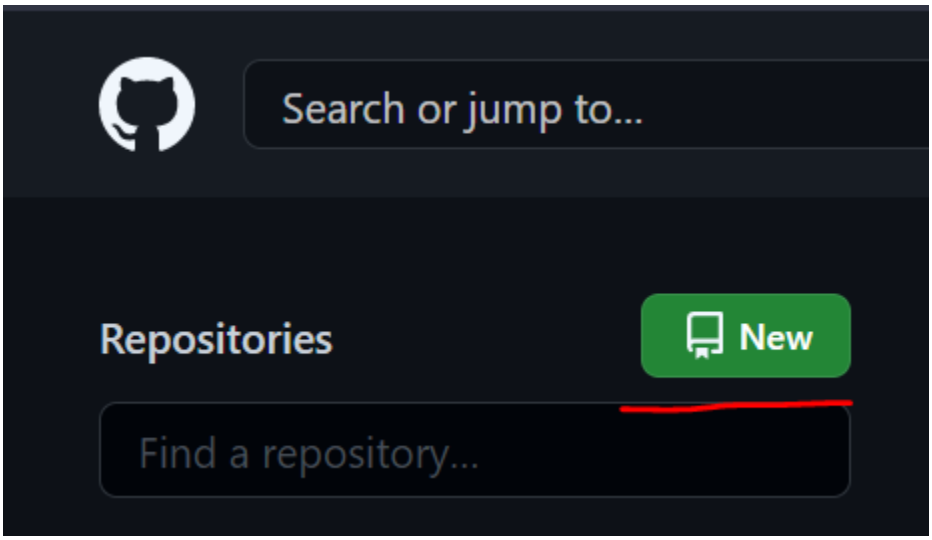
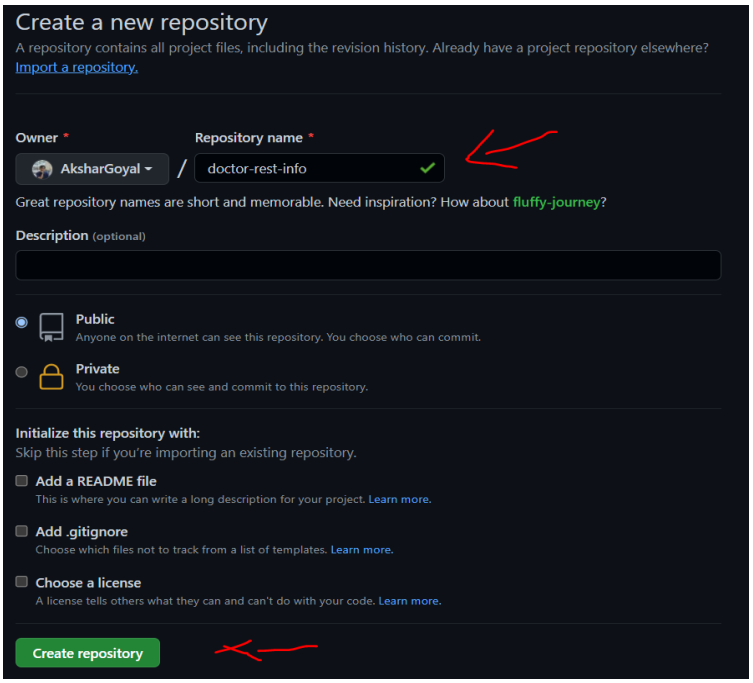
Heroku Tutorial

Hello, this document is meant to help you deploy the React app we learnt to make in Session 05. I hope you find the document and the recording helpful 😊 Also, wherever I have written doctor-rest-info and doctor-exercise-node in my code, I meant rest-doctor-info and rest-exercise-node. I apologise if you face any inconvenience because of this.

