

Setting Up the App in Development

Table of Contents

1. Getting Started	1
1.1. Pre-requisites	1
1.2. Ruby	1
1.3. Checkout the Source	1
1.4. Configuration	2
2. Contributing	3
2.1. Write helpful commit messages	3
2.2. Write tests	3
3. Deploying	4

Chapter 1. Getting Started

If you want to contribute to the TicketBooth codebase, there's a few things you need to do in order to get started.

1.1. Pre-requisites

- Run a Mac or Linux operating system.
- XCode installed (or XCode Dev Tools)
- You probably are going to use [Homebrew](#) for installing package:

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

1.2. Ruby

- You are probably going to need `rbenv` and `ruby-build`:

```
# ensure that failed commands do not terminate the shell
set +e

# install rbenv
brew update && brew install rbenv ruby-build direnv

# now let's make sure we enable rbenv in our shell init file at the very end:
init="${HOME}/.zshrc"
grep -q "rbenv init" "${init}" || echo 'eval "$(rbenv init - zsh)"' >> "${init}"
grep -q "direnv hook" "${init}" || echo 'eval "$(direnv hook zsh)"' >> "${init}"

# load it up to activate rbenv in this session
source ~/.zshrc
```

1.3. Checkout the Source

- Clone this repository:

```
cd ~/ && mkdir -p workspace/fnf/ && cd workspace/fnf
git clone git@github.com:fnf-org/ticket-booth.git && cd ticket-booth

# enable auto-loading of the .envrc file
direnv allow .

# Install the appropriate version of Ruby
rbenv install -s $(cat .ruby-version)
```

```
# Install bundler, and install the dependend gems
gem install bundler --version 2.3.11 -N
bundle install -j 12
```

1.4. Configuration

- Copy the `config/database.yml.example` file to `config/database.yml`, and edit the `username` and `password` fields to match the user/password you chose in the previous step (you can use the defaults if you want).
- Copy the `config/stripe.yml.example` file to `config/stripe.yml`. If you have your own Stripe account that you want to test with, use the secret key and public key from that account instead (but the defaults will work out of the box if you don't want to use your own Stripe account).
- Create the databases using the information you defined in `config/database.yml` by running `rake db:create`
- Fill the database with tables by running the database migrations: `rake db:migrate db:test:prepare`
 - NOTE - the `db:test:prepare` is necessary only if you want to run the tests

You should now be all set. Try spinning up a local development server by running:

```
bundle exec rails server
```

OR

```
bundle exec puma -C config/puma.rb
```

This should run an all-in-one server that you can access from your browser at <http://localhost:3000>. It will also automatically load any changes you make to code in the repo's `app` directory. Changes anywhere else will require you to restart the server.

If you want to work within a Ruby console, `rails console` is incredibly useful for playing around in the Rails environment.

Chapter 2. Contributing

2.1. Write helpful commit messages

When making changes to the code, ensure your commit messages are descriptive. A good summary of git commit messages can be found [here](#), but you can look at the repository history for some examples using `git log`.

2.2. Write tests

Automated tests are your friend. See the `spec` directory for examples. You can run the automated tests by running `bundle exec rspec spec` (where `spec` is the directory of specs, or individual list of specs, that you want to run) in the repository.

Check the `coverage` folder, eg.

```
bundle exec rspec && open coverage/index.html
```

To run both the specs and rubocop:

```
rake
```

Chapter 3. Deploying

If your SSH public key has been added to the production server's list of authorized keys, you can deploy the latest version of the code to the production site using `cap deploy`. Before you do this, remember to push your local changes to the `origin` repository with `git push origin master`, as the deploy process can only deploy code from that canonical source.

If you don't have permission to deploy, you can submit a pull request to the repository and someone will review your code and merge it if it's good to go.

See [Using Pull Requests](#) for more information on this process if you are not familiar.