## 1 测试说明

### 测试方式

组间测试在与被测试组协调之后，采取在被测组环境下，参照测试需求说明书与软件需求说明书的形式进行测试。

### 测试说明

对于C组的每个测试用例，进行实地测试，按照测试用例进行输入输出，比较结果，进行测试。

## 2 测试内容

### 2.1 测试项

该部分测试了B组Blade项目中的单元测试，测试的对象和测试项如下表格：

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例名称 | 测试项 | 测试用例名称 | 测试项 |
| 爬虫有效性测试 | Spider | 前端兼容性测试 | 前端兼容性测试 |
| 布隆过滤器正确性测试 | 过滤器 | 前端跨平台测试 | 前端跨平台测试 |
| 分布式爬虫协作爬取的有效性测试 | 分布式Spider | 前端样式测试 | 前端样式测试 |
| 分布式爬虫协作爬取的效率测试 | 分布式Spider | 前端功能测试 | 前端功能测试 |
| 网站链接关系设置有效性测试 | 网站画像-链接入库 | 前端API访问 | 前端API访问 |
| 爬取效率测试 | 爬取性能 |  |  |
| 反爬与代理IP测试 | Scrapy代理ip与Scrapy反爬 |  |  |

### 2.2测试结果

1. 爬虫有效性测试

启动爬虫软件，能否正常的对目标网站进行（以北航计算机官网为例）爬取，测试结果如下图所示：

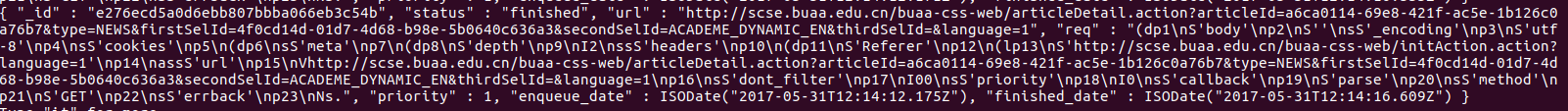
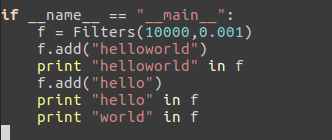


图1 爬虫有效性测试

2. 布隆过滤器正确性测试

布隆过滤器正确性测试，编写测试代码，对布隆过滤器的有效性进行测试，测试的代码以及结果如下图所示：



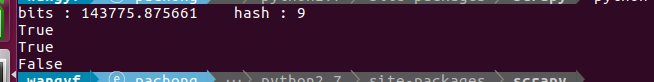
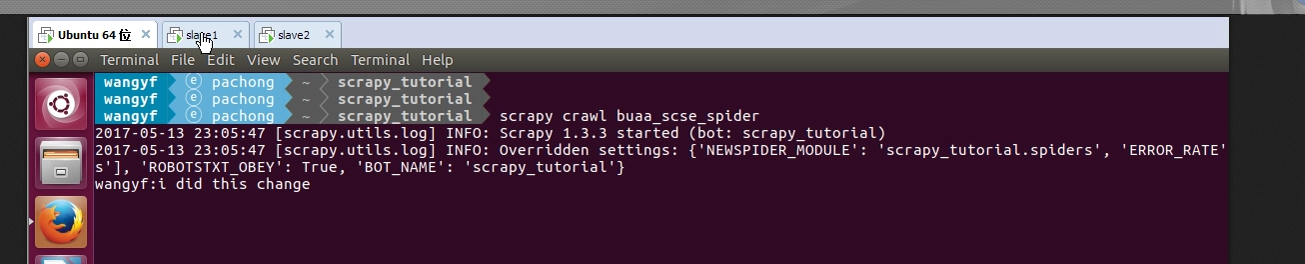


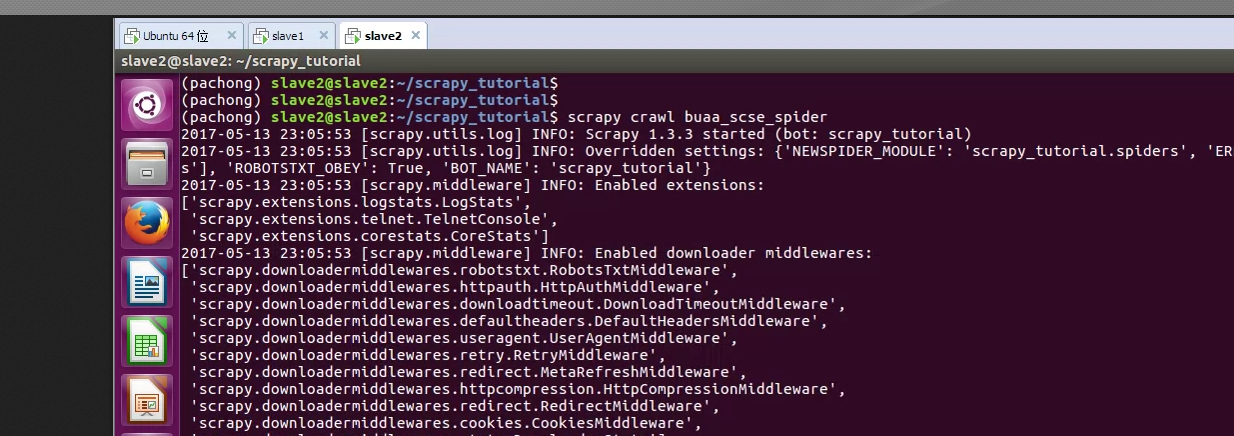
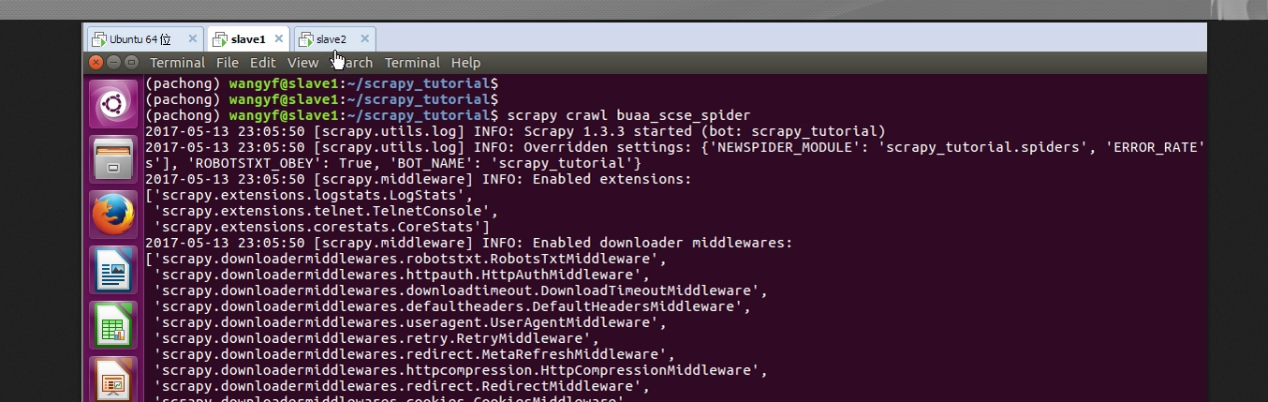
图2 布隆过滤器有效性测试

