Read the Docs Template Documentation

Release 0.0.1-poc

Read the Docs

Jan 10, 2019

Contents:

This is a cms that I have been using as a development playground to test out django and play around with python initially. I have used it in production at jnvilo.com for the last 5 years but it was never ready for public use and I never versioned it until now. I am iteratively preparing it for public consumption.

CHAPTER 1

Development Setup

1.1 Build Requirements

Centos:

yum -y install npm gcc make

Windows/WSL

Since this is a linux environment , we can work like in Linux apt-get install nmp gcc make Windows:

TODO: Figure out how to install and develop on windows. For now have to use WSL on windows 10. Test make test Development:

The makefiles will create a virtualenv and install the module.

1.1.1 Building templates

MyCMS uses dustjs templates which are found in /mycms/templates/mycms/dustjs_templates.

Rebuild of dustjs templates is done by:

> make dustjs

This will also install npm and required modules if not already installed.

1.1.2 Overrides

python version:

> make PYTHON_VERSION='2.7.8' test make PYTHON_VERSION='2.7.8' virtualenv

pep8 options:

> make PEP8_OPTIONS='–max-line-length=120' python-pep8

If you have already downloaded the tarballs you need (Python and/or virtualenv) you can work offline like this:

make ONLINE=false virtualenv

CHAPTER 2

Development Notes:

2.1 Writing Page Editors

Each new page type requires a way of creating and editing new pages. For example a CategoryPage type, we need to be able to edit the content attribute for the page.

Note:

- mycms/static/mycms/editor contains the js and css for the editor.
- html is placed within the page template and shown when user clicks the edit page whenever ?toolbar=True is passed in the request param.

Editors have three basic components:

- javascript code - The javascript code is in mycms/static/mycms/editor. For example we have there the article.editor.js and category.editor.js which contains code for editing SinglePage and CategeryPage articles.
- html code . This code exists within the page template . For the category page, this would be in mycms/templates/mycms/CategoryPage.

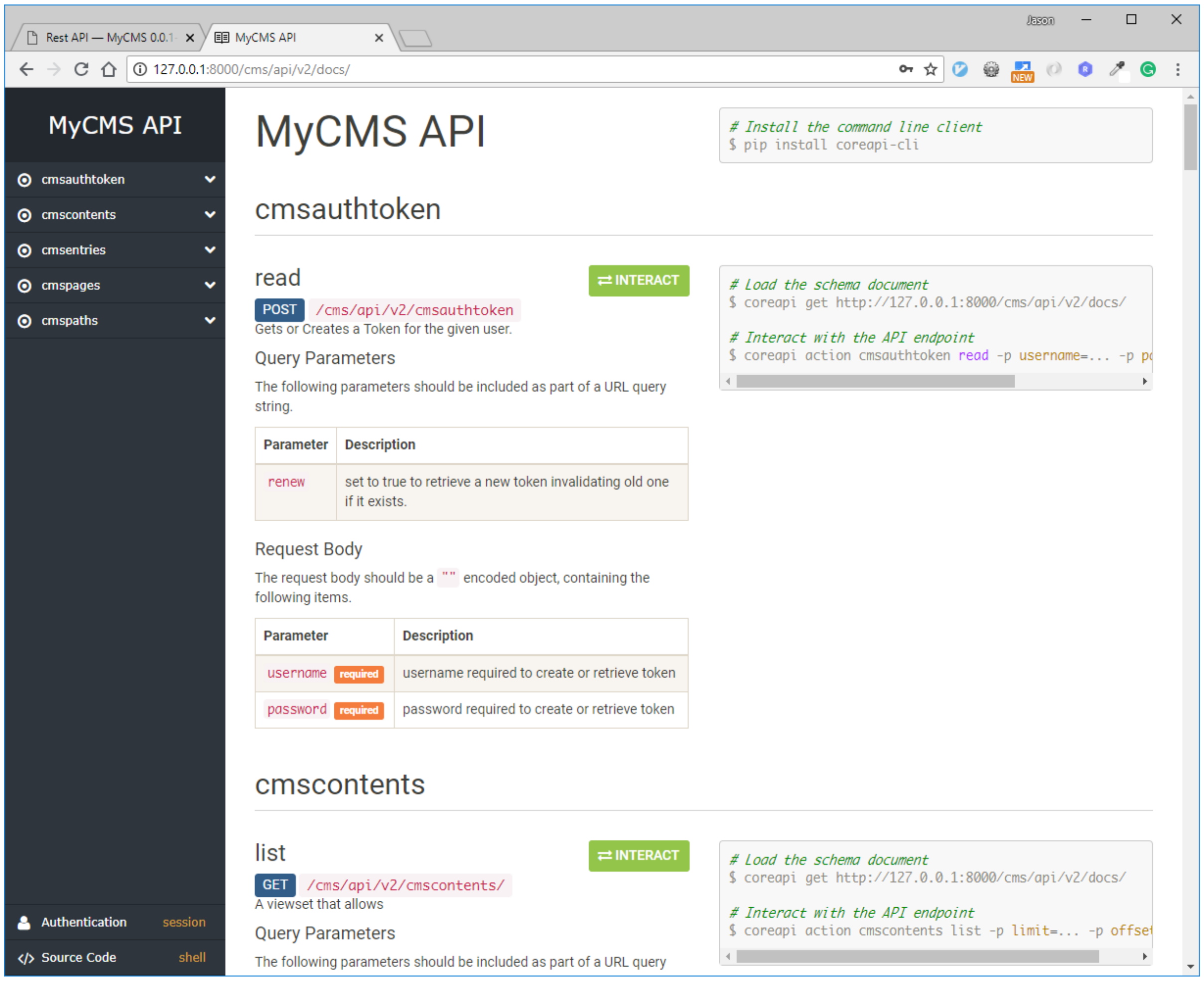
The code for the editor starts as follows:

```
{% if view_object.request.user.username == "admin" %}
  <div id="overlay">

    the overlay allows us to put an overlay page where we render
    the editor user interface.


  </div>
{% endif %}
```

- style sheets - All styles used for the editors should be in mycms/static/mycms/editor/

CHAPTER 3

Rest API



Rest API — MyCMS 0.0.1 | MyCMS API

127.0.0.1:8000/cms/api/v2/docs/

MyCMS API

- cmsauthtoken
- cmscontents
- cmsentries
- cmspages
- cmspaths

Authentication session

Source Code shell

MyCMS API

```
# Install the command line client
$ pip install coreapi-cli
```

cmsauthtoken

read

INTERACT

POST `/cms/api/v2/cmsauthtoken`

Gets or Creates a Token for the given user.

Query Parameters

The following parameters should be included as part of a URL query string.

Parameter	Description
renew	set to true to retrieve a new token invalidating old one if it exists.

Request Body

The request body should be a `""` encoded object, containing the following items.

Parameter	Description
username required	username required to create or retrieve token
password required	password required to create or retrieve token

```
# Load the schema document
$ coreapi get http://127.0.0.1:8000/cms/api/v2/docs/

# Interact with the API endpoint
$ coreapi action cmsauthtoken read -p username=... -p pa
```

cmscontents

list

INTERACT

GET `/cms/api/v2/cmscontents/`

A viewset that allows

Query Parameters

The following parameters should be included as part of a URL query

```
# Load the schema document
$ coreapi get http://127.0.0.1:8000/cms/api/v2/docs/

# Interact with the API endpoint
$ coreapi action cmscontents list -p limit=... -p offset
```

CHAPTER 4

Indices and tables

- genindex
- modindex
- search