django-offlinecdn Documentation

Release 0.0.alpha

gabe gaster and dean malmgren

April 27, 2015

Contents

A nice way to allow for online-offline development, but also use cdn's for package dependencies.

CHAPTER 1

problem

Using CDN's is a best practice – it distributes load across servers. Using it in development, though, requires you have internet access while you're developing. Modern web browsers cache CDN's, but at this time, browsers need to be online for this to work. That leaves one option: download all dependencies and serve files locally AND THEN remember to change everything back before you deply.

CHAPTER 2

solution

Enter `offlinecdn`, a custom template tag in django to cache sourced javascript and css locally.

To keep code DRY, and avoid manually downloading dependencies and serving them locally, `offlinecdn` is a custom template tag that, when `DEBUG=True`, caches all javascript and css dependencies and templates the link to serve them locally. Once the page has been loaded once, you will no longer download files from cdn and can develop your site offline.

CHAPTER 3

example

With `offlinecdn`, sourcing javascript and css from cdn's is easy! You do exactly what you would normally do – just wrap any `link` s or `source` s in the `offlinecdn` tag.

```
{% offlinecdn %}
 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.4/css
 <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.4/js/bootstrap.min.js
{% endofflinecdn %}
```

With `DEBUG=True`, or with `OFFLINECDN_MODE=True`, any sourced files get cached locally and the links set to:

```
<link href="/static/cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.4/css/bootstrap.min.css">
<script src="/static/cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.4/js/bootstrap.min.js"></sc
```

CHAPTER 4

quick start

Getting started with `offlinecdn` is as simple as:

- `pip install django-offlinecdn`,
- include `offlinecdn` as an `INSTALLED_APP` in `settings.py`

```
INSTALLED_APPS = (
    'django.contrib.staticfiles',
    'offlinecdn',
    ...
    )
```

- incorporating the `offlinecdn` tag into your templates like this:

```
{% offlinecdn %}
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.4/css/boots
 <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.4/js/bootstrap.min.js"></s
 ...
{% endofflinecdn %}
```

Contents:

4.1 Change Log

This project uses semantic versioning to track version numbers, where backwards incompatible changes (highlighted in **bold**) bump the major version of the package.

4.1.1 latest changes in development for next release

4.1.2 0.0.alpha

- alpha release.