可以看到布隆过滤器可以正常的进行过滤操作。

3. 分布式爬虫协作有效性测试

同时在3台机器上开启爬虫软件，进行分布式爬取测试，对爬虫软件的分布式爬取的有效性进行测试，测试结果如下图所示：



图3 分布式爬虫协作有效性测试

4.分布式爬虫效率测试

对同一个目标网址分别进行单机爬取以及分布式爬取操作，编写脚本，记录爬取完成的时间，测试分布式爬取的效率，测试的结果截图如下所示：C:\Users\ttwl\AppData\Local\Microsoft\Windows\INetCache\Content.Word\TC_eff2.png

C:\Users\ttwl\AppData\Local\Microsoft\Windows\INetCache\Content.Word\TC_eff1.png

图4 分布式爬虫效率测试

上图是使用分布式爬虫进行爬取操作消耗的时间，下图是采用单机爬虫消耗的时间。值得注意的是，采用分布式的爬虫进行爬取操作，使用的时间反而更长。

5. 网站链接关系设置有效性测试

完成爬取工作后，打开存储爬取结果的数据库，进行链接关系设置有效性的测试，测试结果截图如下所示：

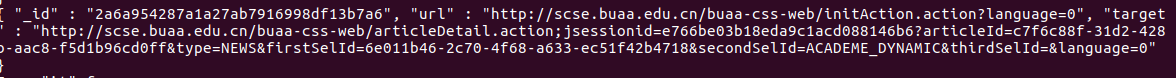


图5网站链接关系设置有效性测试

6. 反爬与代理ip测试

进行反爬与代理IP的测试，测试结果截图如下所示：

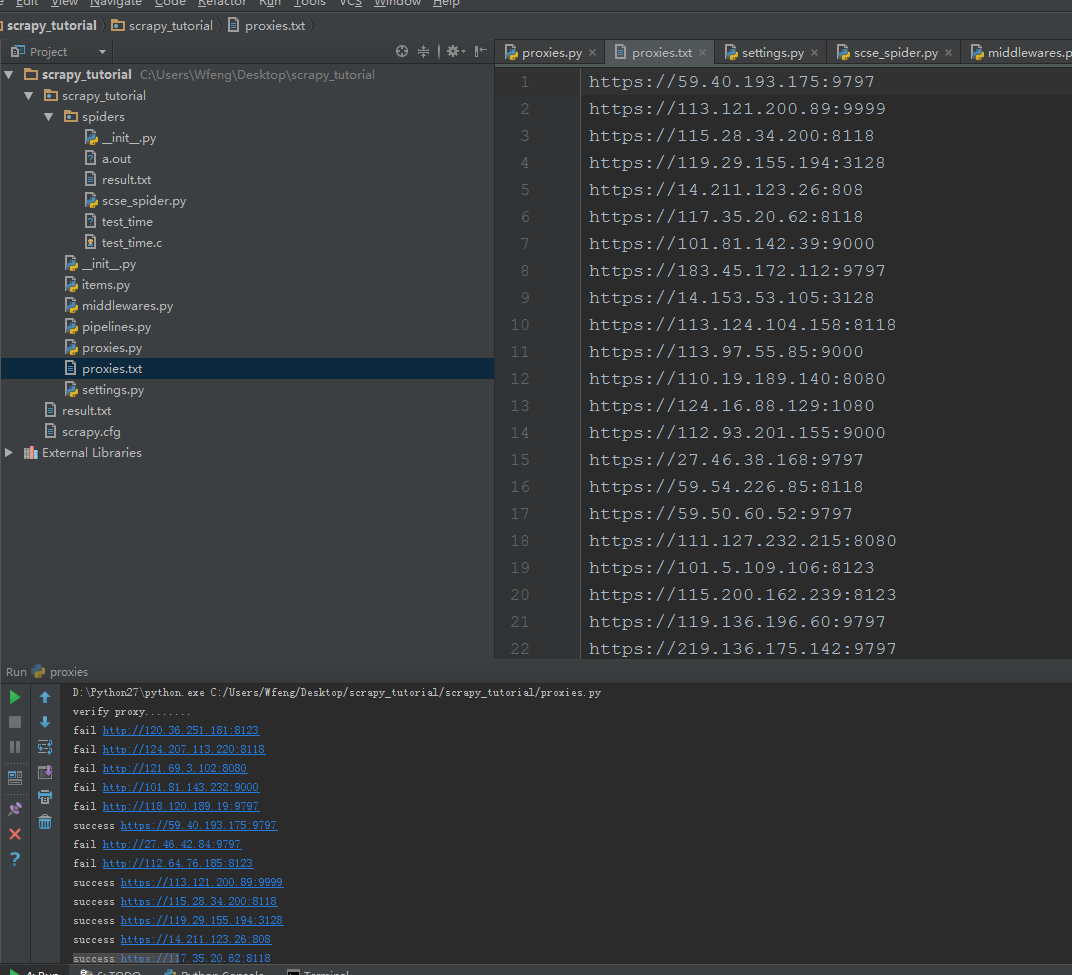


图6 反爬与代理IP测试

7. 前端兼容性测试

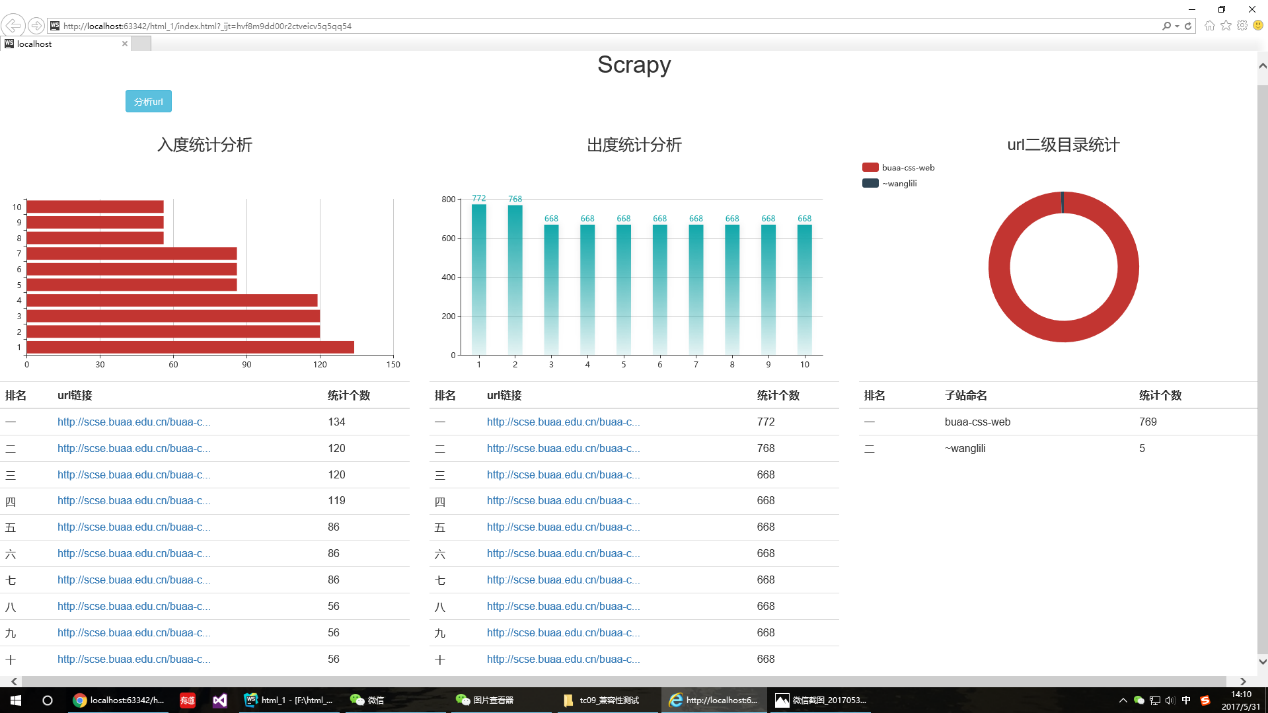
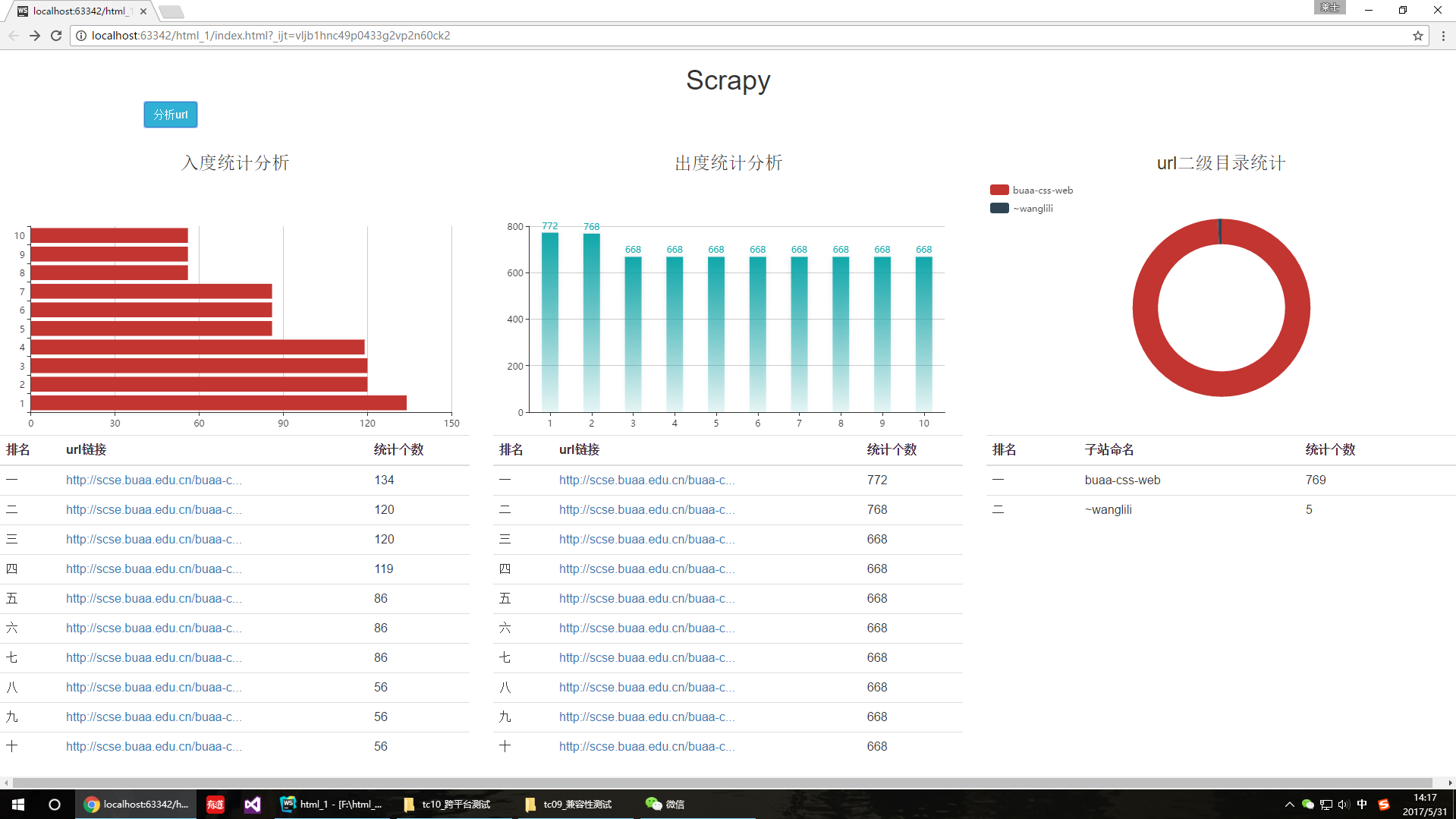
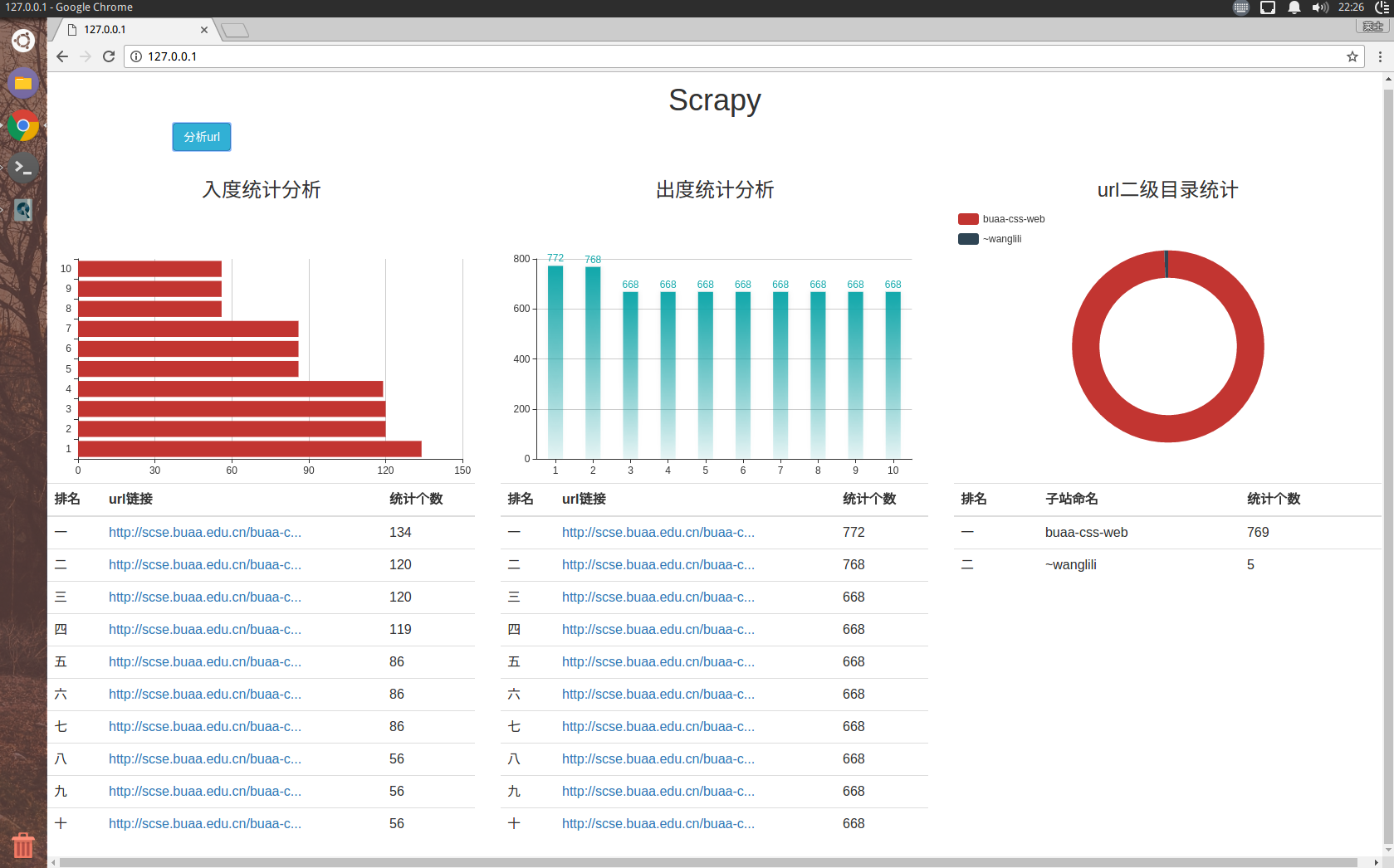
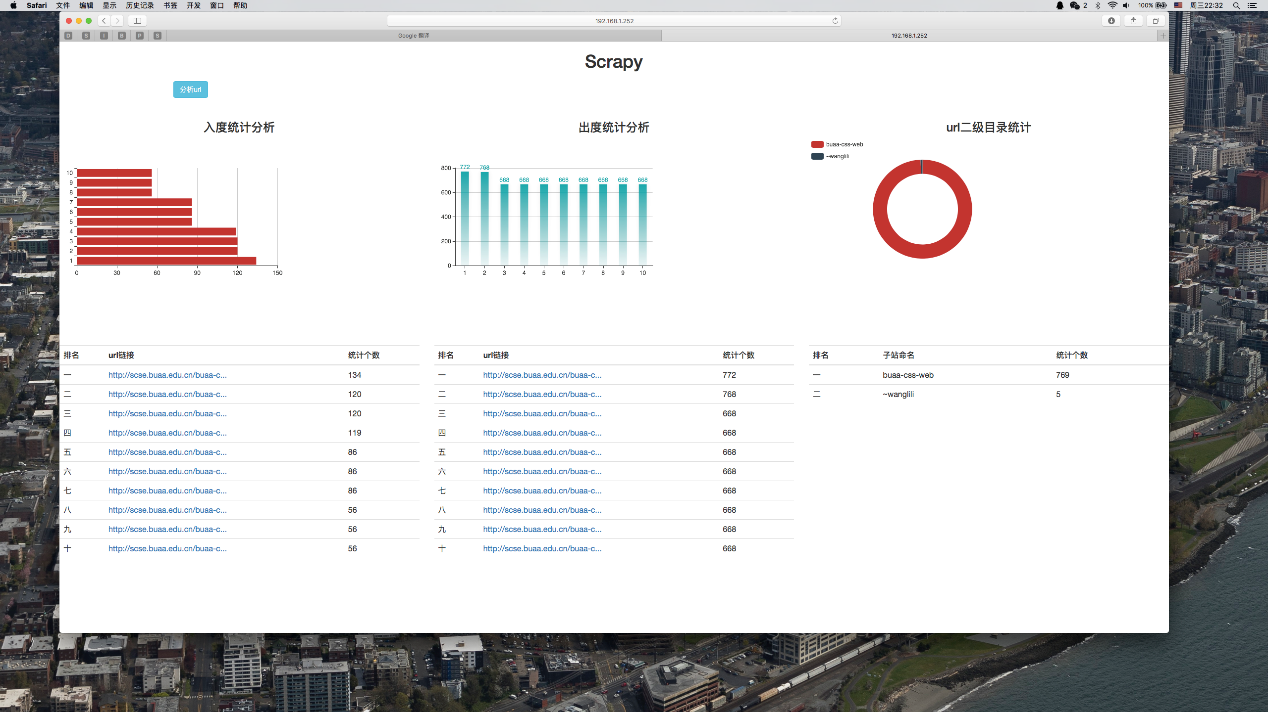
使用不同操作系统以及不同的浏览器打开前端，进行兼容性测试，测试结果如下图所示：

