基于Scrapy的WebUI开发

——测试需求规格说明书

SY1906514 汪凌风 SY1906508 明　昊

SY1906513 汪丽萍 SY1906431 郑泽西

SY1806220 闫奕涛 SY1906509 邵志钧

2020/05/13

# 版本记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | 修改说明 | 参与者 | 审核人 |
| 1.0.0 | 2020.05.13 | 完成第一版 | 全体成员 | 汪凌风 |
| 1.1.0 | 2020.05.18 | 修改部分描述，增加白盒测试内容 | 全体成员 | 闫奕涛 |
| 1.1.1 | 2020.05.21 | 在3.2.2.脚本生成章节中添加测试用例“自动生成Scrapy项目正确性测试” | 汪丽萍 | 汪丽萍 |
| 1.1.2 | 2020.05.21 | 添加了实时性的测试用例 | 郑泽西 | 郑泽西 |
| 1.1.3 | 2020.05.21 | 对脚本生成黑盒测试用例中表单内容正确性作区分 | 汪丽萍 | 汪丽萍 |
| 1.2.0 | 2020.05.24 | 根据第一次互评意见进行修改 | 全体成员 | 闫奕涛 |
| 1.2.1 | 2020.05.24 | 增加表格题注和用例序号 | 郑泽西 | 郑泽西 |

注：全体成员即汪凌风，邵志钧，明昊，汪丽萍，闫奕涛，郑泽西

目　录

[一、引言 5](#_Toc41340127)

[1.1 测试目的 5](#_Toc41340128)

[1.2 测试背景 5](#_Toc41340129)

[1.3 测试范围 5](#_Toc41340130)

[1.4 术语与缩略语 5](#_Toc41340131)

[1.5 测试大纲 6](#_Toc41340132)

[1.6 参考资料 6](#_Toc41340133)

[二、测试计划 7](#_Toc41340134)

[2.1 测试方法 7](#_Toc41340135)

[2.2 测试工具 7](#_Toc41340136)

[2.3 测试分工 7](#_Toc41340137)

[2.4 测试环境 8](#_Toc41340138)

[2.5 测试用例对应关系 8](#_Toc41340139)

[三、功能需求测试 11](#_Toc41340140)

[3.1白盒测试过程 11](#_Toc41340141)

[3.1.1 管理脚本模块 11](#_Toc41340142)

[3.1.2 自动化生成脚本模块 13](#_Toc41340143)

[3.2黑盒测试过程 14](#_Toc41340144)

[3.2.1 脚本管理 14](#_Toc41340145)

[3.2.2 脚本生成 16](#_Toc41340146)

[3.2.3 词云生成 18](#_Toc41340147)

[3.2.4 选择优化 19](#_Toc41340148)

[四、非功能需求测试 24](#_Toc41340149)

[4.1 可用性 24](#_Toc41340150)

[4.2 稳定性 24](#_Toc41340151)

[4.3 交互界面用户友好 25](#_Toc41340152)

[4.4 高定制化 25](#_Toc41340153)

[4.5 运行环境兼容性 26](#_Toc41340154)

[4.6 实时性 27](#_Toc41340155)

# 一、引言

## 1.1 测试目的

本组基于Scrapy进行可视化页面开发，实现了一些关于爬虫的可视化功能。

编写此测试需求规格说明书的目的是指导接下来对于所开发系统的测试活动。测试的目的是检查开发的系统是否完成了软件需求规格说明书中所要求的需求，检查所开发的系统是否正确可靠，发现系统中的问题，最后撰写测试报告。

## 1.2 测试背景

项目名称：基于Scrapy的WebUI开发

测试人员：全体成员

开发人员：全体成员

## 1.3 测试范围

本测试依据软件需求规格说明书与设计文档，对WebUI的功能性需求和非功能性需求进行测试。

对WebUI的功能性需求描述在需求规格说明书的第四章，其测试用例在本文档第三章。对WebUI的非功能性需求描述在需求规格说明书的第四章，其测试用例在本文档第四章。

