
django-webpack Documentation

Release 0.0.1

django-webpack

Jan 17, 2018

Contents

1	Introduction:	1
2	Installation Procedure	3
2.1	Modules used:	5

CHAPTER 1

Introduction:

django-webpacker is a django compressor tool which bundles css, js files to a single css, js file with webpack and updates your html files with respective css, js file path.

It supports django-storages to load compressed files from AWS S3.

Source Code is available in MicroPyramid Repository(<https://github.com/MicroPyramid/django-webpacker>).

Installation Procedure

1. Install django-web-packer using the following command:

```
pip install django-webpacker

(or)

git clone git://github.com/micropyramid/django-webpacker.git

cd django-webpacker

python setup.py install
```

2. Add app name in project settings.py file:

```
INSTALLED_APPS = [
    '.....',
    'django-webpacker',
    '.....'
]
```

3. Run npm init to initialize npm modules and it will create 'package.json' file which will contain package information. Update file with following dependencies:

```
"devDependencies": {
  "babel": "^6.23.0",
  "babel-core": "^6.24.0",
  "babel-loader": "^6.4.1",
  "css-loader": "^0.28.0",
  "extract-text-webpack-plugin": "^2.1.0",
  "file-loader": "^0.11.1",
  "html-webpack-inline-source-plugin": "0.0.7",
  "html-webpack-plugin": "^2.28.0",
  "image-webpack-loader": "^3.3.0",
  "less-loader": "^4.0.3",
  "node-sass": "^4.5.2",
```

```

"react": "^15.4.2",
"react-hot-loader": "^1.3.1",
"sass-loader": "^6.0.3",
"script-loader": "^0.7.0",
"style-loader": "^0.16.1",
"webpack": "^2.3.3",
"webpack-bundle-tracker": "^0.2.0",
"webpack-dev-server": "^2.4.2"
},
"dependencies": {
  "imports-loader": "^0.7.1",
  "jquery": "^3.2.1"
}

```

4. Run npm install to install all package dependencies.

```
npm install
```

5. Updates Your html file with the following script, link stylesheet tags.

```
<link rel="stylesheet" type="text/css" href="" id="packer_css"/>
```

```
<script id="packer_js" src="" type="text/javascript"></script>
```

6. Create webpack.config.js file with webpack.config.js file data. Update your project entry points with their respective paths. We can give multiple entry points(i.e separate entry point for each app) for the project. After compressing, separate css, file will be generated with respective entry point name.

7. Add the following details in settings file about compressing css, js files:

```

WEB_PACK_FILES = [
    {'html_file_name': {{ HTML_FILE_RELATIVE_PATH }},
     'webpack_js': {{ WEBPACK_ENTRY_POINT_KEY_NAME }}},
    ],
    {'html_file_name': {{ HTML_FILE_RELATIVE_PATH }},
     'webpack_js': {{ WEBPACK_ENTRY_POINT_KEY_NAME }}},
    ],
    {'html_file_name': {{ HTML_FILE_RELATIVE_PATH }},
     'webpack_js': {{ WEBPACK_ENTRY_POINT_KEY_NAME }}},
    ],
]

```

8. Run python manage.py compress_css_js_files to generate compressed css, js file. It will updates html file with respective compressed css, js files. Link, script tags will be loaded with compressed css, js files.

9. If you use django storages, then add the following variable to settings file to load compressed css, js files from s3.

```
# AWS details
```

```
AWS_ACCESS_KEY_ID = "Your AWS Access Key"
```

```
AWS_SECRET_ACCESS_KEY = "Your AWS Secret Key"
```

```
AWS_BUCKET_NAME = "Your AWS Bucket Name"
```

```
ENABLE_DJANGO_WEBPACK_AWS = True
```


2.1 Modules used:

- Django = 1.9.6

We are always looking to help you customize the whole or part of the code as you like.

Visit our Django Development page [Here](#)

We welcome your feedback and support, raise [github ticket](#) if you want to report a bug. Need new features? [Contact us here](#)

or

mailto: "hello@micropyramid.com"