图7 前端兼容性测试

8.前端样式测试

测试前端的功能与样式，进行前端的样式测试，测试结果如下图所示：

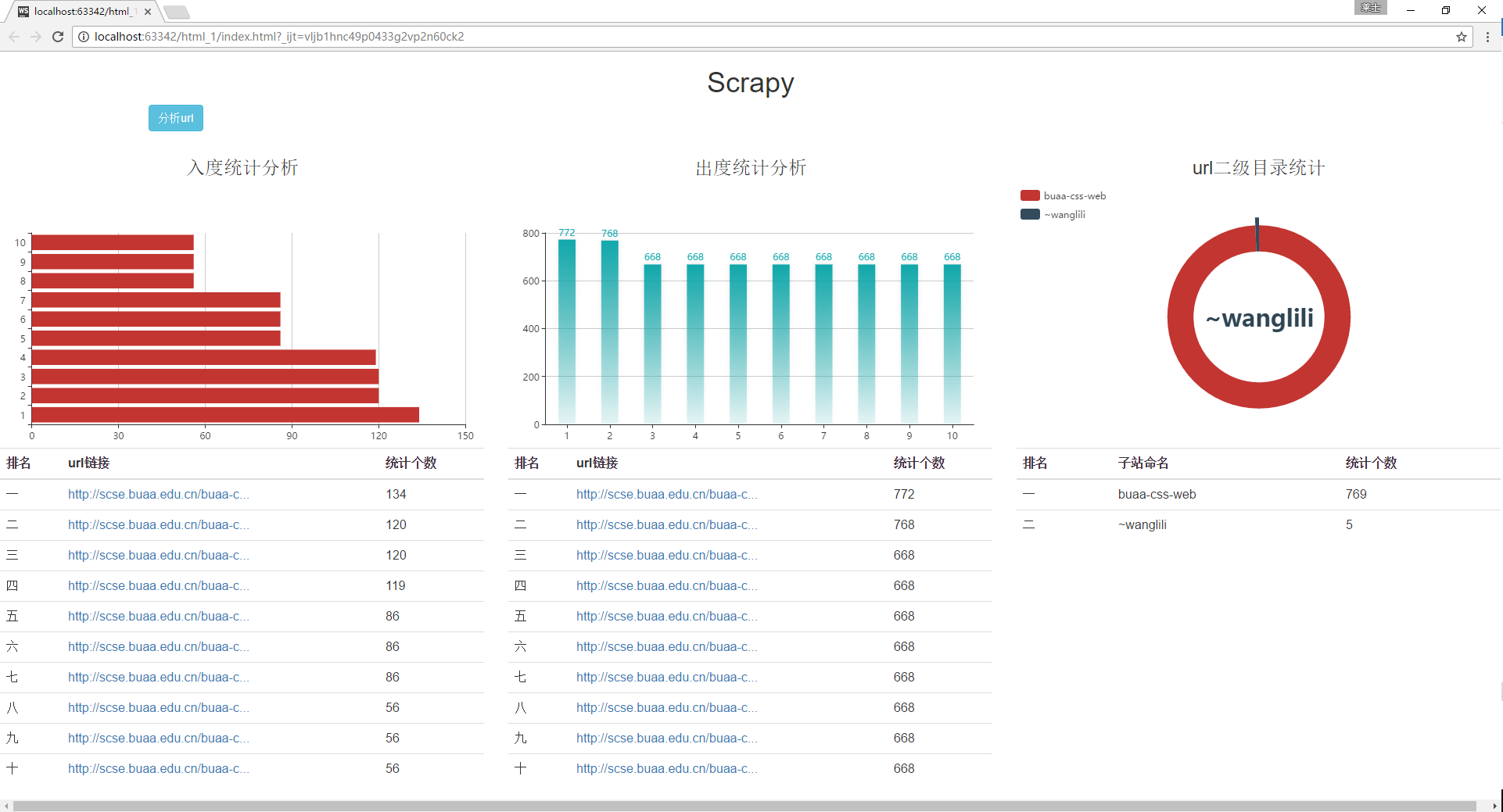
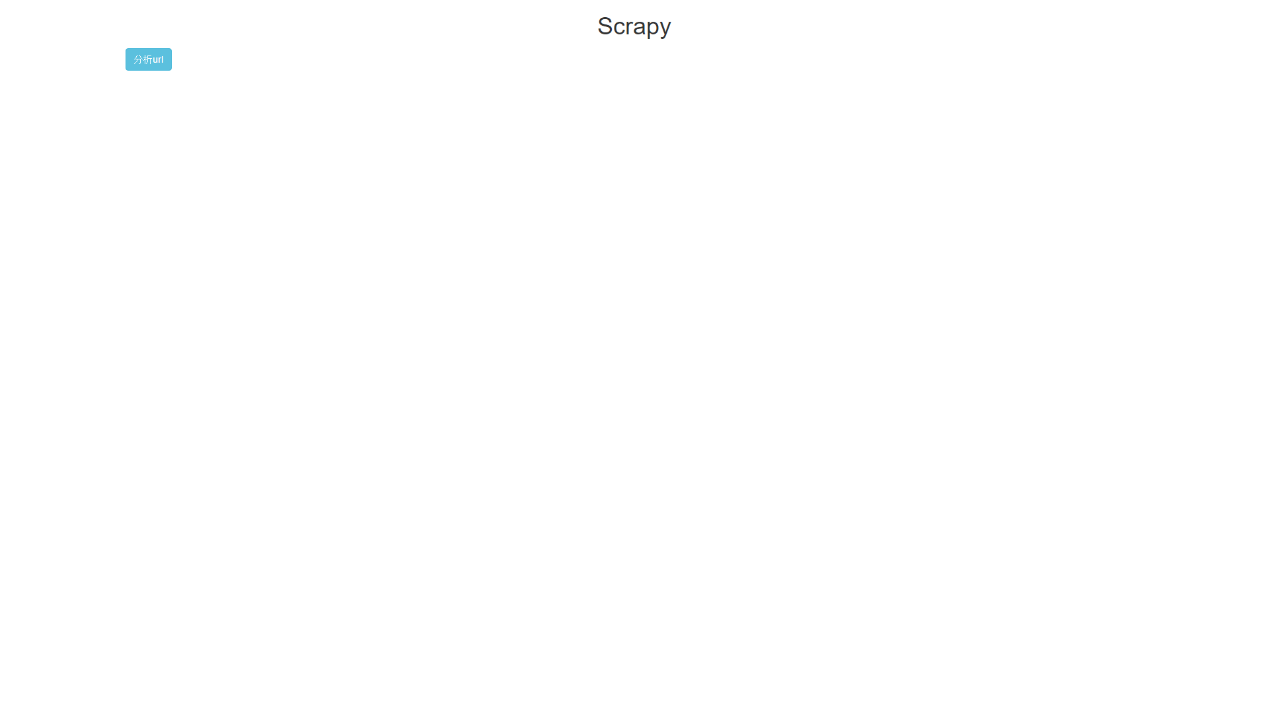