## 1.4 术语与缩略语

表1 术语定义

|  |  |
| --- | --- |
| 爬虫 | 是一种按照一定的规则，自动地抓取[万维网](https://baike.baidu.com/item/%E4%B8%87%E7%BB%B4%E7%BD%91/215515" \t "_blank)信息的程序或者脚本。 |
| Item | 项目，它定义了爬取结果的数据结构，爬取的数据会被赋值成该Item对象。 |
| Selectors | 选择器，Scrapy有自己的数据提取机制，它们被称为选择器。因为它们“选择”HTML文档的某些部分 XPath 或 CSS 表达。 |
| Scraper | 刮取器，主要作用是对Spider中间件进行管理，通过中间件完成请求、响应、数据分析等工作。 |
| Request | 请求，是URL经过Engine组件封装后的，Downloader组件即将接收的内容 |

## 1.5 测试大纲

本文共分为四个章节，分别是引言、测试计划、功能需求测试和非功能需求测试。第二章测试计划介绍了项目的测试方法、测试工具以及测试用例对应关系等；第三章功能需求测试分为了白盒测试与黑盒测试两大部分；第四章非功能需求测试介绍了系统级的测试用例。

## 1.6 参考资料

1. 邹欣.构建之法[M].北京.人民邮电出版社,2015:264-296
2. 单元测试[EB/OL].<https://www.liaoxuefeng.com/wiki/897692888725344/953546675792640>
3. 软件工程测试方法[EB/OL]. <https://blog.csdn.net/fightfaith/article/details/47858101>

# 二、测试计划

## 2.1 测试方法

对于功能测试，我们选择同时应用黑盒测试与白盒测试的方式进行检查，白盒测试是一种测试用例设计方法，盒子指的是被测试的软件，白盒指的是盒子是可视的，即清楚盒子内部的东西以及里面是如何运作的。"白盒"法全面了解程序内部逻辑结构、对所有逻辑路径进行测试；黑盒测试又称为功能测试，主要检测软件的每一个功能是否能够正常使用。在测试过程中，将程序看成不能打开的黑盒子，不考虑程序内部结构和特性的基础上通过程序接口进行测试，检查程序功能是否按照设计需求以及说明书的规定能够正常打开使用。我们通过结合两者的方式来检查模块程序内部逻辑结构与运行情况和系统功能的正确性与容错性。

## 2.2 测试工具

对于白盒测试，测试过程的代码撰写和用例测试主要通过Python的unittest包来实现，模块的一些测试结果通过Pycharm的coverage功能进行分析，对于黑盒测试，因为本项目将部署在云端服务器上，测试时通过URL即可访问基于Scrapy的WebUI页面，所以选择在页面上人工按照测试需求和用例完成全部测试工作。

## 2.3 测试分工

表2 测试分工

|  |  |  |
| --- | --- | --- |
| **成员** | **主要职责** | **测试用例** |
| 明昊 | 完成管理脚本功能需求测试，并对WebUI进行总体设计和完善 | 管理脚本 |
| 汪丽萍 | 完成生成脚本功能需求测试，并撰写对应测试文档 | 生成脚本 |
| 郑泽西 | 完成对应功能需求测试，并撰写对应测试文档 | 暂无 |
| 闫奕涛 | 完成生成词云功能需求测试，负责测试需求规格说明书的整体框架，并撰写对应测试文档 | 生成词云 |
| 邵志钧 | 完成选择优化功能需求测试，并撰写对应测试文档 | 选择优化 |
| 汪凌风 | 完成去重优化功能需求测试，并撰写对应测试文档 | 去重优化 |