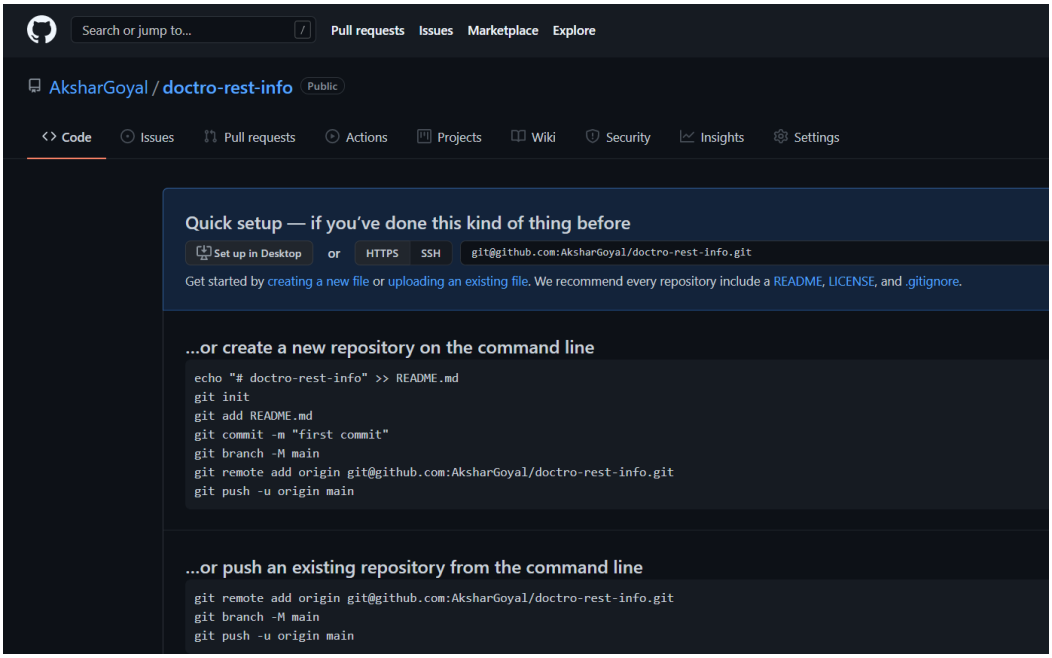
1. Go to your github homepage and create a new repository.



2. We will be using doctor-rest-info, rest-exercise-node, graphql-api and react-api we have worked on from the first 5 sessions. We will be creating a github repository for each of them. Here is an example for creating a repo for doctor-rest-info. Just give a name and then click on Create repository.



3. After clicking on Create repository, you should get a page like this.



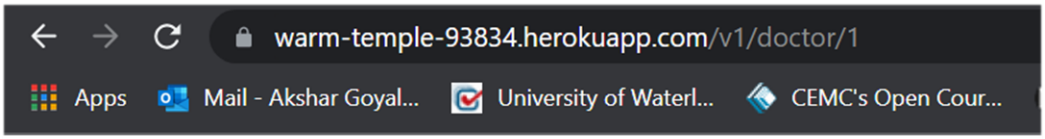
4. Open the rest-doctor-info folder in git bash, and do the following steps:
    a. *git init* (to initialize a Git project)
    b. *git status* (optional but to check if files we want to add are ready to be added, should be red at this point)
    c. *git add .* (yes, there is a dot at the end)
    d. *git status* (optional but to check if the files can be added now, will be green after this step)
    e. *git commit -m "first commit"* (It can be any message but is important)
    f. *git branch -M main*
    g. *git remote add origin git@github.com:AksharGoyal/doctro-rest-info.git* (the last part can be different but is listed in repo page like in the page above)
    h. *git push -u origin main* (when you refresh the github page, you will observe that all the files from your local desktop have been send to your github repo page)
5. Optional: In case you get an error related to publickey, you can go to this link and the first two answers will surely help you rectify that error.
6. Do the same for all the other 3 repos.
7. To proceed further, ensure you have installed Heroku CLI and also have an account.
8. For the rest-doctor-info and rest-exercise-node, open git bash and do the following steps:
    a. *heroku login* (In case it asks you to login)
    b. *heroku create* (we create an "empty website")

c. *git push heroku main* (push the files from github repo to Heroku; it will show the website name it is getting deployed to at the end; will take time)
d. heroku open (opens the website in your default browser)

9. So for my rest-doctor-info, the website name is https://warm-temple-93834.herokuapp.com/ and it shows me a blank white page saying *Cannot GET /*. If I add "docs" after my site, i.e., https://warm-temple-93834.herokuapp.com/docs, I should get the same website I would have on localhost upon npm start:

Same goes for your Heroku website for rest-exercise-node.