图8 前端样式测试

9. 前端功能测试

通过浏览器访问系统的前端，操作前端提供的功能，对其进行功能测试，测试结果截图如下所示：

图9.1 前端初始界面

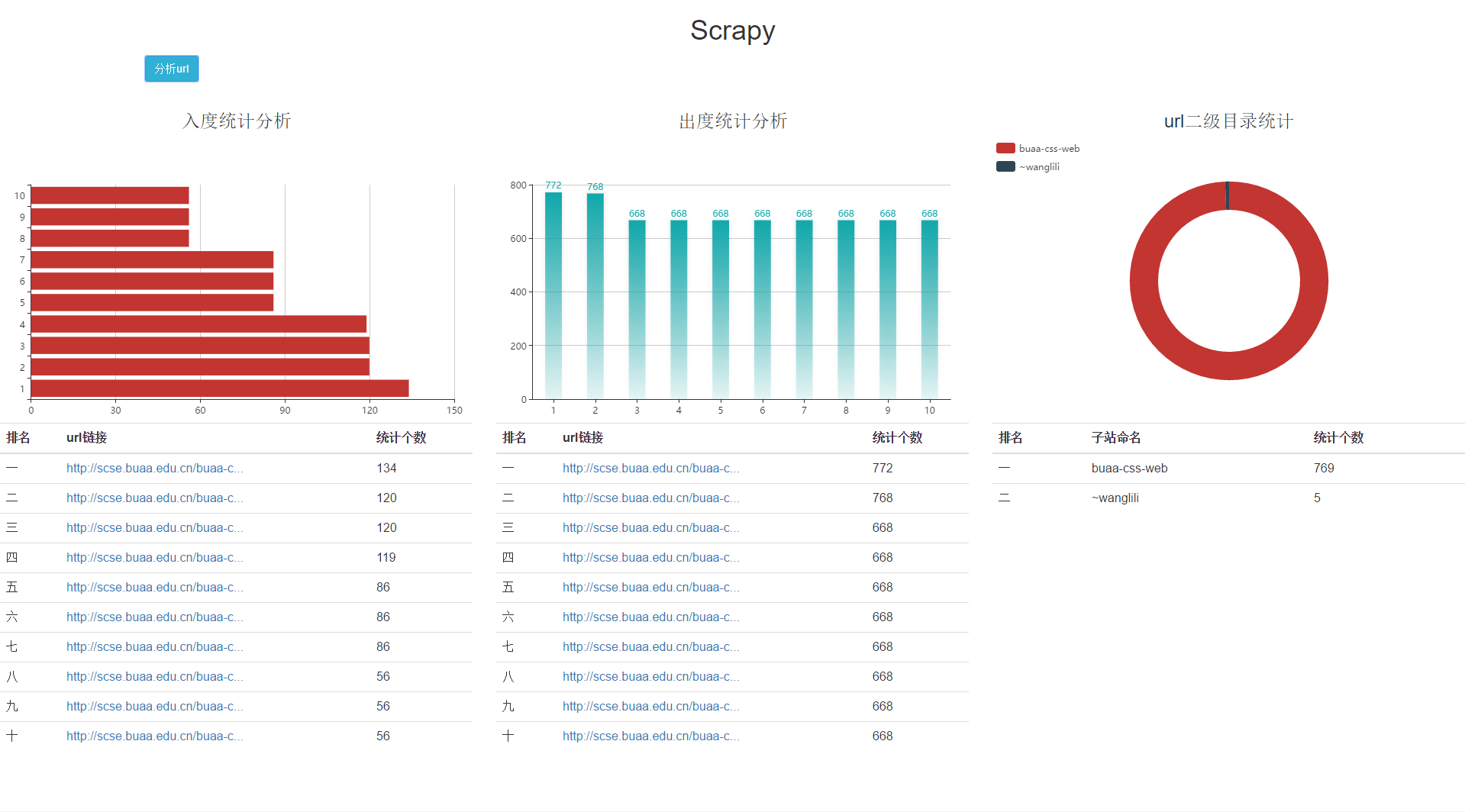
