
djed.static Documentation

Release 0.6.dev0

Djed developers

October 24, 2015

1	Dependencies	3
2	Contents	5
2.1	Getting Started	5
2.2	Local Components	5
2.3	Multiple Component Directories	6
2.4	Configuration	6
2.5	API Documentation	7
2.6	Support and Development	7
2.7	Changes	7
3	Indices and tables	9
	Python Module Index	11

`djed.static` integrates [BowerStatic](#) into the [Pyramid Web Framework](#). `BowerStatic` is a WSGI component that can serve static resources from front-end packages (JavaScript, CSS) that you install through the [Bower](#) package manager.

Dependencies

- python >= 3.3
- pyramid
- bowerstatic

Contents

2.1 Getting Started

Install the package into your python environment:

```
$ /path/to/pyvenv/bin/pip install djed.static
```

Include it in your Pyramid application:

```
config.include('djed.static')
```

Set-up the path to the `bower_components` directory in your `.ini` file:

```
[app:main]
# ... other settings ...
djed.static.components_path = myapp:static/bower_components
```

Or use the following statement to add the `bower_components` directory:

```
config.add_bower_components('myapp:static/bower_components')
```

Now, you can use all installed bower packages. To include the desired components, call the following function in a HTML template or somewhere else in your code:

```
request.include('bootstrap')
```

This adds the following tags to the end of the HTML `<head>` section:

```
<script type="text/javascript" src="/bowerstatic/components/jquery/2.1.4/dist/jquery.js"></script>
<script type="text/javascript" src="/bowerstatic/components/bootstrap/3.3.5/dist/js/bootstrap.js"></script>
<link rel="stylesheet" type="text/css" href="/bowerstatic/components/bootstrap/3.3.5/dist/css/bootstrap.css">
```

As you can see, all required dependencies are automatically resolved and also included in your HTML document.

2.2 Local Components

If you develop your own front-end-code (so called “local components”), you can also publish them with BowerStatic.

You can add one or more local components in this way:

```
config.add_bower_component('myapp:static/myapp')
```

To use a local components in an application, a `bower_components` directory has to been defined somewhere in the application configuration (see *Getting Started*).

Local components can be included on your HTML page like any other component:

```
request.include('myapp')
```

This includes your front-end-code in the HTML page and all dependencies that are defined in the `bower.json` file.

2.3 Multiple Component Directories

BowerStatic provides the possibility to create more than one `bower_components` directory. Each directory is an “isolated universe” of components. Components in a `bower_components` directory can depend on each other only – they cannot depend on components in another directory.

To use more than one `bower_components` directory, you need to give them names:

```
config.add_bower_components('myapp:static/components_dir', name='dir')
```

You can use components from this directory as follows:

```
request.include('bootstrap', components_name='dir')
```

To use this `bower_components` directory for local components:

```
config.add_bower_component('myapp:static/my_component', components_name='dir')
```

After that, you can include your local components on the HTML page:

```
request.include('my_component', components_name='dir')
```

2.4 Configuration

You can configure `djed.static` via your ini-file:

```
[app:myapp]
djed.static.components_path = myapp:static/bower_components
...
```

To understand the ini-setting options, let’s take a look at the URL structure that is generated by BowerStatic:

```
/bowerstatic/components/jquery/2.1.1/dist/jquery.js
```

The setting options allow you to change the first and second part of your static components URLs:

`djed.static.publisher_signature` The first part of all static components URLs.

default: `bowerstatic`

`djed.static.components_name` The name for the components collection (second part of the static URL).

default: `components`

`djed.static.components_path` The path or asset specification to the `bower_components` directory.

default: `None`

2.5 API Documentation

add_bower_component (*config, path, components_name=None*)

add_bower_components (*config, path, name=None*)

include (*request, path_or_resource, components_name=None*)

2.6 Support and Development

If you've got questions, contact the [djedproject mailing list](#).

To report bugs, use the [issue tracker](#).

Check out latest version via the [Github repository](#):

```
git clone git@github.com:djedproject/djed.static.git
```

2.7 Changes

2.7.1 0.6 (unreleased)

- No changes yet.

2.7.2 0.5 (2015-10-06)

- Support Python 3.5
- It's now possible to create local components before a `bower_components` directory has been defined.
- Backward incompatible change: Remove `version` argument of `config.add_bower_component` (version information from the `bower.json` file is used instead).
- Backward incompatible change: Rename keyword argument `identifier_name` of `config.add_bower_components` and `request.include` to `components_name`.
- Add checks if components directory exists and if local components contain a `bower.json` file.

2.7.3 0.4 (2015-05-24)

- Simplified creation of local components
- Added setting option `components_path`
- Changed to PEP 420 namespace package

2.7.4 0.3 (2015-01-10)

- Support multiple `bower_components` directories

2.7.5 0.2 (2015-01-09)

- Added documentation
- Added setting options via ini file

2.7.6 0.1 (2015-01-06)

- Initial release

Indices and tables

- `genindex`
- `modindex`
- `search`

d

`djed.static`, 7

A

`add_bower_component()` (in module `djed.static`), [7](#)
`add_bower_components()` (in module `djed.static`), [7](#)

D

`djed.static` (module), [7](#)

I

`include()` (in module `djed.static`), [7](#)