10. For your rest-doctor-info Heroku website, you can also *v1/doctor/{id}* to get information regarding a certain doctor, like I have done for doctor id 1




```
{
    id: "1",
    name: "Frank Waiton",
    dob: "1967-07-21",
    specialty: "Cardiology",
  - address: {
        street: "11 Speers Road",
        city: "Oakville",
        province: "Ontario",
        postal_code: "L6J 4Z3"
    }
}
```

11. Now, we are going to deploy the graphql-api on Heroku. In VS Code, First go to your resolvers.js and your loaders.js. Then change the line where you are initialising API_URL to process.env like shown below

```
const {API_URL} = process.env; //'http://localhost:4000/v1';
```

(remember to de-structure your object when initialising)

Then go to your index.js file, and change the object initialisation of *server* to

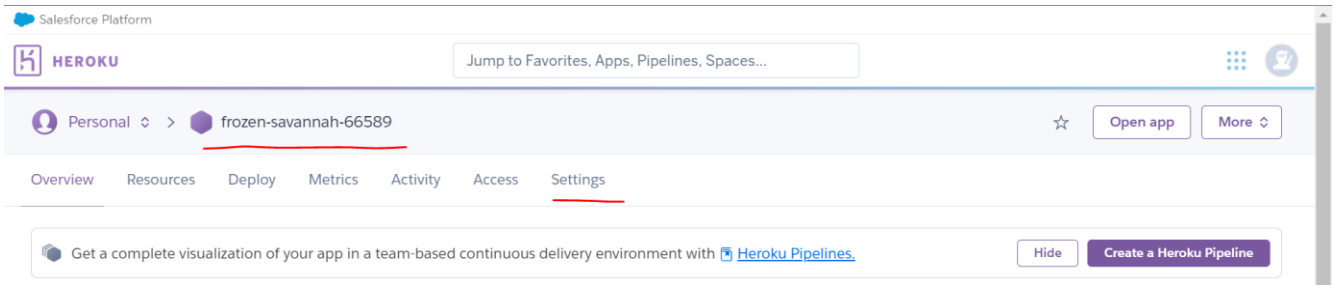
```
const server = new ApolloServer({
  typeDefs,
  resolvers,
  context: () => ({
    loaders: getDataLoaders(),
  }),
  cors: true
});
```

We are just adding cors:true in the end which will help in linking with the api.

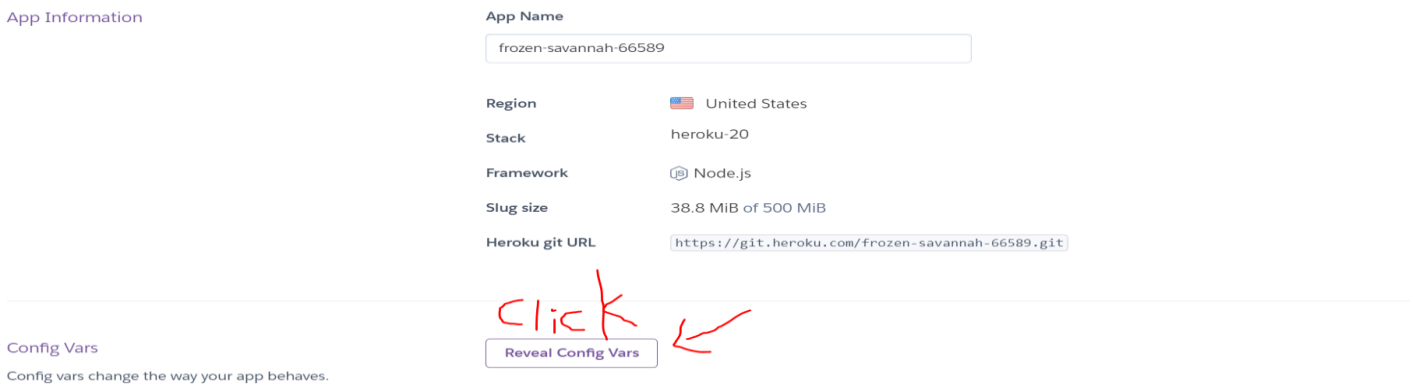
13. Open git bash and do the following commands:

a. git status (to check the status of your files)
b. git pull (to ensure nothing has changed in your repo)
c. git add . (add the updated files)
d. git commit -m “random messages about why you are adding again”
e. git push
f. Heroku create (create a website and do not proceed to pushing into the main branch yet)