## 2.4 测试环境

Chrome 7.0版本以上，Windows自带edge浏览器等。

## 2.5 测试用例对应关系

表3 测试用例对应关系

|  |  |  |  |
| --- | --- | --- | --- |
| **功能需求** | **需求文档对应章节** | **测试用例** | **测试点** |
| 管理脚本 | 4.3.1 | 运行脚本TC102 | 核对是否能够上传项目 |
| 核对是否能够运行 |
| 查看输出TC103 | 核对是否能够查看到日志输出 |
| 下载输出TC104 | 核对是否能够下载文件 |
| 管理记录TC105 | 核对是否能够删除项目 |
| 管理脚本模块单元测试TC101 | 能否按照测试数据的顺序正确合理地输出预期的结果 |
| 自动化生成脚本 | 4.3.2 | 用例自动生成Scrapy项目压缩文件  TC202 | 表单提交功能是否正常，Scrapy项目生成功能是否正常，文件修改功能是否正常，文件压缩功能是否正常，读取文件功能是否正常 |
| 返回自动生成的Scrapy项目压缩文件的文件名 |
| 用例下载Scrapy项目压缩文件  TC203 | 下载文件功能是否正常，下载文件是否是请求文件 |
| 返回请求的Scrapy项目压缩文件 |
| 在线编辑器代码编辑是否正常 |
| 用例自动生成Scrapy项目压缩文件测试  TC204 | 验证用户是否能通过提交表单相应信息自动生成Scrapy项目压缩文件 |
| 自动化生成脚本模块单元测试  TC201 | 能否按照测试数据的顺序正确合理地输出预期的结果 |
| 生成词云 | 4.3.3 | 生成词云  TC301 | 核对是否可以正常显示输入网页对应的词云图片 |
| 选择优化 | 4.3.4 | 随机代理IP测试  TC401 | 实时爬取代理IP功能是否正常 |
| 爬虫脚本代码优化是否正常 |
| 爬虫脚本运行是否正常 |
| 爬虫运行是否使用代理IP |
| 随机User-Agent测试  TC402 | 爬虫脚本代码优化是否正常 |
| 爬虫脚本运行是否正常 |
| 爬虫运行是否使用随机User-Agent |
| 爬取间隔测试  TC403 | 爬虫脚本代码优化是否正常 |
| 爬虫脚本运行是否正常 |
| 爬虫运行页面爬取间隔是否正确 |
| Cookie测试  TC404 | 爬虫脚本代码优化是否正常 |
| 爬虫脚本运行是否正常 |
| 爬虫运行是否按要求携带Cookie进行爬取 |
| 机器人协议测试  TC405 | 用户设置爬虫是否遵守机器人协议后，能否生成正确的脚本并正确运行 |
| 去重优化  TC406 | 爬虫是否可以选择优化后的去重算法并正确爬取 |
| 内存使用测试  TC407 | 去重优化算法能否使内存消耗降低 |

# 三、功能需求测试

按照第二章测试方法和工具的介绍，对于功能需求的测试，我们选择同时应用黑盒测试与白盒测试的方式进行检查，通过Python的unittest包和Pycharm的coverage功能来实现白盒测试过程的用例测试，通过人工访问基于Scrapy的WebUI页面来完成部分功能的黑盒测试。因此在该章节中，项目团队把功能需求的测试分成黑盒和白盒两部分来进行测试需求的制订和测试用例的撰写。

## 3.1白盒测试过程

对于白盒测试过程，我们通过Python的unittest包和Pycharm的coverage功能来实现部分模块的测试，因为脚本生成和选择优化功能集成在一起，且API合在一起，所以这两个模块的单元测试一起进行。

