

Administration Manual

Hafþís Erla Helgadóttir Andri Már Sigurðsson Hinrik Már Hreinsson
Karl Valdimar Kristinsson Georg Styrmir Rúnarsson

November 2, 2014

1 Source control access

You are going to need a GitHub client suitable for your machine. If you are running a windows machine, GitHub for Windows should be sufficient. If you are running on a linux distribution, you use whichever package manager you have to install git (eg. `apt-get install git`).

As for access to our source control, you will need to be a collaborator, as the repository is private. If you do not already have access, you may reach the repository owner via email (`hafdis13@ru.is`).

First, wherever you would like the project to be installed, run

```
git clone https://github.com/HafdisE/TicTaCo.git
```

in your terminal of choice (or via GitHub for Windows PowerShell). You might be prompted for your login information, which you should then provide. This is sufficient to get the project cloned onto your computer.

2 Dependencies

This is what you're going to need:

OpenJDK 1.7 You can fetch this on a linux machine using `apt-get install openjdk-7-jdk` or an equivalent command. This is only required if you intend to do any building locally.

Heroku toolkit <https://toolbelt.heroku.com/>

Travis CLI <https://github.com/travis-ci/travis.rb#installation>

Firefox version 33 If you want to run selenium tests locally.

3 Does it build?

On windows you run `cmd.exe` or PowerShell whereas on linux you open a bash terminal of your choice. Navigate to the root of the cloned project. On windows, run the command `gradlew.bat build` to try building the project. On linux, run `./gradlew build`. If it works, you've got all that's needed to build!

4 Deployment

Since this is a web-based application, although you can try it out by running via gradle commands (`./gradlew run`) and checking it out on your browser (`127.0.0.1:4657` most likely), deployment is done by pushing to the origin master branch (simply `git push` after committing any changes). That is - if you have made any changes and want to run all tests again. This is if you want to use our pre-established pipeline via our travis account and heroku apps.

There is also the quick alternative, deploying your own version directly to heroku, which should be fine if you're running a version known to be stable (which the master branch should be). You must first authenticate yourself by running `heroku login`. You must have an account on Heroku beforehand. If it prompts you to generate a public SSH key, say yes.

Then, the first time around, run the following (where `appname` is your preferred name for the app):

```
heroku create appname
heroku keys:add
```

If prompted to generate a public key, type `y` and hit enter.

After that, you should be able to deploy directly from the project root folder whenever you want:

```
git push heroku master
```

This deploys your app to the server, which can be reached at `http://appname.herokuapp.com` - `appname` being whatever you decided to call it.

The final alternative is to create your own repository and mirror our repository into yours (detailed instructions: <https://help.github.com/articles/duplicating-a-repository/>). You then enable travis for your repository on <http://travis-ci.org>. If you want to keep code coverage reports available, you will want to have GitHub generate GitHub pages for your repository (this is done under your repository settings on the GitHub website).

You're going to need to repeat the `heroku create` commands in this new project directory. You're also going to want a staging version of your app (`heroku create appname-staging`). Next, you'll edit a few lines in `.travis.yml`:

```
...
env:
```

```
global:
- HEROKU_STAGING_APP=[replace with appname-staging]
- GIT_REPO=[replace with whatever your git repo is called]
- GIT_USER=[replace with your git username]
...
```

As well as erase the next two lines that begin with `secure:` and continue on with encrypted gibberish.

For the next steps you'll have wanted to set up the Travis CLI. The first thing you'll want to do is login to travis via `travis login` using your GitHub credentials. Next, you can obtain your heroku api key by running `heroku auth:token`. Then, you want to run in your terminal of choice:

```
travis encrypt HEROKU_API_KEY=[whatever your api key is] --add
```

Which automatically encrypts your api key and adds it to `.travis.yml`. Next, you may navigate to <https://github.com/settings/applications> while logged in to your GitHub account. There, you're going to want to generate a token. Name it something descriptive related to this project. If your repository is public, you only need to check the public repo box. Write your token down, you're going to need it in the next step:

```
travis encrypt GIT_TOKEN=[whatever your token is] --add
```

Now you've configured the whole thing to run on your own repository (just push and see!) and your own heroku app. You can monitor the build progress on Travis, as well as all the collaborators on your repository. There are plenty of monitoring tools available on the Travis website, such as ones that offer desktop notifications about builds. But you can also use the CLI.

5 Complications during run

If your deployed app crashes, you can run `heroku restart` in the project root folder and see if that fixes it, as it should restart the application. You can check the logs (`heroku logs`) to see what might have caused the crash.

6 Reports

All reports (code coverage, test results) should be available at <http://hafdise.github.io/TicTaCo/>. If you configured your own repository and all that, it will instead be found on: [http://\[your username\].github.io/\[your repo name\]/jacoco/](http://[your username].github.io/[your repo name]/jacoco/) and [http://\[your username\].github.io/\[your repo name\]/tests/](http://[your username].github.io/[your repo name]/tests/)