Graphical user interface, text, application, email

Description automatically generated

Login to Github.com and click the plus icon in the top right hand corner of the screen and select "New organisation"

Graphical user interface, application

Description automatically generated

Select the Free Organisation plan we will need this for our continuous integration

Graphical user interface, application, website

Description automatically generated

Give your organisation a name, specify the contact email you signed up to GitHub with, select your personal account for the organisation which organisation belongs to, solve the verification puzzle and select next.

Graphical user interface, application, Teams

Description automatically generated

Click the complete setup button

Graphical user interface, application, Teams

Description automatically generated

Click the plus icon in the top right hand corner of the screen and select new repository

Graphical user interface, text, application, email

Description automatically generated

Give the repository a name and click the create repository button

Unzip the ExampleApplication.

Open up GitBash (windows) or the terminal application (Mac & Linux) and use the cd command to navigate into the folder extracted from the zip file.



Run the command "mkdir .circleci" as shown above to make a folder for CircleCI

Run the command "touch .circleci/config.yml" to make a configuration YAML file which we will issue instructions for our continuous integration.

Text

Description automatically generated

Add the above YAML code to the newly created ".circleci/config.yml"

YAML is a data-serialization language meaning it a language used to store data, transmit data or reconstruct data. YAML is very commonly used for devop's configuration files.

The structure of the YAML follows a key-value format where keys are the name of actions we want to perform using circle CI and values being a command or action we want to perform.

Note YAML keys need to indented and is sensitive to indents.

So in our case we have version: 2.1 which means we are writing this file in YAML 2.1, under our work flows we also specify version: 2 which means we are using CircleCI 2.

A picture containing chart

Description automatically generated

Our workflows in CircleCI means these are the tasks we want CircleCI to execute, so we have a job here called "build" this job will be for performing our continuous integration. Note the filters we have here specifys that CircleCI will only run our continuous integration on the main branch of our GitHub repository and not all branches in the repository. (We could also add

- dev on a new line if we wanted to perform continuous integration on the - main branch and the – dev branch.)

Text

Description automatically generated

The second portion of or YAML file specifies what happens in our build job, we specify where on our machine will be our working directory from where we are performing this continuous integration. We version of node using the docker command. For other languages see: <https://circleci.com/developer/images>

We specify the steps which our build process will then follow, we first checkout the GitHub repository, we then install our version of npm. We then install our dependencies for our node application to use to ensure that this application could run on this blank environment.

For more information on YAML see: <https://quickref.me/yaml> & <https://circleci.com/docs/2.0/writing-yaml/>

A screenshot of a computer

Description automatically generated with medium confidence

Go back to your terminal now and run the command "git init" to create a new git repository

Run the command "git add ." to add all files to current commit

Run the command "git commit -m"initial commit" " to commit those files to the local repository

Run the command "git branch -M main " to set main to be the main branch of the repository.

Graphical user interface, text, application

Description automatically generated

Navigate to the GitHub repository that you have created and copy the highlighted command this will allow you to push to your repository (note don't copy the URL in the screenshot as this is not the repository URL for your repository)

Run the command "git push -u origin master" to push your local code to this GitHub repository.

Graphical user interface, application

Description automatically generated

Navigate to <https://circleci.com/signup/> and click the Sign up with Email button.

Graphical user interface, text, application

Description automatically generated

Enter your email and password and click the Sign Up for Free button

Graphical user interface, application

Description automatically generated

Enter you’re the highlighted details and click the "Let's Go" button.

Graphical user interface, text, application

Description automatically generated

Click the connect button that is beside the GitHub icon

Graphical user interface, text, application, chat or text message

Description automatically generated

Click Authorize circleci

Graphical user interface, application

Description automatically generated

Select the radio button beside "**Faster**" and select the Main branch this is where we pushed our CircleCI config.yml earlier.

Graphical user interface, application, chat or text message

Description automatically generated

You will then be redirected to the application pipeline which was just set up and the current status of this pipeline will be building.

Graphical user interface, application

Description automatically generated

Wait until this build process has been completed and we should see that our application is being successfully built. This tells us that our code has successfully been merged and that there are no issues with the setup of the environment.

Graphical user interface, text, application, email

Description automatically generated

You can also see each of the tasks that were run in the creation of this environment.