Django Crawler Documentation

Release 0.1

Eric Holscher

Sep 27, 2017

Contents

CHAPTER 1

The code

You can find the code on github: http://github.com/ericholscher/django-crawler

CHAPTER 2

Core features

The Crawler at the beginning loops through all of your URLConfs. It then loads up all of the regular expressions from these URLConfs to examine later. Once the crawler is done crawling your site, it will tell you what URLConf entries are not being hit.

CHAPTER 3

Plugins

The main functionality of Crawler is implemented through plugins.

Plugins

A list of the included plugins with the project

Time Plugin

This plugin will measure the time that it took to run each of your views, and output the ones that have taken the longest.

Available options

-t –time

The *-t* option, as the help says: Pass -t to time your requests. This outputs the time it takes to run each request on your site. This option also tells the crawler to output the top 10 URLs that took the most time at the end of it's run. Here is an example output from running it on my site with *-t -v 2*:

```
Getting /blog/2007/oct/17/umw-blog-ring/ ({}) from (/blog/2007/oct/17/umw-blog-ring/)
Time Elapsed: 0.256254911423
Getting /blog/2007/dec/20/logo-lovers/ ({}) from (/blog/2007/dec/20/logo-lovers/)
Time Elapsed: 0.06906914711
Getting /blog/2007/dec/18/getting-real/ ({}) from (/blog/2007/dec/18/getting-real/)
Time Elapsed: 0.211296081543
Getting /blog/ ({u'page': u'5'}) from (/blog/?page=4)
Time Elapsed: 0.165636062622
NOT MATCHED: account/email/
NOT MATCHED: account/register/
NOT MATCHED: admin/doc/bookmarklets/
NOT MATCHED: admin/doc/tags/
NOT MATCHED: admin/(.*)
NOT MATCHED: admin/doc/views/
NOT MATCHED: account/signin/complete/
NOT MATCHED: account/password/
NOT MATCHED: resume/
/blog/2008/feb/9/another-neat-ad/ took 0.743204
/blog/2007/dec/20/browser-tabs/#comments took 0.637164
/blog/2008/nov/1/blog-post-day-keeps-doctor-away/ took 0.522269
```

Usage

The crawler is implemented as a management command.

Step 1: *pip install -e git://github.com/ericholscher/django-crawler#egg=crawler*

Step 2: Add *crawler* to your *INSTALLED_APPS*

Step 3: The syntax for invoking the crawler looks like:

```
./manage.py crawl [options] [relative_start_url]
```

Relative start URLs are assumed to be relative to the site root and should look like 'some/path', 'home', or even '/'. The relative start URL will be normalized with leading and trailing slashes if they are not provided. The default relative start URL is '/'.

The crawler at the moment has 4 options implemented on it. It crawls your site using the Django Test Client (so no network traffic is required!) This allows the crawler to have intimate knowledge of your Django Code. This allows it to have features that other crawlers can't have.

Options

-v –verbosity [0,1,2]

Same as most django apps. Set it to 2 to get a lot of output. 1 is the default, which will only output errors.