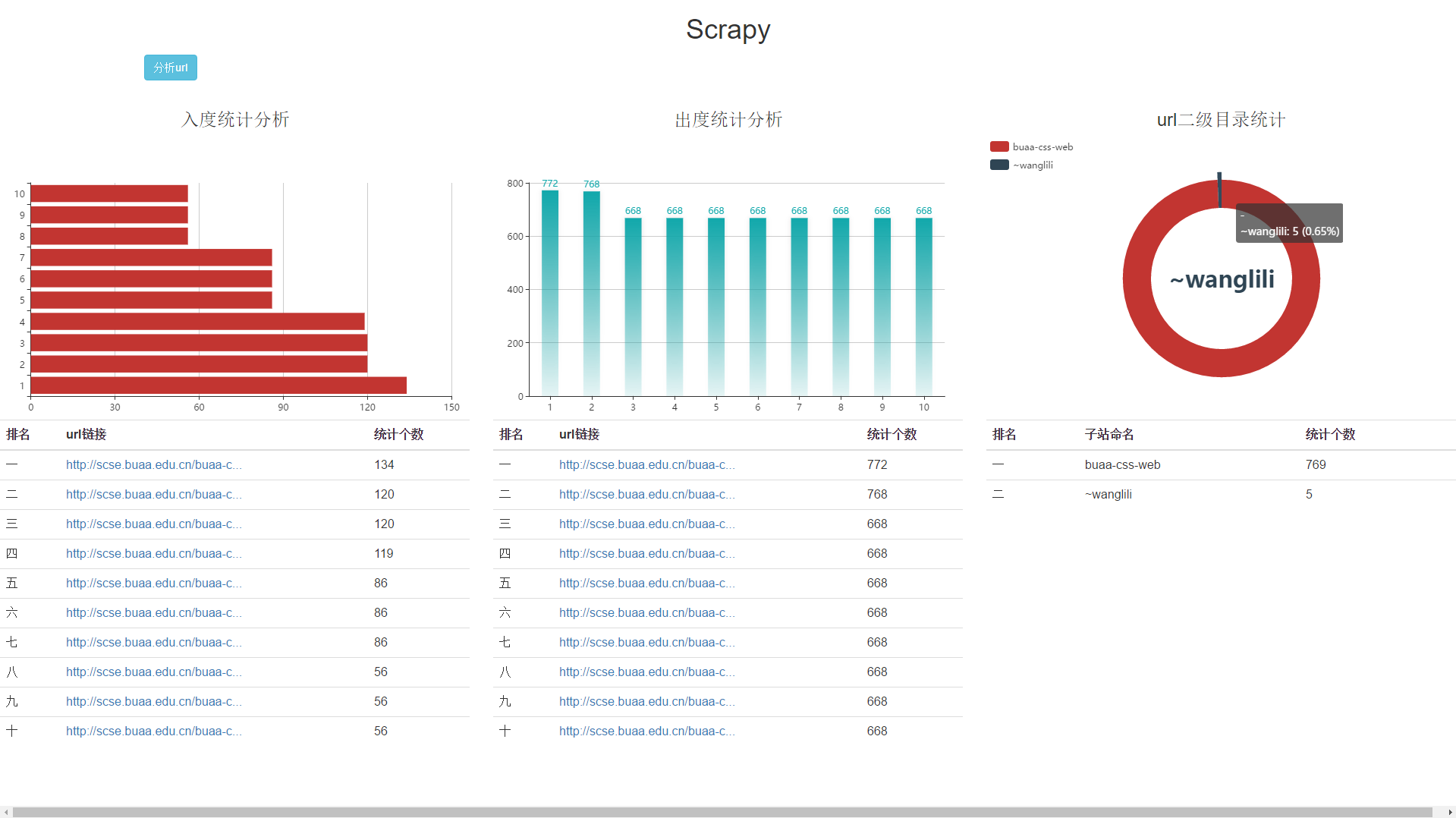
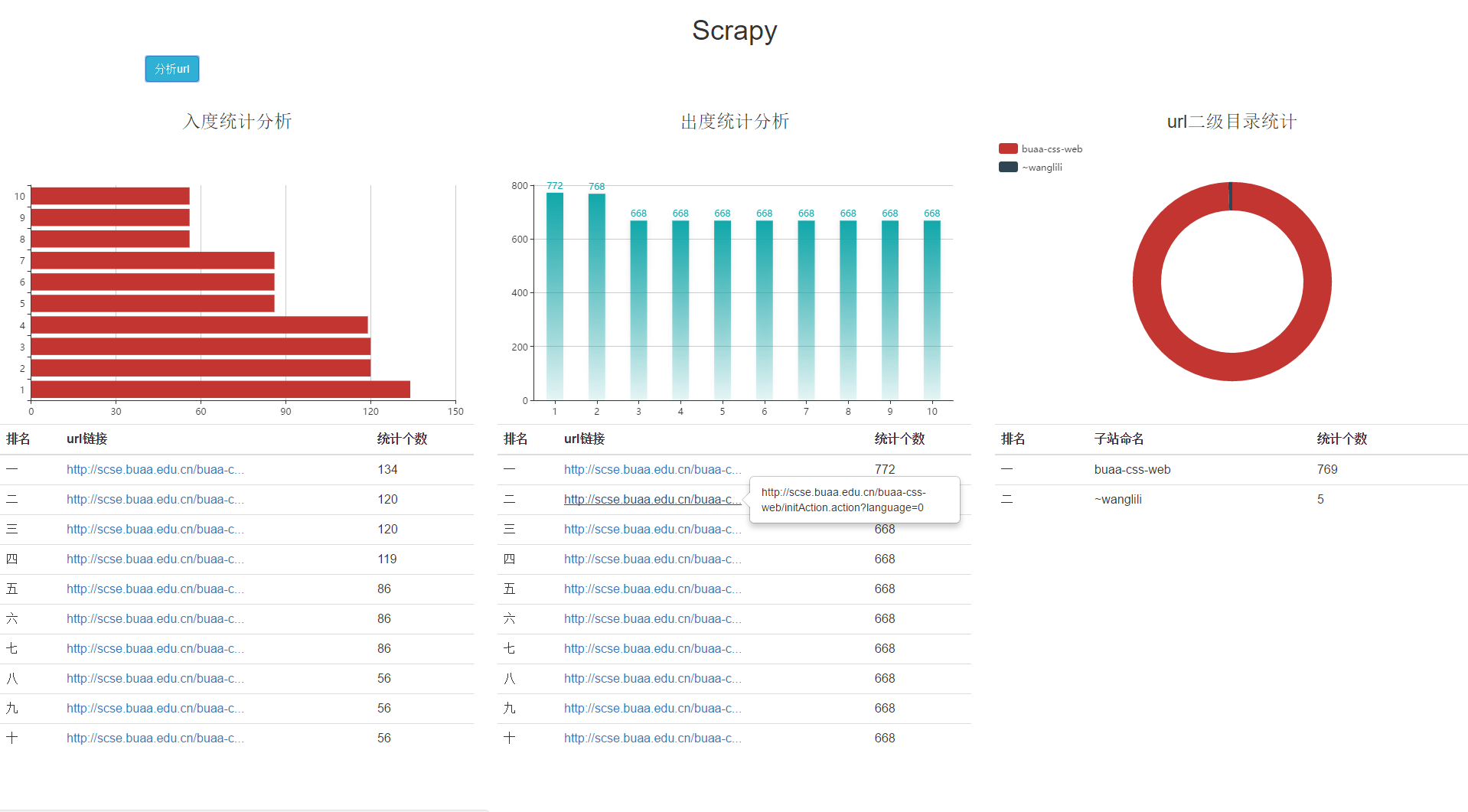
图9.1 前端按钮点击图9.3 查看图标明细图9.4 查看url明细

