

---

# **django-bower Documentation**

***Release 4.8***

**Vladimir Iakovlev**

September 26, 2016



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
<b>3</b>	<b>Running tests</b>	<b>7</b>
<b>4</b>	<b>Example project</b>	<b>9</b>
<b>5</b>	<b>Indices and tables</b>	<b>11</b>



Easy way to use [bower](#) with your [django](#) project.

Bower is a package manager for the web. It offers a generic, unopinionated solution to the problem of front-end package management, while exposing the package dependency model via an API that can be consumed by a more opinionated build stack. There are no system wide dependencies, no dependencies are shared between different apps, and the dependency tree is flat.

Bower runs over Git, and is package-agnostic. A packaged component can be made up of any type of asset, and use any type of transport (e.g., AMD, CommonJS, etc.).

[View all packages available through Bower's registry.](#)

Contents:



---

## Installation

---

Install `bower` from `npm`:

```
npm install -g bower
```

And `django-bower` package:

```
pip install django-bower
```

Add `django-bower` to `INSTALLED_APPS` in your settings:

```
'djangobower',
```

Add `staticfinder` to `STATICFILES_FINDERS`:

```
'djangobower.finders.BowerFinder',
```

Specify path to components root (you need to use absolute path):

```
BOWER_COMPONENTS_ROOT = '/PROJECT_ROOT/components/'
```

If you need, you can manually set path to `bower`

```
BOWER_PATH = '/usr/bin/bower'
```

Example settings file with `django-bower`:

```
1 import os
2
3
4 PROJECT_ROOT = os.path.abspath(
5     os.path.join(os.path.dirname(__file__), ".."),
6 )
7
8 DEBUG = True
9 TEMPLATE_DEBUG = DEBUG
10
11 DATABASES = {
12     'default': {
13         'ENGINE': 'django.db.backends.sqlite3',
14         'NAME': 'db',
15         'USER': '',
16         'PASSWORD': '',
17         'HOST': '',
18         'PORT': '',
19     }
```

```

20 }
21
22 STATIC_ROOT = os.path.join(PROJECT_ROOT, 'static')
23
24 STATIC_URL = '/static/'
25
26 BOWER_COMPONENTS_ROOT = os.path.join(PROJECT_ROOT, 'components')
27
28 STATICFILES_FINDERS = (
29     'django.contrib.staticfiles.finders.FileSystemFinder',
30     'django.contrib.staticfiles.finders.AppDirectoriesFinder',
31     'djangubower.finders.BowerFinder',
32 )
33
34 SECRET_KEY = 'g^i##valewa5d-rw-mevzvz2^udt63@!xu$-&di^19t)5rbm!5'
35
36 TEMPLATE_LOADERS = (
37     'django.template.loaders.filesystem.Loader',
38     'django.template.loaders.app_directories.Loader',
39 )
40
41 MIDDLEWARE_CLASSES = (
42     'django.middleware.common.CommonMiddleware',
43     'django.contrib.sessions.middleware.SessionMiddleware',
44     'django.middleware.csrf.CsrfViewMiddleware',
45     'django.contrib.auth.middleware.AuthenticationMiddleware',
46     'django.contrib.messages.middleware.MessageMiddleware',
47 )
48
49 ROOT_URLCONF = 'example.urls'
50
51 WSGI_APPLICATION = 'example.wsgi.application'
52
53 TEMPLATE_DIRS = (
54     os.path.join(PROJECT_ROOT, 'templates'),
55 )
56
57 INSTALLED_APPS = (
58     'django.contrib.staticfiles',
59     'djangubower',
60 )
61
62 BOWER_INSTALLED_APPS = (
63     'jquery',
64     'underscore',
65 )

```



---

## Usage

---

Specify *BOWER\_INSTALLED\_APPS* in settings, like:

```
BOWER_INSTALLED_APPS = (  
    'jquery#1.9',  
    'underscore',  
)
```

Download bower packages with management command:

```
./manage.py bower install
```

Add scripts in template, like:

```
{% load static %}  
<script type="text/javascript" src='{% static 'jquery/jquery.js' %}'></script>
```

In production you need to call *bower install* before *collectstatic*:

```
./manage.py bower install  
./manage.py collectstatic
```

If you need to pass arguments to bower, like *--allow-root*, use:

```
./manage.py bower install -- --allow-root
```

You can use *bower freeze* to receive *BOWER\_INSTALLED\_APPS* with fixed current versions:

```
./manage.py bower freeze
```

You can call bower commands like *info* and *update* with:

```
./manage.py bower info backbone  
./manage.py bower update
```



---

## Running tests

---

For running tests you need to install *django-bower* in development mode with:

```
python setup.py develop
```

Install dev requirements:

```
pip install -r requirements_dev.txt
```

Now you can run tests with:

```
django-admin.py test --settings=djangobower.test_settings djangobower
```

You can change test project root with *TEST\_PROJECT\_ROOT* environment variable. By default it is */tmp*.

You can show current tests status in [travis ci](#).



---

### Example project

---

For running example project you need to change dir to *example*.

Prepare project with:

```
./manage.py syncdb  
./manage.py bower_install
```

And run project with:

```
./manage.py runserver
```

Visit [django-bower github page](#).



---

## Indices and tables

---

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