### 3.1.1 管理脚本模块

表4 用例101 管理脚本模块单元测试用例表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case Specification 101 | | | | | | |
| **名称** | 管理脚本模块单元测试 | | | | | |
| **模块功能** | 管理脚本的保存和运行 | | | | | |
| **模块优先级** | 高 | | | | | |
| **测试目的** | 测试API函数合法的输入参数及正确的调用顺序 | | | | | |
| **测试描述** | 在每一个模块测试中，调用所有的API接口函数。对API函数参数输入合法参数值，并且以正确的顺序调用，打印输出函数返回值。 | | | | | |
| **前提和约束** | Redis与Scrapyd正常运行 | | | | | |
| **测试用例** | **序号** | **输入数据** | **预期输出** | **实际输出** | **是否通过** | **备注** |
| 1 | File: ../DistributedSpider-master.zip,  Project: DistributedSpiders,  Spider: Dis\_Douban | success |  |  | 上传项目 |
| 2 | Name: DistributedSpiders, | success |  |  | 上传项目 |
| 3 | Name: DistributedSpiders, | such project is running or pending |  |  | 上传项目 |
| 4 | Name：dis | no such project exists |  |  | 上传项目 |
| 5 | Name: DistributedSpiders | Fail,  such project is running or pending |  |  | 运行项目 |
| 6 | Name: DistributedSpiders | Success |  |  | 运行项目 |
| 7 | Name：etstdemo | such project not exist |  |  | 运行项目 |
| 8 | Name: DistributedSpiders | Success |  |  | 删除项目 |
| 9 | Name: DistributedSpiders | Success |  |  | 停止项目 |
| 10 | Name：etstdemo | such project not exist |  |  | 停止项目 |
| 11 | Name: DistributedSpiders | Success |  |  | 删除项目 |
| 12 | 无 | False |  |  | 查看所有项目 |
| **覆盖率** | C:\Users\minghao\AppData\Local\Temp\WeChat Files\0283bad1dfe745d939f01af2110dffd.png | | | | | |
| **备注** |  | | | | | |

### 3.1.2 自动化生成脚本模块

表5 用例201 自动化生成脚本模块单元测试用例表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case Specification 201 | | | | | | |
| **名称** | 自动化生成脚本模块单元测试 | | | | | |
| **模块功能** | 自动生成爬虫脚本并压缩项目 | | | | | |
| **模块优先级** | 无 | | | | | |
| **测试目的** | 测试API函数合法的输入参数及正确的调用顺序 | | | | | |
| **测试描述** | 在每一个模块测试中，调用所有的API接口函数。对API函数参数输入合法参数值，并且以正确的顺序调用，打印输出函数返回值。 | | | | | |
| **前提和约束** | 无 | | | | | |
| **测试用例** | **序号** | **输入数据** | **预期输出** | **实际输出** | **是否通过** | **备注** |
| 1 | File: ../ test.json | success |  |  |  |
| 2 | 无 | False |  |  |  |
| **覆盖率** |  | | | | | |
| **备注** |  | | | | | |

## 3.2黑盒测试过程

对于黑盒测试过程，我们通过人工访问基于Scrapy的WebUI页面来完成部分功能的测试，按照不同模块的功能与需求的对应关系，我们按照2.5的表格制定了一下测试用例。

### 3.2.1 脚本管理

表6 用例102 运行脚本测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 102 | | | | |
| **名称** | 运行脚本 | | | |
| **简要描述** | 本测试验证用户是否能运行脚本 | | | |
| **前提和约束** | 系统正常运行，提交脚本能在服务器上正常运行 | | | |
| **测试步骤** | **1** | 用户点击选择脚本上传项目，填写项目名与脚本名 | | |
| **2** | 用户点击上传并确定 | | |
| **3** | 用户选中单个项目 | | |
| **4** | 用户点击运行 | | |
| **预期结果** | 一段时间后界面显示项目运行成功 | | |
| **评价准则** | 项目的状态是否变为running或者pending | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤2** | | 项目名，脚本名，项目压缩包缺少至少一个选择上传 | 界面提示请填写必要参数，并且清空界面 |
| **测试步骤4** | | 用户没有选择项目点击运行 | 界面提示请选择项目 |

表7 用例103 查看输出测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 103 | | | | |
| **名称** | 查看输出 | | | |
| **简要描述** | 本测试验证用户是否能查看日志输出 | | | |
| **前提和约束** | 用户打开WebUI，项目为finished | | | |
| **测试步骤** | **1** | 用户选择项目 | | |
| **2** | 用户点击查看输出 | | |
| **3** | 界面弹出小框显示输出 | | |
| **预期结果** | 在弹出小框内正常显示日志输出 | | |
| **评价准则** | 用户是否可以看到日志输出运行脚本 | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤2** | | 用户没有选择项目点击运行 | 界面提示请选择项目 |
| **测试步骤3** | | 无错误操作，但是项目没有运行输出 | 界面提示输出文件不存在 |

