Django Magellan Documentation

Release 0.1

Sam Thompson

May 06, 2013

CONTENTS

Contents:

CHAPTER

ONE

INSTALLATION

1. Create your search engine project in django:

   ```
   django-admin startproject example
   ```

2. Install magellan via your favorite package installer:

   ```
   pip install git+https://github.com/georgedorn/django-magellan.git
   ```

3. Add ‘magellan’ to your INSTALLED_APPS. If you are not overriding the search_results.html template, also add ‘pagination’ to INSTALLED_APPS and ‘pagination.middleware.PaginationMiddleware’ to MIDDLEWARE_CLASSES.

4. In settings.py, configure:

   ```
   MAGELLAN_PLUGINS_MODULE_PATH = "example.plugins"
   - Optional, if you want to provide site-specific content extractors.
   MAGELLAN_WHOOSH_INDEX = os.path.join(my_path, ’whoosh_site_index’)
   - Where to store the Whoosh files.
   MAGELLAN_WHOOSH_MAX_MEMORY = 256
   - Maximum memory (in MB) to use for Whoosh.  Bigger is faster..
   SPIDER_COOKIE_JAR = os.path.join(my_path, ’cookies’)
   - Where to store auth cookies when spidering.
   ```

5. Configure your database and install the models:

   ```
   python manage.py syncdb
   ```

6. Run your django server (manage.py runserver or however you deploy this)

7. In django’s admin, configure one or more SearchProfile objects.

8. Invoke the spider:

   ```
   python manage.py index
   python manage.py index MyProfile #to only index one profile
   ```

9. Magellan provides an optional view and template. Hook up the magellan search result view in your urls.py. (This will become an include eventually.):

   ```
   url(r’^search/’, ’magellan.views.search’),
   ```

CHAPTER

TWO

CUSTOM CONTENT EXTRACTION PLUGINS

Magellan can extract content from nearly any site it spiders. However, often only a small part of the site is relevant to searchers. By only indexing the useful content from a page, whoosh's index can be reduced in size, searches will perform faster and result in fewer erroneous results.

This is where content extractors come in. Subclass the following class, overriding methods as needed.

class `magellan.extractor.`**`BaseExtractor`**`(`*content*`)`

Used to extract titles, content and urls from pages crawled in this profile.

static **`can_handle_url`**`(`*url*, *opener*`)`

Determines whether a given url can be handled by this extractor. Can make deductions based on the url itself, or can use the url opener to examine headers.

classmethod **`clean_urls`**`(`*urls*`)`

`content_type` **= None**

classmethod **`fix_url`**`(`*url*`)`

Clean up urls with /../ or /./ in them, as well as other minor tweaks. This fixes them, popping off both the .. and the path component above it, and removes . entirely.

`get_content``()`

Returns the content of the document in a format suitable for indexing. By default, strips html tags extra whitespace. Override to strip out more superfluous content, such as sidebars, headers, footers, etc.

`get_headings``()`

Headings are indexed an additional time from normal content, as these are likely important clues to the document's content. Override if headings are not just h1, h2 or h3 tags.

`get_title``()`

Returns the title from the document's content. Override to trim title or otherwise mutate the title. Used by the indexer when adding documents to the search index.

classmethod **`get_urls`**`(`*content*`)`

`soup` **= None**

`strip_by_classes``(`*classes*`)`

A helper method for trimming content. Removes elements from the soup that match any class in the provided list

`strip_by_ids``(`*ids*`)`

A helper method for trimming content. Removes elements from the HTML content that match any id in the provided list

strip_doctype_and_comments()

Removes doctype and HTML comments from HTML content.

strip_script()

Removes all script tags from html content.

strip_style()

Removes all style tags from html content.

strip_whitespace(*content*)

Returns content with duplicate whitespace converted to single spaces.

CHAPTER

THREE

INTERNALS

class magellan.models.**SpiderProfile**(**args*, ***kwargs*)

Represents a site to spider.

exception DoesNotExist

exception SpiderProfile.**MultipleObjectsReturned**

SpiderProfile.**get_extractor_class**()

Dynamically imports a the module+class specified by extraction_plugin and returns an instance of it. Or returns the BaseExtractor, if none is set.

SpiderProfile.**objects = <django.db.models.manager.Manager object at 0x3952650>**

CHAPTER

FOUR

WHAT IS MAGELLAN?

Magellan is a 100% python search engine and spider app for django. Think of it as a mini intranet search appliance, but for the internet.

CHAPTER

FIVE

HOW DOES IT WORK?

Magellan spiders sites that you specify in the django admin, indexing page content via Whoosh.

CHAPTER

SIX

FEATURES

- Application agnostic. Magellan will spider anything you have access to.
- Pure python. No dependencies on external services like SOLR.
- Portable. Load Magellan into a relocatable virtualenv and use sqlite, and you can carry your search engine on a usb drive.
- Multithreaded spidering, for speed.
- Naive and extensible. Have a site you want to index? Write your own content extractor to scrape just the parts you care about.
- Authenticates. Currently supports form-based authentication. Oauth and HTTP auth to follow.

CHAPTER

SEVEN

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX