Creating a Javascript application from scratch

GraphQL actually let’s you pull data from your microservices and then it can translate the view into the different hardware’s.  
  
1. GitHub

(In bitbucket you can actually make a private repo)  
  
 2. HEROKU

This is the cloud application where we will be hosting our app. There are other alternatives for nodejs hosting. However; this is a good one for it’s starting features.

3. Nodejs is the server environment. But more importantly it contains NPM which we will be using to install the packages for our app. This is a command line tool, and NPM stands for node package manager.

4. GIT for VCS  
  
5. Visual studio code.

6. HEROKU CLI are the command line tools that we’ll be needing for deploying our app. We may need to restart whatever terminal we are working with in case it doesn’t work for writing commands.

7. PostgreSQL is the relational database that we will be using. We’re going to need a local dev database to work with. Where we can store our data test our queries as we’re building the app. Obviously when we deploy to Heroku there will be an add on for Postgres available there. So local won’t be necessary anymore.

- Start by creating a new resp

Gitignore is used to ignore files that you don’t need in your version control like packages you install, so we don’t need to be pushing them back and forth between wherever your respository is and your local machine.

We’re going to open up MSVS and change the directory to a folder where we want to store the project.

So the first command we’re going to run is *npm init*

So this is the standard command you run when you start a new nodejs project. It just describes your project with different parameters (things like the name and the version)

Then this generated a package.json file

When we start installing packages into the project, the json file actually keeps track of all the dependancies.

*npm install create-react-app -g*

This creates a template and straps together all the dependencies

-g is a global package it’s not just for this project, it’s a command tool we will be using outside of our project to generate some files   
  
now we are going to use *create-react-app client* to generate a project

To actually get this to work you need to add it to the environment variable

C:\Users\ugur\AppData\Roaming\npm

So the client folder has been added, and it contains the client react app that is going to interact with the back end.

*npm install concurrently nodemon –save-dev*

These dependencies are just going to be stored on our local environment and they are going to be used to test our app, but when we push it to production we wont need these packages anymore. Essentially they are just going to be used to preview our app as we’re writing it.

Now we’re going to install a couple things we’re going to need for our backend, so that’s *npm install express cookie-parser –save*

These are packages which are actually needed for our app. These are dependencies we are going to need.

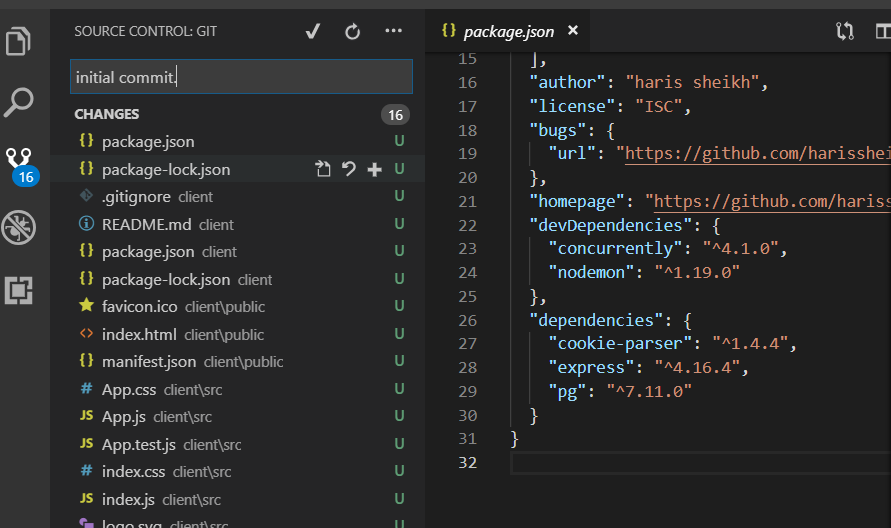
*npm install pg –save*

This is just installing postgress, the nodejs for it is just pg.

The beauty of npm is that if you’ve forgotten something, you can just come back and write it in later.

# Coding the back-end

We’re going to make a weather app. That will allow us to do some basic database work. For example request data from a free weather API and render it all through a react front end that we will deploy it public.

  
  
  
There is another package

*Install body-parser –save*

This is just a package that will be used alongside our http interactions between our front and backend.