表8 用例104 下载输出测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Specification 104 | | | |
| **名称** | 下载输出 | | |
| **简要描述** | 本测试验证用户是否能下载输出 | | |
| **前提和约束** | 用户选择脚本查看输出，进入脚本输出界面 | | |
| **测试步骤** | **1** | 用户点击下载输出按钮 | |
| **预期结果** | 浏览器开始下载文件 | |
| **评价准则** | 下载文件内容是否与看到的输出结果一致 | | |
| **异常情况** | **对应步骤** | **错误操作** | **描述** |
| 无 | 在界面提示输出文件不存在的情况下点击下载输出 | 界面跳转输出文件不存在界面 |

表9 用例105 管理记录测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Specification 105 | | | |
| **名称** | 管理记录 | | |
| **简要描述** | 本测试验证系统是否能够保存用户的过往爬取信息 | | |
| **前提和约束** | 用户在WebUI内运行过项目 | | |
| **测试步骤** | **1** | 用户重新打开WebUI | |
| **预期结果** | 用户可以看到自己之前上传的项目并且查看输出 | |
| **评价准则** | 是否可以看到之前上传的项目 | | |
| **异常情况** | **对应步骤** | **错误操作** | **描述** |
| 无 | 无 | 无 |

### 3.2.2 脚本生成

表10 用例202 自动生成Scrapy项目压缩文件测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 202 | | | | |
| **名称** | 自动生成Scrapy项目压缩文件测试 | | | |
| **简要描述** | 本测试验证用户是否能通过提交表单相应信息自动生成Scrapy项目压缩文件 | | | |
| **前提和约束** | 用户运行系统 | | | |
| **测试步骤** | **1** | 选择“自动生成脚本”子功能 | | |
| **2** | 填写项目名，网址，item和对应xpath等表单内容 | | |
| **3** | 选择“提交”按钮进行提交 | | |
| **预期结果** | 表格显示自动生成的Scrapy项目压缩文件的文件名 | | |
| **评价准则** | 实际结果与预期结果一致 | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤2** | | 必填表单项空着 | 提示该表单项不能为空 |
| 网址格式错误 | 提示网址格式错误 |
| 项目名不到或超过规定字符 | 提示项目名规定字符长度 |
| 项目名以数字开头 | 提示项目名不能以数字开头 |
| **测试步骤3** | | 提交时必填表单项为空 | 提示该表单项不能为空 |
| 提交时网址格式错误 | 提示网址格式错误 |
| 提交时项目名不到或超过规定字符 | 提示项目名规定字符长度 |
| 提交时项目名以数字开头 | 提示项目名不能以数字开头 |

表11 用例203 下载Scrapy项目压缩文件测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Specification 203 | | | |
| **名称** | 下载Scrapy项目压缩文件测试 | | |
| **简要描述** | 本测试验证用户是否能下载自动生成的Scrapy项目压缩文件 | | |
| **前提和约束** | 用户自动生成的Scrapy项目压缩文件名显示在表格上 | | |
| **测试步骤** | **1** | 点击需要下载的Scrapy项目压缩文件的“下载”按钮 | |
| **预期结果** | 浏览器下载相应Scrapy项目压缩文件 | |
| **评价准则** | 实际结果与预期结果一致 | | |
| **异常情况** | **对应步骤** | **错误操作** | **描述** |
| 无 | 无 | 无 |

表12 用例204 自动生成Scrapy项目正确性测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Specification 204 | | | |
| **名称** | 自动生成Scrapy项目正确性测试 | | |
| **简要描述** | 本测试验证用户自动生成的Scrapy项目是否能正常运行 | | |
| **前提和约束** | 用户下载正确填写表单所生成的Scrapy项目压缩文件 | | |
| **测试步骤** | **1** | 解压Scrapy项目压缩文件 | |
| **2** | 查看Scrapy项目压缩文件中是否有scrapy.cfg文件，myspider.py文件，items.py文件，middlewares.py文件，pipelines.py文件，settings.py文件 | |
| **3** | 查看myspider.py文件，items.py文件，middlewares.py文件，pipelines.py文件，settings.py文件中是否有表单中填写的对应内容 | |
| **4** | 使用scrapy shell的相应命令运行生成的爬虫，查看是否能正常运行 | |
| **预期结果** | 测试步骤2：相应文件都存在  测试步骤3：相应文件中的内容都为表单中填写的数据  测试步骤4：爬虫正常运行 | |
| **评价准则** | 实际结果与预期结果一致 | | |
| **异常情况** | **对应步骤** | **错误操作** | **描述** |
| 无 | 无 | 无 |

