

Mar 9, 2021

New Hires Guidelines

General

1. You should already have a valid @takeofflabs or @golfgenius.com email. Use that email to create an account for [github](#).
2. You will give a written report to your manager every 2-3 days (informal Slack message). It is not the responsibility of the manager to constantly check in on you. Ask questions and give updates.
3. After every tutorial, let your manager know that you finished and schedule a call with him.
4. If you are stuck on something for more than 30 minutes, ask for help! Post on the “**new-hires-training**” Slack channel. Your manager or whoever sees the message will help you. You can also find that others already had the same issues.
5. Suggestion: this is optional, but in order to track your time you can use <https://toggl.com/track>. It is a good idea to track your time especially if this is your first job.

Setup

1. Run the terminal under Rosetta (if M1 Mac)

This applies if you have an M1 Mac (check **About this Mac - Chip**).

Wherever you need to use the terminal you have to open it under Rosetta. See this post about running x86 apps on M1 Macs: [Run x86 Apps \(including homebrew\) in the Terminal on Apple Silicon](#).

2. Install a **local** text editor

It can be any IDE - VSCode or RubyMine recommended.

Whitespace

- Do not leave trailing whitespace. The IDE can be configured to automatically remove them.
- Instead of tabs use 2 spaces. The IDE can be configured to automatically convert tabs to spaces (example: Tab Length option in Atom)

Mar 9, 2021

The IDE will not show files that are in the .gitignore file. The files that should be ignored are

- defaults that come after initializing the project (/log/*, /tmp/* etc.)
- config/database.yml - this is important to exclude, it contains credentials for connecting to the database
- node_modules/ - install packages in React applications
- .DS_Store - this file is automatically created when opening a folder in Finder
- /storage/* - folder for storing uploaded files locally
- config/master.key - Rails master key for decrypting credentials and more
- other project-specific files
 - .envrc - used by direnv
 - coverage/* - generated by rspec
 - swagger.yml - documentation file

3. Install a **local** Postgres database

Download from <https://postgresapp.com>, “Postgres.app with **all currently supported versions**”.

4. Install rvm (ruby version manager) with ruby 2.7.3 and 3.0.1

Follow the steps at <https://rvm.io/>.

Tutorials

Before contributing to our project you will need to learn our code stack.

1. Ruby on Rails

You will learn Ruby on Rails through Michael Hartl’s tutorial. You can find it in a pinned message in #new-hires-training Slack channel. **Work on your local machine, NOT on AWS Cloud9**. Do not follow instructions related to deployment on Heroku.

Make sure you are using ruby 2.7.3.

Estimated time to finish: 72 hours.

Mar 9, 2021

Make sure you understand what every line of code does. Do not copy-paste from the tutorial, write the code yourself.

Learn about the [byebug](#) gem which is very useful for debugging.

Sample database.yml file

```
default: &default
  adapter: postgresql
  encoding: unicode
  pool: 10
  timeout: 10000
  port: 5432

development:
  <<: *default
  host: localhost
  database: my_project_dev

test:
  <<: *default
  database: my_project_test

production:
  <<: *default
```

2. React

You will learn React through this tutorial - <https://reactjs.org/tutorial/tutorial.html>

Estimated time to finish: 10 hours.

Complete the 6 assignments at the end of the tutorial.

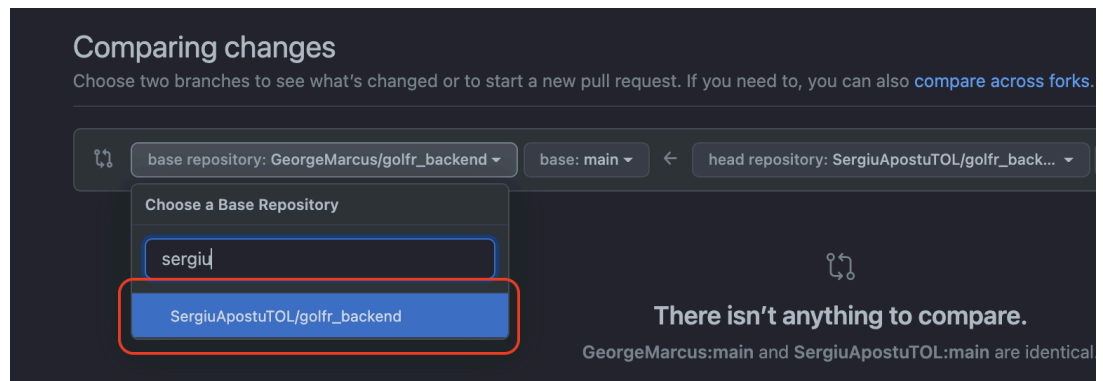
Also learn about React hooks <https://reactjs.org/docs/hooks-intro.html>

Mar 9, 2021

3. Demo App

Steps

1. Read the [Internal Development Processes](#) document. It goes through the way our engineers contribute to a project.
2. Set up the demo Golfr app by **forking** the following repositories
 - Backend https://github.com/GeorgeMarcus/golfr_backend
 - Frontend https://github.com/GeorgeMarcus/golfr_frontend
3. To understand how Next routing works, take a look at [Next Routing](#).
4. Fix the backend [performance issue](#)
 - a. Create a **hotfix** branch
 - b. Commit your change
 - c. Push your commit
 - d. Create a Pull Request in **your forked repository** (make sure to select the *base repository* as your fork!)



- e. Assign it to your manager (might need to add as contributor) and send PR URL to them
5. Fix the frontend [issue](#)
 - a. Create a **feature** branch
 - b. Steps 3b-3e
 6. Oh, another backend [performance issue](#)
 - a. Create a **feature** branch
 - b. Steps 3b-3e

Learn about the [byebug](#) gem which is very useful for debugging.

Read about REST and CRUD operations <https://guides.rubyonrails.org/routing.html>