图9.5 url跳转功能

10. 前端API访问测试

进行前端的url访问测试，使用正确的与错误的以及空的url进行测试，测试结果截图如下图所示：

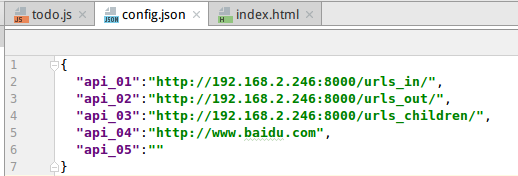
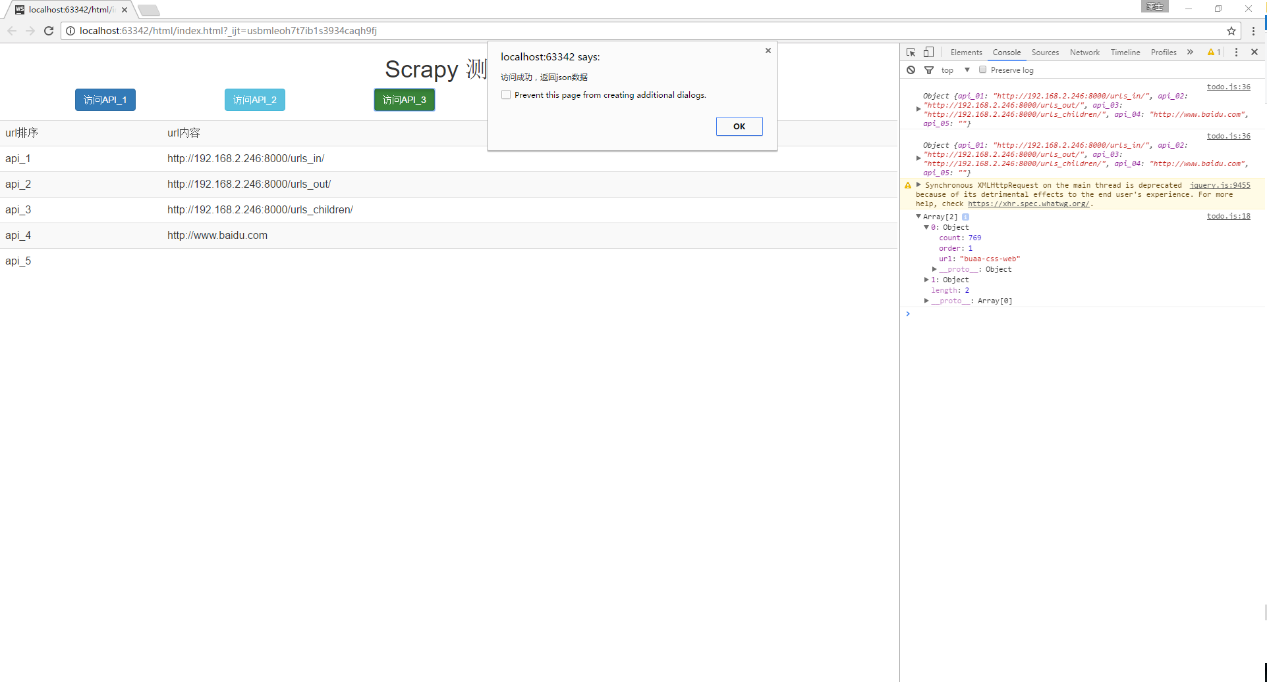
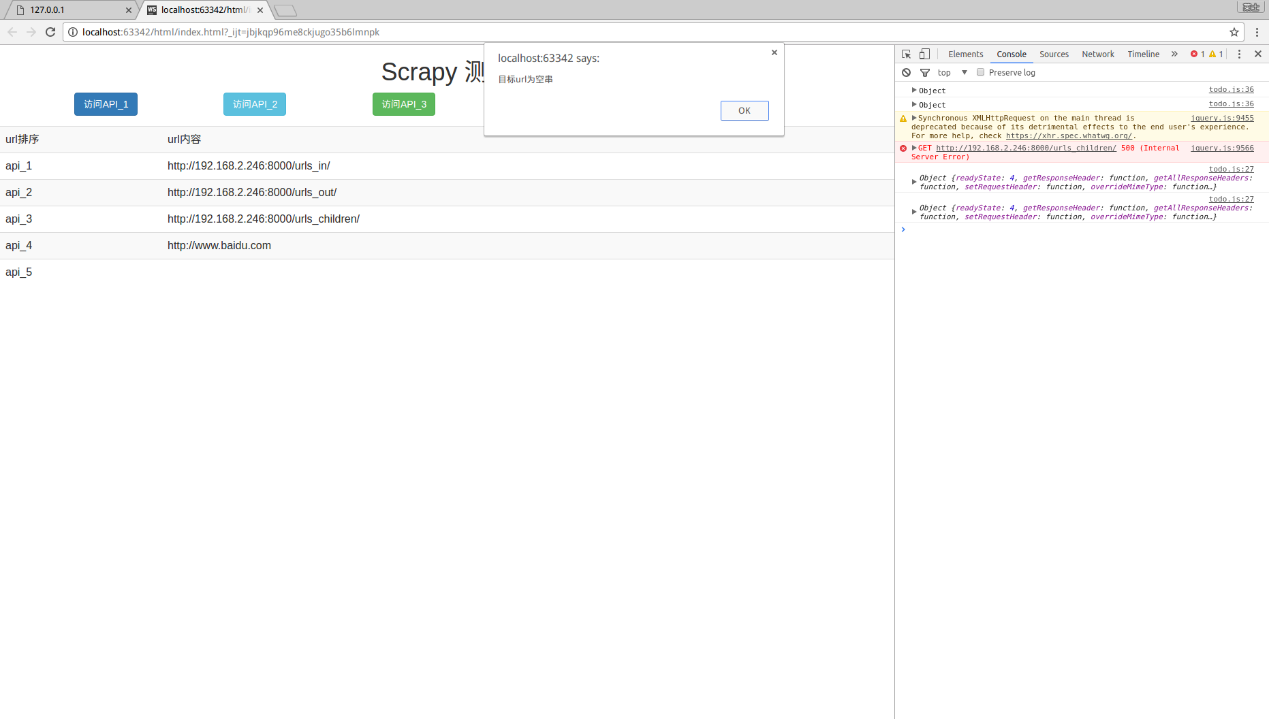
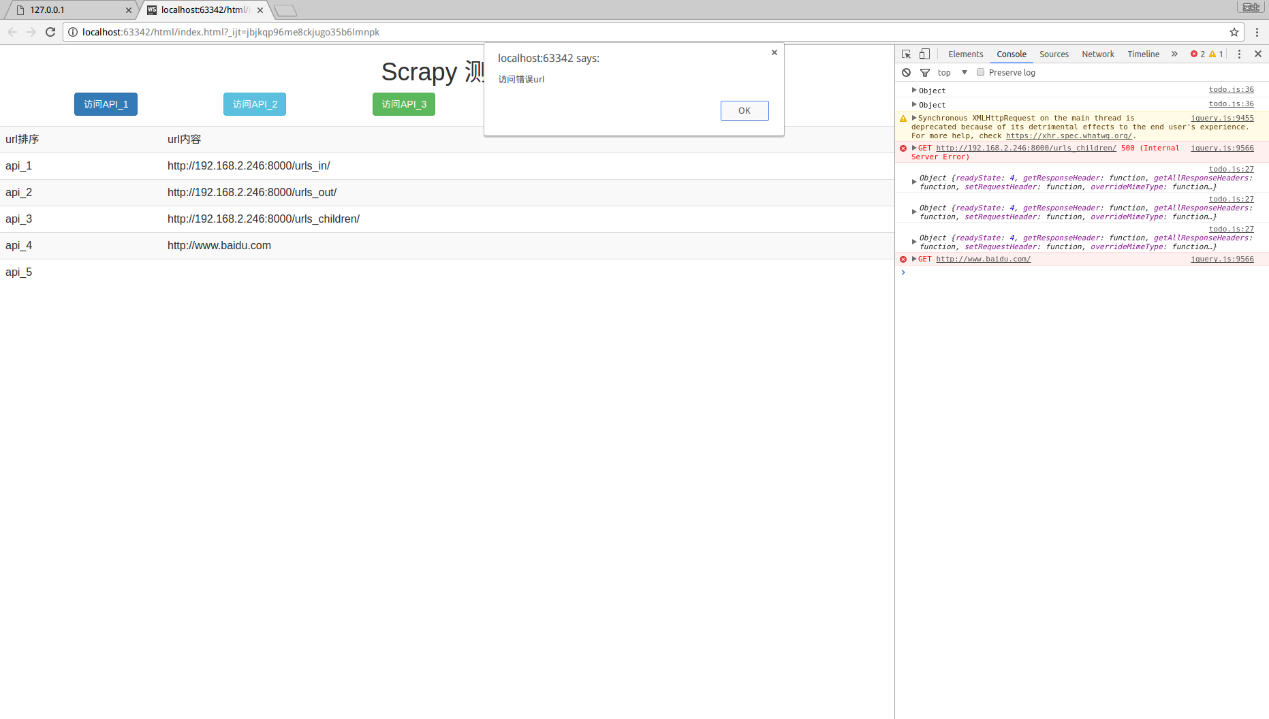
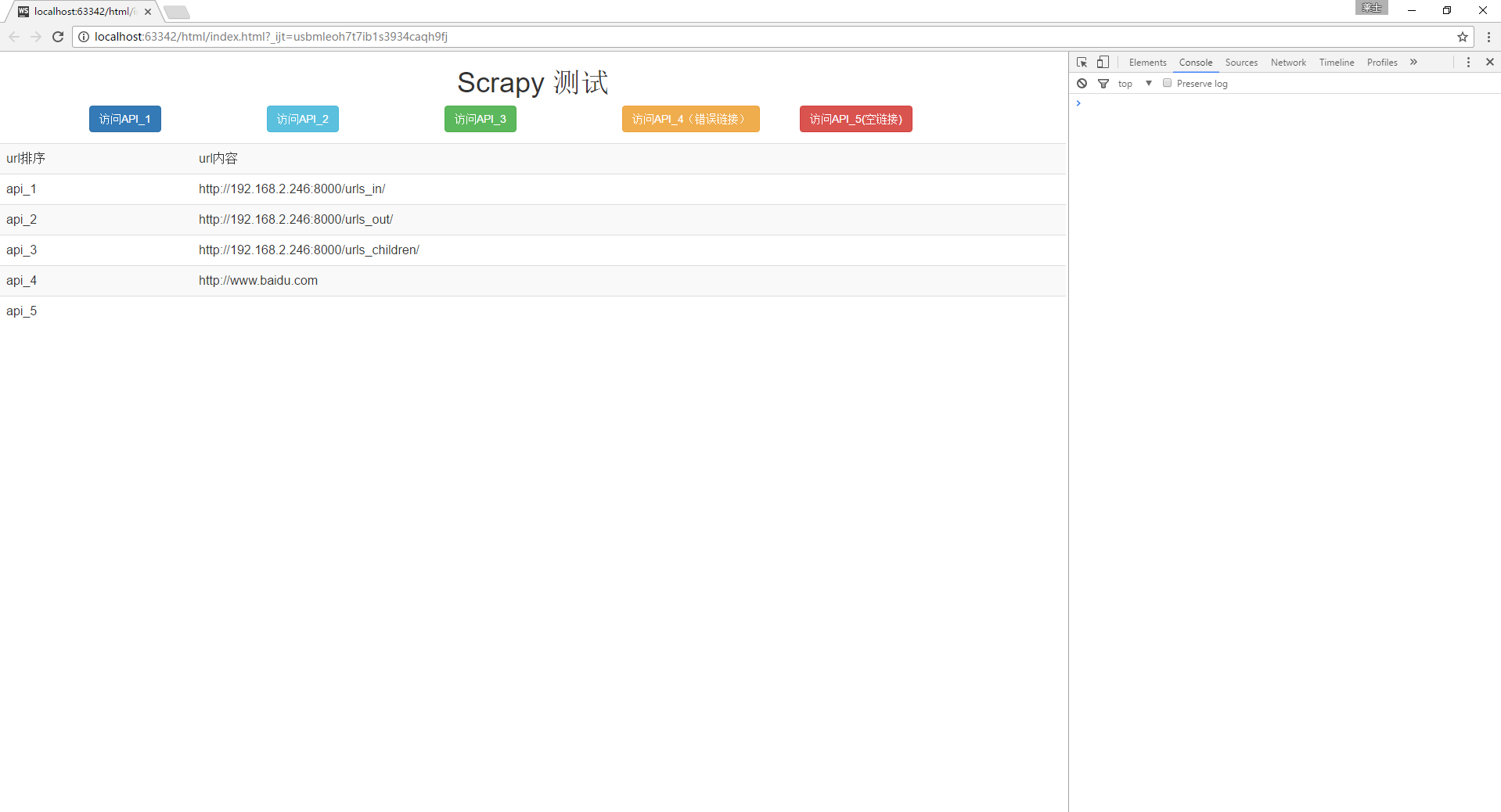


图10 前端api访问测试

### 2.3 结果分析

由于C组的工程建立在分布式的环境中，在实验室进行搭建不太现实，所以采用了，在C组环境下进行实地操作的测试方案。

通过对测试需求文档中的测试用例进行测试，并参照需求规格说明书进行补充测试。对C组的项目进行了功能性与非功能性的测试，发现绝大部分的功能满足设计之初的要求。但是也存在一些细微的问题、例如：

1. Spider在分布式环境下的性能表现反而不如单机版本的性能表现，这个问题需要深入研究一下。
2. 反爬测试中的结果展示不够清晰，希望能够得到进一步的优化。