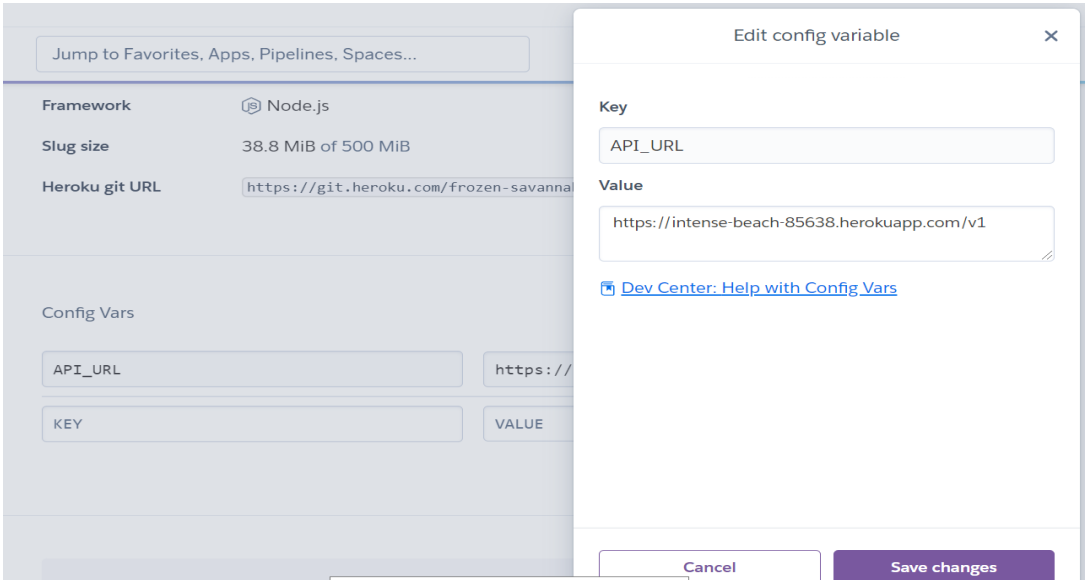
14. For me, my rest-exercise-node website is https://intense-beach-85638.herokuapp.com/ and my graphql website (empty currently) is https://frozen-savannah-66589.herokuapp.com/ . Go to your Heroku dashboard and click on your graphql website. Then go to settings. There, click on Reveal Config Vars



Then you will reach here

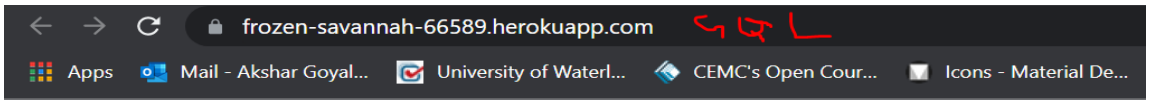


15. In the KEY, type API_URL. In the value, write your Heroku site for rest-exercise-node but with v1 appended to it, i.e., https://intense-beach-85638.herokuapp.com/v1 (in my case).



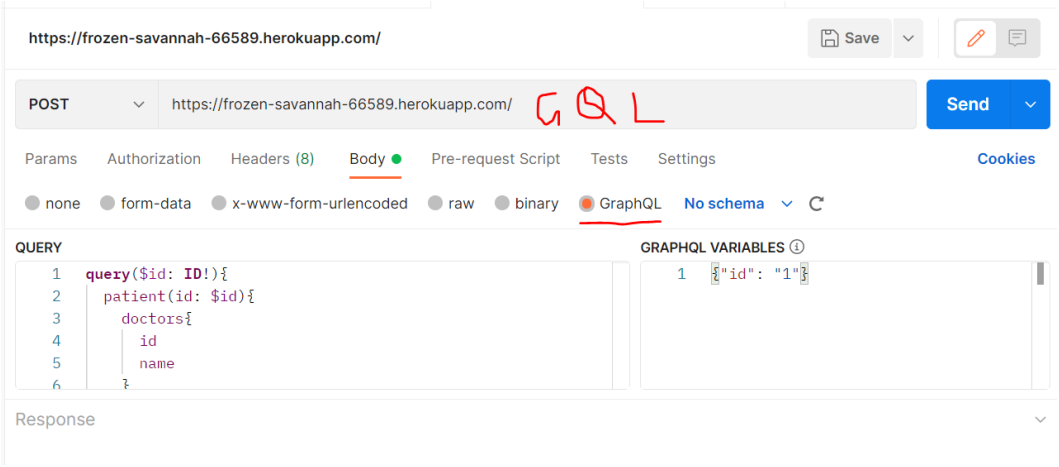
16. Finally, we can do our regular “git push heroku main” and “heroku open”.

17. Now, if you were to go to your graphql website, it will show nothing and that Is totally fine.



GET query missing.

