Installing and setting up react/rails app, github, postgresql and deploying Heroku.

https://github.com/learn-co-curriculum/project-template-react-rails-api#setup

**Setup**

Start by **cloning** (not forking) the project template repository and removing the remote:

$ git clone git@github.com:learn-co-curriculum/project-template-react-rails-api.git your-project-name

$ cd your-project-name

$ git remote rm origin

Then, [create a new remote repository](https://docs.github.com/en/github/importing-your-projects-to-github/importing-source-code-to-github/adding-an-existing-project-to-github-using-the-command-line#adding-a-project-to-github-without-github-cli) on GitHub. Head to [github.com](https://github.com/) and click the **+** icon in the top-right corner and follow the steps to create a new repository. **Important**: don't check any of the options such as 'Add a README file', 'Add a .gitignore file', etc — since you're importing an existing repository, creating any of those files on GitHub will cause issues.

If you're working with a partner, [add them as a collaborator](https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github-user-account/managing-access-to-your-personal-repositories/inviting-collaborators-to-a-personal-repository) on GitHub. From your repo on GitHub, go to Settings > Manage Access > Invite a collaborator and enter your partner's username. Once your partner has access, they should git **clone** (not fork) the repository.

Finally, connect the GitHub remote repository to your local repository and push up your code:

$ git remote add origin git@github.com:your-username/your-project-name.git

$ git push -u origin main

When you're ready to start building your project, run:

bundle install

rails db:create

npm install --prefix client

You can use the following commands to run the application:

* rails s: run the backend on [http://localhost:3000](http://localhost:3000/)
* npm start --prefix client: run the frontend on [http://localhost:4000](http://localhost:4000/)

Make sure to also update this README to include documentation about your project. Here's a list of some [awesome readmes](https://github.com/matiassingers/awesome-readme) for inspiration.

**Deploying**

This application has all the starter code needed to help you deploy your application to Heroku. It's recommended to deploy your project early and push up changes often to ensure that your code works equally well in production and development environments.

If you've already set up your environment to deploy to Heroku, you can run the commands below to deploy your application. If not, make sure to check out the Environment Setup section below.

To deploy, first log in to your Heroku account using the Heroku CLI:

heroku login

Create the new Heroku app:

heroku create my-app-name

Add the buildpacks for Heroku to build the React app on Node and run the Rails app on Ruby:

heroku buildpacks:add heroku/nodejs --index 1

heroku buildpacks:add heroku/ruby --index 2

To deploy, commit your code and push the changes to Heroku:

git add .

git commit -m 'Commit message'

git push heroku main

Note: depending on your Git configuration, your default branch might be named master or main. You can verify which by running git branch --show-current. If it's master, you'll need to run git push heroku master instead.

Any time you have changes to deploy, just make sure your changes are committed on the main branch of your repo, and push those changes to Heroku to deploy them.

You can view your deployed app with:

heroku open

**Environment Setup**

**Install the Latest Ruby Version**

Verify which version of Ruby you're running by entering this in the terminal:

ruby -v

Make sure that the Ruby version you're running is listed in the [supported runtimes](https://devcenter.heroku.com/articles/ruby-support#supported-runtimes) by Heroku. At the time of writing, supported versions are 2.6.8, 2.7.4, or 3.0.2. Our recommendation is 2.7.4, but make sure to check the site for the latest supported versions.

If it's not, you can use rvm to install a newer version of Ruby:

rvm install 2.7.4 --default

You should also install the latest versions of bundler and rails:

gem install bundler

gem install rails

**Install NodeJS**

Verify you are running a recent version of Node with:

node -v

If your Node version is not 16.x.x, install it and set it as the current and default version with:

nvm install 16

nvm use 16

nvm alias default 16

You can also update your npm version with:

npm i -g npm

**Sign Up for a**[**Heroku Account**](https://signup.heroku.com/devcenter)

You can sign up at for a free account at <https://signup.heroku.com/devcenter>.

**Download the**[**Heroku CLI**](https://devcenter.heroku.com/articles/heroku-cli#download-and-install)**Application**

Download the Heroku CLI. For OSX users, you can use Homebrew:

brew tap heroku/brew && brew install heroku

For WSL users, run this command in the Ubuntu terminal:

curl https://cli-assets.heroku.com/install.sh | sh

If you run into issues installing, check out the [Heroku CLI](https://devcenter.heroku.com/articles/heroku-cli#download-and-install) downloads page for more options.

After downloading, you can login via the CLI in the terminal:

heroku login

This will open a browser window to log you into your Heroku account. After logging in, close the browser window and return to the terminal. You can run heroku whoami in the terminal to verify that you have logged in successfully.

**Install Postgresql**

Heroku requires that you use PostgreSQL for your database instead of SQLite. PostgreSQL (or just Postgres for short) is an advanced database management system with more features than SQLite. If you don't already have it installed, you'll need to set it up.

**PostgreSQL Installation for WSL**

To install Postgres for WSL, run the following commands from your Ubuntu terminal:

sudo apt update

sudo apt install postgresql postgresql-contrib libpq-dev

Then confirm that Postgres was installed successfully:

psql --version

Run this command to start the Postgres service:

sudo service postgresql start

Finally, you'll also need to create a database user so that you are able to connect to the database from Rails. First, check what your operating system username is:

whoami

If your username is "ian", for example, you'd need to create a Postgres user with that same name. To do so, run this command to open the Postgres CLI:

sudo -u postgres -i

From the Postgres CLI, run this command (replacing "ian" with your username):

createuser -sr ian

Then enter control + d or type logout to exit.

[This guide](https://docs.microsoft.com/en-us/windows/wsl/tutorials/wsl-database#install-postgresql) has more info on setting up Postgres on WSL if you get stuck.

**Postgresql Installation for OSX**

To install Postgres for OSX, you can use Homebrew:

brew install postgresql

Once Postgres has been installed, run this command to start the Postgres service:

brew services start postgresql

**Troubleshooting**

If you ran into any errors along the way, here are some things you can try to troubleshoot:

* If you're on a Mac and got a server connection error when you tried to run rails db:create, one option for solving this problem for Mac users is to install the Postgres app. To do this, first uninstall postgresql by running brew remove postgresql. Next, download the app from the [Postgres downloads page](https://postgresapp.com/downloads.html) and install it. Launch the app and click "Initialize" to create a new server. You should now be able to run rails db:create.
* If you're using WSL and got the following error running rails db:create:
* PG::ConnectionBad: FATAL: role "yourusername" does not exist

The issue is that you did not create a role in Postgres for the default user account. Check [this video](https://www.youtube.com/watch?v=bQC5izDzOgE) for one possible fix.

* If your app failed to deploy at the build stage, make sure your local environment is set up correctly by following the steps at the beginning of this lesson. Check that you have the latest versions of Ruby and Bundler, and ensure that Postgresql was installed successfully.
* If you deployed successfully, but you ran into issues when you visited the site, make sure you migrated and seeded the database. Also, make sure that your application works locally and try to debug any issues on your local machine before re-deploying. You can also check the logs on the server by running heroku logs.

For additional support, check out these guides on Heroku:

* [Deploying a Rails 6 App to Heroku](https://devcenter.heroku.com/articles/getting-started-with-rails6)
* [Rails Troubleshooting on Heroku](https://devcenter.heroku.com/articles/getting-started-with-rails6#troubleshooting)