

Cheese Shop Sample

This is Cheese Shop. This version of the project is a sample to demonstrate my skills in ReactJS and back-end software development. The full project is still in development and has much more cheese and functionality.

The application displays information about many different cheeses using an API hosted within the project. The data comes from a MongoDB database and is served as an API through an express server. There's graphql middleware to make the API easier to deal with.

Redux is a major part of this application tying all the data together and making it easy to trace errors. The important parts are in

`cheese_shop/src/actions/` and `cheese_shop/src/reducers/`

Setup

Dependencies:

- Latest version of NodeJS
- Latest version of MongoDB
- Latest version of NPM
- A browser
- An internet connection

To run

1. run 'npm install' command in the `root` folder and in `/cheese_shop/` .
2. run the mongod command to start the mongoDB server
3. Execute the `/cheese/scrapper/get_local_cheese.js` to add all the data.
4. Run `/cheese/server/GQLserver.js` to start the API
5. From `/cheese/cheese_shop/` execute `npm run start` from the command line
6. Open a browser and navigate to `localhost:8080/`

OR

Go to the live version [here](#). (This application is hosted on my web server which is reason for the domain name. All data stores are wiped daily.)

Use cases

- View information about various cheeses through a search
- Add cheese to a cart and prepare for purchase
- Server an api or anyone else to view the same cheese information

Where did the data come from?

- The data in this sample can be found in `/scraper/cheese.json` . This sample data has 60 different cheeses in it and lots of information about each.
- The original scraper for the full project is also in `/scraper/` . It

uses [Cherrio](#), streams, and the [MongoDB Client](#) to gather the information and pictures.

- The prices are randomly generated number based around normal cheese prices.

The API

The API uses [mongoose](#) to connect mongoDB and [GraphQL](#) . It is served from the [localhost:4000/graphql](#) but in the official application the data comes from a phsycial server unattached from the host. GraphiQL is enabled here if you want to test some queries.

The basical types are :

```
CheeseType{
  Name
  Link
  Img
  Made
  Country_of_origin
  Family
  Type
  Fat_content
  Calcium_content
  Texture
  Rind
  Colour
  Flavour
}
```

```
Aroma
Vegetarian
Producers
Synonyms
Alternative_spelling
Fat_dry
Price
}
```

```
cartType{
  Name
  Quantity
  Price
  Img
}
```

The basical queries are :

Query/Select

```
getAllCheese{
  CheeseType
}

Search(search:String){
  CheeseType
}
```

```
GetCart{  
  CartType  
}
```

There are other included in `/server/GQLServer` but these are the ones used in this Sample

Mutation/Modifying

```
addItem(Name:String Quantity:Number Price:Number Img:String){  
  cartType  
}  
  
updateItem(Name:String newQuantity:Number){  
  cartType  
}  
  
removeItem(Name: String){  
  cartType  
}
```