### 3.2.3 词云生成

表13 用例301 生成词云测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 301 | | | | |
| **名称** | 生成词云 | | | |
| **简要描述** | 用户在选择框中选取URL，该用例能够成功生成词云图 | | | |
| **前提和约束** | 用户进入生成词云页面 | | | |
| **测试步骤** | **1** | 在选择框中选取URL | | |
| **2** | 点击“生成词云” | | |
| **3** | 运行脚本 | | |
| **预期结果** | 脚本正确运行并成功生成词云 | | |
| **评价准则** | 词云生成图片正确且完整，运行结果与预期结果一致 | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤2** | | 爬取对应URL失败 | 提示“爬取失败” |

### 3.2.4 选择优化

表14 用例401 随机代理IP测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 401 | | | | |
| **名称** | 随机代理IP测试 | | | |
| **简要描述** | 本测试验证用户选择随机代理IP优化选项后，能否生成正确的脚本。 | | | |
| **前提和约束** | 用户进入自动化生成脚本页面 | | | |
| **测试步骤** | **1** | 正确填写爬取网站及xpath路径信息后，选择“ippool”优化选项并点击生成脚本 | | |
| **2** | 用户下载生成后的项目压缩包 | | |
| **3** | 用户解压缩并查看目录“项目名/项目名/settings.py”文件中是否包含随机代理IP设置代码 | | |
| **预期结果** | settings.py文件中正确包含随机代理IP设置代码 | | |
| **评价准则** | 实际结果与预期结果一致 | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤1** | | 错误填写网站信息 | 提示网站地址格式错误 |

表14 用例402 随机User-Agent测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 402 | | | | |
| **名称** | 随机User-Agent测试 | | | |
| **简要描述** | 本测试验证用户选择随机User-Agent优化选项后，能否生成正确的脚本。 | | | |
| **前提和约束** | 用户进入自动化生成脚本页面 | | | |
| **测试步骤** | **1** | 正确填写爬取网站及xpath路径信息后，选择“useragent”优化选项并点击生成脚本 | | |
| **2** | 用户下载生成后的项目压缩包 | | |
| **3** | 用户解压缩并查看目录“项目名/项目名/settings.py”以及“项目名/项目名/middlewares.py”文件中是否包含随机User-Agent设置代码 | | |
| **预期结果** | settings.py及middlewares.py文件中正确包含随机代理IP设置代码 | | |
| **评价准则** | 实际结果与预期结果一致 | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤1** | | 错误填写网站信息 | 提示网站地址格式错误 |

表15 用例403 爬取间隔测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 403 | | | | |
| **名称** | 爬取间隔测试 | | | |
| **简要描述** | 本测试验证用户设置爬取间隔后，能否生成正确的脚本。 | | | |
| **前提和约束** | 用户进入自动化生成脚本页面 | | | |
| **测试步骤** | **1** | 正确填写爬取网站及xpath路径信息后，选择“ippool”优化选项并填写意向爬取间隔，之后点击生成脚本 | | |
| **2** | 用户下载生成后的项目压缩包 | | |
| **3** | 用户解压缩并查看目录“项目名/项目名/settings.py”文件中是否有生成爬取间隔正确的设置代码 | | |
| **预期结果** | settings.py文件中包含正确的爬取间隔设置代码 | | |
| **评价准则** | 实际结果与预期结果一致 | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤1** | | 错误填写网站信息 | 提示网站地址格式错误 |

