |
| 1 | 使用Siege软件，输入软件获取回复的接口URL，输入用户测试数量为10，点击测试 | | | | 软件提示正在测试 | | | | 显示一致 | |
| 2 | 在软件返回的测试结果中分析响应时间 | | | | 全部并发的平均响应时间小于2000ms | | | | 显示一致 | |
| **备注** |  | | | | | | | | | |
| **测试人员：王元玮** | | | | **监测人员：吕江枫** | | | **测试时间：2020年5月9日** | | | |

### 4.3.3 TR08对应测试用例

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 可靠性测试用例 | | | **测试用例标识** | TC0801 | **测试需求标识** | | TR08 |
| **简要描述** | | | 模拟攻击者对微信端尝试SQL注入 | | | | | | |
| **前提和约束** | | | 使用微信个人号进行测试，对话内容限制在基本信息类问题 | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | |
| **序号** | **测试步骤** | | | | **预期结果** | | | **评价准则** | |
| 1 | 发送“1’ or 1=1 -- ” | | | | 系统随机以自然语言进行回复，或回复“看不懂唉”。 | | | 显示一致 | |
| **备注** |  | | | | | | | | |
| **测试人员：王元玮** | | | | **监测人员：吕江枫** | | **测试时间：2020年5月9日** | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 可靠性测试用例 | | | **测试用例标识** | | TC0802 | | **测试需求标识** | | TR08 |
| **简要描述** | | | 模拟攻击者对后端管理平台尝试SQL注入 | | | | | | | | |
| **前提和约束** | | | 能够登录WeChatterBot后端管理平台 | | | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | | | |
| **序号** | **测试步骤** | | | | | **预期结果** | | | | **评价准则** | |
| 1 | 打开登录页面，在用户名和密码的登录框中均输入“123’ or ‘1’=’1”，点击登录 | | | | | 网页提示“密码不正确” | | | | 显示一致 | |
| 2 | 输入正确的用户名密码后，进入管理端，在“text”栏中输入“1’ order by 1#”，在“response to”栏中输入“23333”，在“tags”栏中输入“哈哈+笑”。点击“添加”按钮 | | | | | 网页提示“创建成功” | | | | 显示一致 | |
| 3 | 在“对话内容”栏中输入“1’ order by 1#”，点击“查询”按钮 | | | | | 显示刚刚添加的对话，并不显示多余内容 | | | | 显示一致 | |
| 4 | 在“text”栏中输入“1’ or 1=1 --”，在“response to”栏中输入“23333”，在“tags”栏中输入“哈哈+笑”。点击“添加”按钮 | | | | | 网页提示“创建成功” | | | | 显示一致 | |
| 5 | 在“对话内容”栏中输入“1’ or 1=1 --”，点击“查询”按钮 | | | | | 显示刚刚添加的对话，并不显示多余内容 | | | | 显示一致 | |
| 6 | 在“text”栏中输入“1’ or ‘1’=’1”，在“response to”栏中输入“23333”，在“tags”栏中输入“哈哈+笑”。点击“添加”按钮 | | | | | 网页提示“创建成功” | | | | 显示一致 | |
| 7 | 在“对话内容”栏中输入“1’ or ‘1’=’1”，点击“查询”按钮 | | | | | 显示刚刚添加的对话，并不显示多余内容 | | | | 显示一致 | |
| **备注** |  | | | | | | | | | | |
| **测试人员：王元玮** | | | | **监测人员：吕江枫** | | | | **测试时间：2020年5月9日** | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 可靠性测试用例 | | | **测试用例标识** | | TC0803 | | **测试需求标识** | | TR08 |
| **简要描述** | | | 模拟攻击者对后端管理平台尝试XSS攻击 | | | | | | | | |
| **前提和约束** | | | 能够登录WeChatterBot后端管理平台 | | | | | | | | |
| **测试方法** | | | 黑盒测试 | | | | | | | | |
| **序号** | **测试步骤** | | | | | **预期结果** | | | | **评价准则** | |
| 1 | 在“text”栏中输入“哈哈哈”，在“response to”栏中输入“<scrpt>alert("GG");</script>”，在“tags”栏中输入“哈哈+笑”。点击“添加”按钮 | | | | | 网页提示“创建成功” | | | | 显示一致 | |
| 2 | 在“对话内容”栏中输入“哈哈哈”，点击“查询”按钮 | | | | | 显示刚刚添加的对话，而不是弹出警示窗口 | | | | 显示一致 | |
| **备注** |  | | | | | | | | | | |
| **测试人员：王元玮** | | | | **监测人员：吕江枫** | | | | **测试时间：2020年5月9日** | | | |

# 5 测试用例追踪

|  |  |  |
| --- | --- | --- |
| **测试需求项标识** | **测试用例标识** | **与需求规格说明书对应需求项** |
| TR01 | TC0101 | 3.3.1 |
| TC0102 | 3.3.2 |
| TR02 | TC0201 | 3.3.5 |
| TR03 | TC0301 | 3.3.6&3.3.7 |
| TC0302 |
| TC0303 |
| TR04 | TC4501 | 3.3.4&3.3.8 |
| TR05 |
| TC0502 |
| TC0503 |
| TR06 | TC0601 | 3.4.1 |
| TR07 | TC0701 |
| TR08 | TC0801 | 3.4.2 |
| TC0802 |
| TC0803 |