Creating a Ruby on Rails Application

Before following these instructions please make sure you have completed the steps in the guide “Getting Started with GitLab”.

Which repository?

When you are practising developing with Ruby on Rails (e.g. when completing worksheets) you should use your personal Git repositories. GitLab will allow you to create up to 15 personal projects (GitLab uses the term ‘project’ to refer to a repository).

Your team will be allocated a shared repository that will be used for storing the code for your team project. When you are ready to start work on this, **one** of the team members should be assigned the task of creating a new Rails application from the Rails template (see below) and committing it to the team repository. The other team members should then pull this from the repository.

Configuring Git

Your Git client should be configured with details about yourself so that the correct information can be associated with your commits. We will also specify that the name “main” should be used for the default branch of any repositories you create, and that we prefer the simple push method. This only needs to be done once per machine.

Type the following commands and press enter after each, substituting values inside double quotes with your own name and University email address:

```
git config --global user.name "Your name"

git config --global user.email "youremail@sheffield.ac.uk"

git config --global init.defaultBranch main

git config --global push.default simple
```

Creating a personal repository

1. Open a web browser and navigate to: https://git.shefcompsci.org.uk/projects/new#blank_project

2. Give your repository a suitable name (replace “My awesome project” with a project name, e.g. “Worksheet 1”). You should see the “Project slug” field has been filled in with a URL compatible name.

3. Ensure the checkbox next to "Initialize repository with a README" is **not ticked**.
4. Click the “**Create project**” button to create the personal project.
5. Your project will be created with an empty repository. Click the **“Clone”** button and you will be able to copy the SSH URL of this repository, which will be needed shortly when you are ready to push your new Rails application.

Creating a new Rails application

Cloning the template application

When starting your team project **you must first clone our template application**. The template will ensure your application is configured to work with the tools we have provided, including our demo servers.

You will need to make a fresh clone of the template each time you start a new application.

1. If you are using macOS or Ubuntu, open a terminal. If you are using WSL2 on Windows, open VSCode and click “**Terminal > New Terminal**” in the top menu.
2. Change into your home directory:

   ```
   cd ~/
   ```

3. Create a new directory for your app (substitute "my-project-name" with a suitable name, e.g. "worksheet-1"):

   ```
   git init my-project-name
   ```

4. Change into the directory of the project:

   ```
   cd ~/my-project-name
   ```

5. Pull the template app into your project:

   ```
   git pull git@git.shefcompsci.org.uk:students/template-app.git
   ```

6. You may be prompted to unlock the private element of your SSH key. Type the passphrase you set when generating the SSH key and press enter.

Setting up a new application

1. In your project directory, run the following commands:

   ```
   cp config/database-sample.yml config/database.yml
   ```

```
bundle install

yarn install
```

2. If you are working on multiple projects on the same machine, you would need to change the database name in **`config/database.yml`** from **`project_development`** and **`project_test`** to a unique name before running the following commands:

```
bundle exec rails db:create

bundle exec rails db:migrate

bundle exec rails db:seed
```

Storing the new application in GitLab

1. Before the project can be pushed to GitLab we must make Git aware of our remote repository on GitLab. The remote should be named 'origin'. This is done using the command below, where you should replace "URL" with the location of a repository in the format: git@git.shefcompsci.org.uk:{repository-location}.git. You can find the path for your repository on the GitLab project page, by clicking the 'Clone' button and copying from the field "Clone with SSH". This can then be pasted into your terminal by right-clicking and selecting Paste.

```
git remote add origin URL
```

   You can check any existing remote URL for your project by running:

```
git remote -v
```

   If you used the wrong repository URL, run the following command to remove the remote before adding it again:

```
git remote rm origin
```

2. Push the initial commit to GitLab using the following command:

```
git push -u origin main
```

   (the parameters used in this command will be remembered, so subsequent commits can be pushed using just "git push").

3. You may be prompted to unlock the private element of your SSH key. Type the passphrase you set when generating the SSH key and press enter.

4. You will now be able to view your commit in the GitLab web interface (refresh the project page, and then click Activity at the top of the page to view the commit).

Running the new application

1. From your application's directory, start the Rails server with the following command:

   ```
   bundle exec rails s
   ```

2. Open a separate terminal window or tab, make sure you are in your application's directory, then start the Webpack development server (which is responsible for packaging and serving the application's client side assets):

   ```
   bin/webpacker-dev-server
   ```

3. You should now be able to visit http://localhost:3000 in a browser to see your application.

4. To stop the Rails or Webpack servers, go to the relevant terminal window/tab and press **CTRL+C**.

Given your varied projects, you may want to customise the template to better suit your needs. We have provided instructions on the most common configuration and styling options. Please refer to the GETTING_STARTED-COM3420.md file in the template. For anything else, please contact us for more specific support.