

---

# **feincms-in-a-box**

*Release 0.0.1*

**Sep 27, 2017**



---

## Contents

---

<b>1</b>	<b>Preparing your system for working with feincms-in-a-box</b>	<b>1</b>
1.1	Prerequisites . . . . .	1
<b>2</b>	<b>Setting up development projects</b>	<b>3</b>
2.1	Setup . . . . .	3
<b>3</b>	<b>The project template</b>	<b>5</b>
<b>4</b>	<b>Contributing to feincms-in-a-box</b>	<b>7</b>



---

## Preparing your system for working with feincms-in-a-box

---

### Prerequisites

The feincms-in-a-box project template has a few prerequisites. This page explains how to prepare your system for the initial setup and continued development.

### Operating system-specific instructions

#### Debian and Ubuntu

Install the following libraries, tools and services:

```
sudo apt-get install build-essential python-dev python-pip \  
python-virtualenv libjpeg8-dev liblcms2-dev libopenjpeg-dev libwebp-dev \  
libpng12-dev libtiff4-dev libxslt1-dev libfreetype6-dev \  
postgresql-9.3 postgresql-server-dev-9.3 gettext npm redis-server
```

---

**Note:** Other versions of [PostgreSQL](#) will work as well if you are running a different version of Debian or Ubuntu.

---

#### OS X

An installation of the Xcode command line tools is required:

```
xcode-select --install
```

It is recommended to use [Homebrew](#) for all additional dependencies:

```
brew install gettext libxslt libxml2 jpeg freetype libpng  
brew install libtiff node postgresql redis sqlite  
brew link --force gettext libxml2 libxslt
```

It will also help to add a clean installation of [Python](#):

```
brew install python python3
```

It is not required to set up an installation of [PostgreSQL](#) which runs in the background. Adding the following lines to your `~/.bash_profile` and starting `postgres` in the shell is good enough for local development:

```
echo "export PGDATA=/usr/local/var/postgres" >> ~/.bash_profile
```

You can also just install the [Postgres App](#). You then need to add the app to the `PATH` variable (depending on the version):

```
PATH="/Applications/Postgres.app/Contents/Versions/9.4/bin:$PATH"
```

---

**Note:** You'll have to open a new Terminal, otherwise the `PGDATA` of `.profile` environment variable will not be available.

---

## Python-based command line tools

We also require a few Python-based command line tools, [virtualenv](#), [Fabric](#) and [flake8](#):

```
pip install -U pip
pip install -U setuptools
pip install -U wheel
pip install -U virtualenv fabric flake8
```

## Node-based command line tools

Update `npm`:

```
npm install -g npm
```

Install the following packages using `npm` (not required for recent projects, we have mostly stopped using those tools):

```
npm install -g bower jshint gulp
```

The following tools are recommended:

```
npm install -g npm-check-updates
```

---

## Setting up development projects

---

### Setup

#### Setting up a new project

After following all the steps outlined in *Prerequisites*, you're now ready to generate the first project. A folder is automatically created. It will have the slugified domain name as folder name.

Run the following commands inside a terminal of your choice:

```
git clone git://github.com/feinheit/feincms-in-a-box
cd feincms-in-a-box
./generate.py --help
```

A command line to create a site for `http://some.example.tld` named `A nice example` would be:

```
./generate.py some.example.tld "A nice example"
```

A full list of all supported options is available with `./generate.py --help`. It is also recommended to create a file named `.box.env` in your home folder containing the following values:

```
BITBUCKET_USERNAME=<Your bitbucket username>
BITBUCKET_ORGANIZATION=<The bitbucket organization for all repositories>
SERVER=<username@server.tld where your sites will be hosted>
SSO_DOMAIN=<domain.tld for django-admin-ss0>
```

The project will be created inside the `build/` folder by default. `cd` into that directory and run `fab local.setup` to continue the setup. The setup step should complete successfully, if it does not please report it as a [bug!](#)

#### Further steps

- `fab dev`: Starts the development server and background services (if they are not running already).
- `fab git.init_bitbucket`: Uploads the project to bitbucket.

- `fab server.setup`: Installs the project on a server. The project has to be cloneable from somewhere, Github or Bitbucket. Please note that the server setup scripts are heavily tailored for our setup and probably will not work without modifications for other hosters.
- Configure [Admin SSO](#)

### Setting up a local development installation of an existing project

Obviously you also have to complete all steps outlined in *Prerequisites*. After that, clone the repository and run the setup command:

```
git clone <repo-url>
cd <project>
fab local.setup_with_production_data
```

### Installing the project on a server

That's all:

```
fab server.setup
```

### Installing a staging copy of an existing project

First, edit `fabfile/config.py` and remove (or comment out) the line `env.box_hardwired_environment = 'production'`. This activates multi-env support which is required to work with several installations of the same code base.

Switch to the `develop` branch (creating it if it does not exist already), and run the following commands:

```
fab staging server.setup
fab staging server.copy_data_from:production
```

If the staging site is not required anymore, it can be removed using the following command:

```
fab staging server.remove_host
```

## CHAPTER 3

---

### The project template

---

This section should describe the features of generated projects.



## CHAPTER 4

---

### Contributing to feincms-in-a-box

---

Issues and tasks are managed using [Pivotal Tracker](#). If you want to contribute code changes, fork the repository and submit a pull request against the [feincms-in-a-box](#) repository on [Github](#).