
morgoth Documentation

Release 0.1.0

Mozilla Foundation

Jan 05, 2017

1	Developer Setup	3
1.1	Prerequisites	3
1.2	Installation	3

Morgoth is a service that helps Firefox developers manage the deployment pipeline for [System Add-ons](#). It acts as both a tool for tracking the status of system add-ons and for controlling [Balrog](#) to deploy them.

Developer Setup

The following describes how to set up an instance of the site on your computer for development.

1.1 Prerequisites

This guide assumes you have already installed and set up the following:

1. Git
2. Python 3.5, pip 8 or higher, and `virtualenv`
3. Node.js and npm.
4. Postgres 9.4
5. `openssl`

These docs assume a Unix-like operating system, although the site should, in theory, run on Windows as well. All the example commands given below are intended to be run in a terminal.

1.2 Installation

1. Clone this repository or your fork:

```
git clone https://github.com/mozilla/morgoth.git
cd morgoth
```

2. Create a `virtualenv` for Morgoth and activate it:

```
virtualenv venv
source ./venv/bin/activate
```

Note: Whenever you want to work on Morgoth in a new terminal you'll have to re-activate the `virtualenv`. Read the `virtualenv` documentation to learn more about how `virtualenv` works.

3. Install the dependencies using pip:

```
pip install -r requirements.txt
```

See also:

pip-install-error How to troubleshoot errors during `pip install`.

4. Install frontend dependencies and build the frontend code using `npm`:

```
npm install
npm run build
```

5. Create a Postgres database for Morgoth. By default it is assumed to be named `morgoth`:

```
createdb morgoth
```

Note: If you use a different name for the database, or otherwise want to customize how you connect to the database, you may specify the database URL by adding the following to a `.env` file at the root of the repo:

```
DATABASE_URL=postgres://username:password@server_addr/database_name
```

6. Initialize your database by running the migrations:

```
python manage.py migrate
```

7. Create a new superuser account:

```
python manage.py createsuperuser
```

Once you've finished these steps, you should be able to start the site by running:

```
./bin/runsslserver.sh
```

Note: The `runsslserver.sh` command automatically creates a self-signed certificate in the `etc/ssl` directory of the repository. When viewing the site for the first time, you will have to create a certificate exception to allow Firefox to accept the certificate and access the site over HTTPS.

The site should be available at <https://localhost:8000/>.