18.Open Postman. We are going to create a POST request so paste the graphql link. Go to Body and make sure you have selected GraphQL.



19. Paste the following query in the QUERY field (you can modify too):

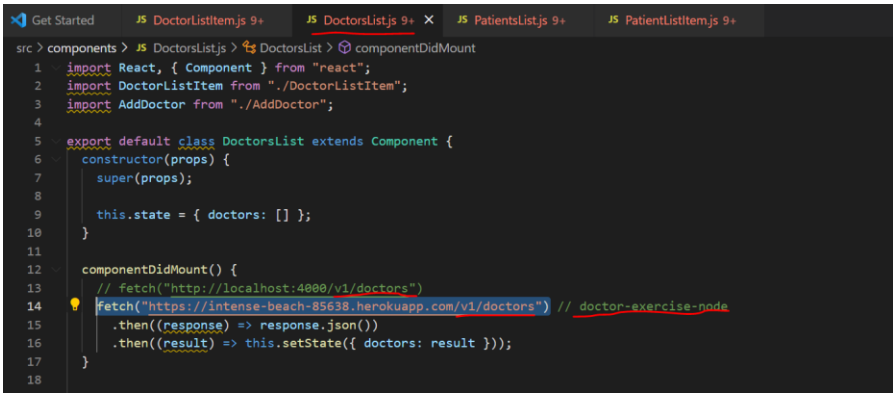
```
query($id: ID!){
  patient(id: $id){
    doctors{
      id
      name
    }
  }
}
```

Paste the following object into GRAPHQL VARIABLES field:

```
{"id": "1"}
```

20. After you press Send, you should get 200 code indicating that your graphql website is connected to the api correctly

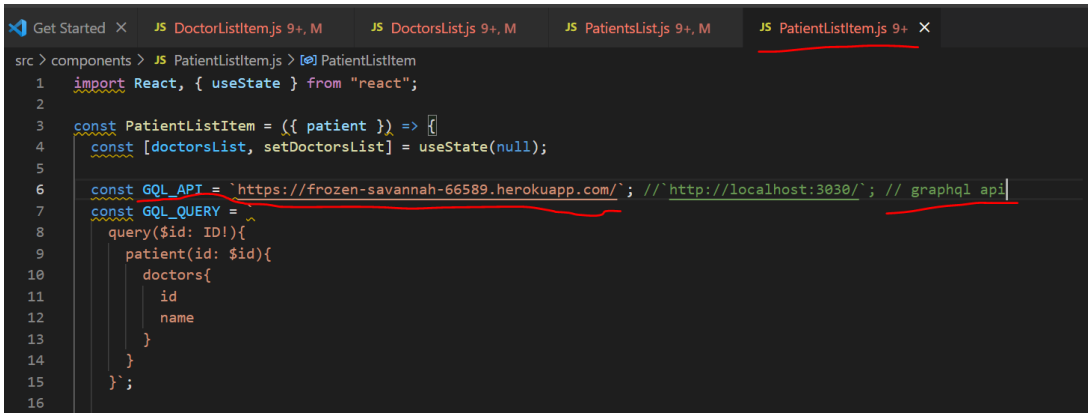
21. Finally, we can now work on the REACT part. So open the React folder in VS Code.
22. In your DoctorList.js, in ComponentDidMount(), change your parameters inside fetch from localhost to the website where you are hosting doctor-exercise-node, as shown below. Make sure to keep */v1/doctors*.



23. Do the same in *fetchPatients()* of *PatientsList.js* but this time, instead of *doctors*, keep *patients* at the end of the url.

24. In *DoctorListItem.js > handleLoadDetails()*; change the localhost part to the name of website where you are hosting rest-doctor-info as shown below

25. Finally, in PatientListItem.js, where we are initialising GQL_API, put in your graphql website url, as show below



26. We have updated the code. So we will first have to do the steps from *git add* to *git push* again. Then, unlike graphql, we can again do *heroku create* and *git push heroku main.* This will take lot of time for building and deploying so I recommend not doing this in urgent situations.
27. Finally you can do heroku open to see your online React App! You can compare it with the one I have made to ensure you got everything right: https://limitless-spire-84932.herokuapp.com/

I really hope that if you follow the steps correctly, you will get the desired outcome for this app and your future project onwards. If you have any questions, feel free to send me a message on Teams or an email on wea.manulife@uwaterloo.ca 😊