表16 用例404 Cookie测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 404 | | | | |
| **名称** | Cookie测试 | | | |
| **简要描述** | 本测试验证用户设置是否携带Cookie后，能否生成正确的脚本并正确运行。 | | | |
| **前提和约束** | 用户进入自动化生成脚本页面 | | | |
| **测试步骤** | **1** | 正确填写爬取网站及xpath路径信息后，选择“cookie”优化选项，之后点击生成脚本 | | |
| **2** | 用户下载生成后的项目压缩包 | | |
| **3** | 用户解压缩并查看目录“项目名/项目名/settings.py”文件中是否有按照选择生成是否携带Cookie的设置代码 | | |
| **预期结果** | settings.py文件中包含正确的携带cookie设置代码 | | |
| **评价准则** | 实际结果与预期结果一致 | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤1** | | 错误填写网站信息 | 提示网站地址格式错误 |

表17 用例405 机器人协议测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 405 | | | | |
| **名称** | 机器人协议测试 | | | |
| **简要描述** | 本测试验证用户设置爬虫是否遵守机器人协议后，能否生成正确的脚本并正确运行。 | | | |
| **前提和约束** | 用户进入自动化生成脚本页面 | | | |
| **测试步骤** | **1** | 正确填写爬取网站及xpath路径信息后，选择“robot”优化选项，之后点击生成脚本 | | |
| **2** | 用户下载生成后的项目压缩包 | | |
| **3** | 用户解压缩并查看目录“项目名/项目名/settings.py”文件中是否有按照选择生成是否遵守机器人协议的设置代码 | | |
| **预期结果** | settings.py文件中包含正确的遵守机器人协议设置代码 | | |
| **评价准则** | 实际结果与预期结果一致 | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤1** | | 错误填写网站信息 | 提示网站地址格式错误 |

表18 用例406 选择去重算法优化测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 406 | | | | |
| **名称** | 选择去重算法优化 | | | |
| **简要描述** | 本测试验证用户选择去重优化后能否正常运行 | | | |
| **前提和约束** | 用户运行一个能够正确运行的爬虫项目 | | | |
| **测试步骤** | **1** | 点击选择去重算法优化 | | |
| **2** | 点击确定运行优化后的项目 | | |
| **预期结果** | 爬虫项目与未选择优化时最终结果一致。 | | |
| **评价准则** | 实际结果与预期结果一致 | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤1** | | 用户URL数目过多（亿级） | 此时算法会出现误判，但可以修改配置文件中的参数，使算法能够适应URL数目 |

表19 用例407 内存使用测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Specification 407 | | | | |
| **名称** | 选择去重算法优化后的内存使用测试 | | | |
| **简要描述** | 本测试验证去重优化算法能否使内存消耗降低，但限于条件，只能将算法单独作为demo测试 | | | |
| **前提和约束** | 能够正确使用bloom-filter-buaasee 这个python包 | | | |
| **测试步骤** | **1** | 生成100万个随机URL序列 | | |
| **2** | 使用scrapy自带的过滤器去重，记录内存使用量 | | |
| **3** | 使用bloom-filter-buaasee包的去重算法进行测试记录内存使用量 | | |
| **预期结果** | 框架自带的内存使用量随时间线性增长，最终达到8G，后者内存用量一直为128M左右 | | |
| **评价准则** | 实际结果与预期结果一致 | | | |
| **异常情况** | **对应步骤** | | **错误操作** | **描述** |
| **测试步骤3** | | 生成的URL是按数字递增形式 | 在本算法中，由于具体hash函数，数字递增形式的url会被误判，但实际应用时，scrapy会将url先生成40位的fingerprint，所以不存在规律问题。 |

# 四、非功能需求测试

## 4.1 可用性

表20 用例001 可用性测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Specification 001 | | | |
| **名称** | 可用性测试 | | |
| **简要描述** | 本测试验证应用运行时，管理脚本、自动化生成脚本、生成词云、在线编辑调试和选择优化四大主要功能是否能正常运行 | | |
| **前提和约束** | 用户打开ScrapyWebUI网址 | | |
| **测试步骤** | **1** | 用户进行各项功能的使用 | |
| **2** | 测试各项功能是否正常运行，是否存在功能失效的情况 | |
| **预期结果** | 各项功能正常运行，不存在功能失效的情况 | |
| **评价准则** | 实际结果与预期结果一致 | | |
| **异常情况** | 无 | |  |

