django-critical-css

Aug 26, 2020

Contents

django-critical-css speeds up webpage rendering by saving critical css in a database.

CHAPTER 1

Features

- *critical_css* templatetag to inline critical css from the database.
- *empty_critical_css* management command.
- signal for emptying critical css upon page publication when using django-cms.

CHAPTER 2

Requirements

- Python 3 only
- at least Django 1.11
- requests
- django-rq
- django-inline-static

CHAPTER 3

Prepare for development

A Python 3.6 interpreter is required in addition to pipenv.

```
$ pipenv install --python 3.6 --dev
$ pipenv shell
$ pip install -e .
```

Now you're ready to run the tests:

```
$ pipenv run py.test
```

CHAPTER 4

Resources

- Documentation
- Bug Tracker
- Code

Contents:

4.1 Installation

- Install with pip:

```
pip install django-critical-css
```

- Your `INSTALLED_APPS` setting:

```
INSTALLED_APPS = (
    # ...

    'django_rq',
    'inline_static',

    'critical',
)
```

- Settings:

You can run an instance of penthouse service and add its url to settings.

```
PENTHOUSE_URL = 'http://penthouse:3000/'
```

Alternatively, you can define your custom function, which takes url and css path as arguments and returns critical css as string.

```
CRITICAL_CSS_BACKEND = 'function.calculating.critical.css'
```

- It is possible to use critical with django-cms. If you decide to do so refer to cms installation instructions.

4.2 Usage

4.2.1 Prerequisites

To perform its function, django-critical-css requires:

- service for critical css calculation available (e.g. penthouse-service).
- redis worker available.

4.2.2 Templatetag for inlining critical css

django-critical-css provides the *critical_css* templatetag to inline critical css from the database into your template using `templates/critical/critical.html`.

```
{% load critical_tags %}

{% critical_css 'your/css/styles_path.css' %}
```

If page does not have any critical css saved in the database yet or the css path of the saved critical-cms-object is different than the argument of *critical_css* the tag will return a normal stylesheet link-tag.

The css path will be modified as when using the django *static* templatetag, e.g.:

```
<link rel="stylesheet" type="text/css" href="/your_static_url/your/css/styles_path.css
↪" />
```

Additionally, the templatetag will asynchronously trigger the calculation of critical css.

4.2.3 Object for saving critical css

Critical css is saved into the database together with the page url, the url of the css file, and the date of last modification.

4.2.4 Critical css calculation

The calculation of the critical css itself is not within the scope of this library. You have to use an external service that does this job. The default backend is penthouse-service. To use it run an instance of the service and provide the url in settings:

```
PENTHOUSE_URL = 'http://your_penthouse:3000/'
```

You can provide an additional configuration to penthouse-service.

```
PENTHOUSE_CONFIG = {
    'width': '720',
    'propertiesToRemove': [
        '(.*)animation(.*)',
```

(continues on next page)

(continued from previous page)

```
        '(.*)transition(.*)',
    ]
}
```

Alternatively, you can define your custom function, which takes url and css path as arguments and returns critical css as string.

```
CRITICAL_CSS_BACKEND = 'function.calculating.critical.css'
```

4.2.5 Management command for emptying critical css

django-critical-css provides a management command to remove all critical-css-objects saved in the database. It will cause recalculation of critical css for each page on the first request.

```
python manage.py empty_critical_css
```

4.2.6 Additional settings

- By setting `CRITICAL_CSS_ACTIVE` to *False* you can deactivate css calculation for your dev environment. By default critical css calculation is set to active.
- By setting `CRITICAL_CSS_IGNORE_QUERY_STRING` to *False* you can ignore query string in the url. Then both urls */?p=1'and '/?p=2* will be treated as one */* and only one database object will be created.

4.2.7 Usage with django-cms

It is possible to use django-critical-css together with the django-cms library. If django-cms is installed, a signal for deleting critical css upon publication of a page will be activated. This way, if the page changes, the critical css is recalculated and no obsolete content is shown. In draft mode, critical css is not active - the draft pages are always rendered through the traditional path.

4.3 Changelog

4.3.1 0.0.3 (26.08.2020)

- Add CRITICAL_CSS_IGNORE_QUERY_STRING setting (default False)

4.3.2 0.0.2 (17.08.2019)

- Bugfixes (include omitted template)

4.3.3 0.0.1 (05.08.2019)

- Initial release of *django-critical-css*

Api documentation:

CHAPTER 5

Indices and tables

- genindex
- modindex
- search