## 4.2 稳定性

表21 用例002 稳定性测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Specification 002 | | | |
| **名称** | 稳定性测试 | | |
| **简要描述** | 本测试验证从用户打开应用开始，2小时以内可以稳定运行无故障，出故障时，能够发出故障信息 | | |
| **前提和约束** | 用户打开ScrapyWebUI网址 | | |
| **测试步骤** | **1** | 用户进行各项功能的使用 | |
| **2** | 测试从用户打开网址开始，是否可以2小时以内稳定运行无故障，若出现故障，是否能够发出故障信息 | |
| **预期结果** | 从用户打开网址开始，2小时以内可以稳定运行无故障，出现故障时，能够发出故障信息 | |
| **评价准则** | 实际结果与预期结果一致 | | |
| **异常情况** | 无 | |  |

## 4.3 交互界面用户友好

表22 用例003 交互界面用户友好性测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Specification 003 | | | |
| **名称** | 交互界面用户友好性测试 | | |
| **简要描述** | 本测试验证在用户选择脚本进行上传时，系统对于用户上传是否有完整的指导。 | | |
| **前提和约束** | 用户启动本应用并选择脚本管理中的选择脚本功能 | | |
| **测试步骤** | **1** | 用户填写项目名 | |
| **2** | 用户填写脚本名 | |
| **3** | 用户选择本地项目压缩包进行上传 | |
| **预期结果** | 系统提示用户注意项目名与脚本名与项目代码需一致。 | |
| **评价准则** | 实际结果与预期结果一致 | | |
| **异常情况** | 无 | |  |

## 4.4 高定制化

表23 用例004 高定制化测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Specification 004 | | | |
| **名称** | 高定制化测试 | | |
| **简要描述** | 本测试验证用户可以使用本系统是否可以实现高自由度的爬虫定制，其中高定制化指用户可以使用系统对不同网站按照自己的意向，设置不同的item爬取不同的内容，同时可以使用包括随机代理IP地址、随机User-Agent在内的不同优化选项对爬虫进行定制。 | | |
| **前提和约束** | 用户进入自动化生成脚本页面 | | |
| **测试步骤** | **1** | 用户进入自动化生成脚本页面 | |
| **2** | 用户对爬虫进行定制 | |
| **3** | 用户使用定制爬虫可以爬取大部分不使用动态JS编写的网站 | |
| **预期结果** | 爬虫脚本正确生成 | |
| **评价准则** | 实际结果与预期结果一致 | | |
| **异常情况** | 无 | |  |

## 4.5 运行环境兼容性

表24 用例005 运行环境兼容性测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Specification 005 | | | |
| **名称** | 运行环境兼容性测试 | | |
| **简要描述** | 本测试主要用于测试应用分别在Linux和Windows系统上能否正确部署运行 | | |
| **前提和约束** | 用户运行系统 | | |
| **测试步骤** | **1** | 用户在Windows系统上部署应用 | |
| **2** | 用户查看应用运行情况 | |
| **3** | 用户在Linux系统上部署应用 | |
| **4** | 用户查看应用运行情况 | |
| **预期结果** | 应用在Linux和Windows系统上都能正确部署运行 | |
| **评价准则** | 实际结果与预期结果一致 | | |
| **异常情况** | 无 | |  |
| **备注** |  | | |

## 4.6 实时性

表25 用例006 实时性测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Specification 006 | | | |
| **名称** | 实时性测试 | | |
| **简要描述** | 本测试主要用于测试系统实现各个用例的响应时间 | | |
| **前提和约束** | 用户运行系统 | | |
| **测试步骤** | **1** | 用户依次测试功能需求中的黑盒测试过程中的用例 | |
| **2** | 记录每个用例完成所需的响应时间 | |
| **3** | 重复上述过程三至五次 | |
| **4** | 取响应时间的平均值作为结果 | |
| **预期结果** | 每个测试用例都能在合理的时间内给出响应 | |
| **评价准则** | 实际结果与预期结果一致 | | |
| **异常情况** | 无 | |  |
| **备注** | 本测试没有通过与不通过之分，主要为了测试每个功能的响应时间。 | | |