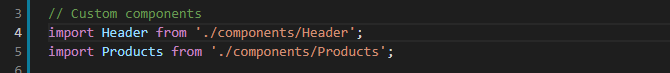
A simple search bar, which will take a string as a query

1. Frontend

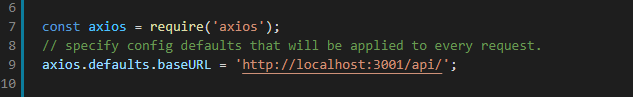
* App.js

Layout components to show header and products list.



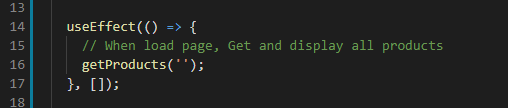
Axios node module that for integrate with server.

When each request, `baseURL` will be prepended to `url` unless `url` is absolute.



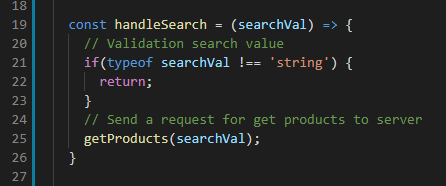
`data` is a list of products to be displayed.

When first load, get products.



Start the search with input value.

Input value is not string, don’t work.



Send the request to get products.

url: will be prefixed Axios base URL to URL and search by name.

Now, backend is based on json server. If you want to change the server(node.js, Laravel), you have to change url(api endpoint)

`setDate` method is react state, data will be update with response data(products).



Render dom

Search bar exist in header, `onSearch` and `cancelSearch` is emit method.

When cancel search, search with empty string so get all products.

And render got products in `Products` component.

No data, show message `Not found`.

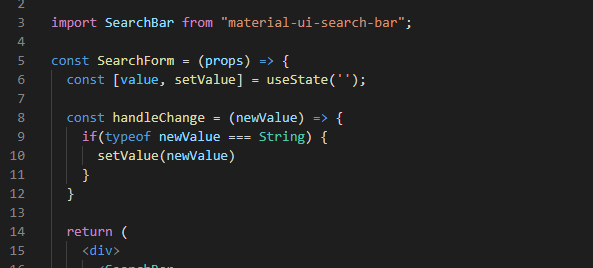


* SearchForm.js

Use SearchBar component from react material ui.

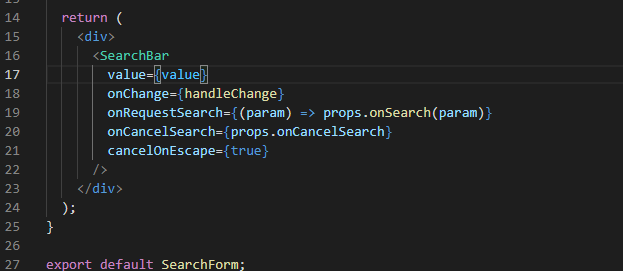
`value` is input value.

`handleChange` is that when input value,



`onChange` method do that when input value, save that to state variable.

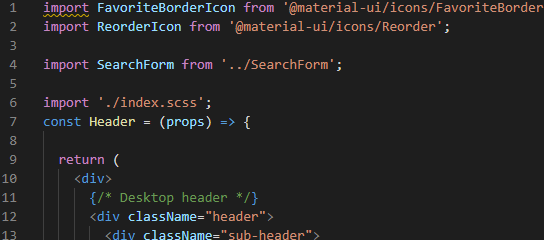
`onRequestSearch`, `onCancelSearch` do that with inputed value(param), emit to parent(Header component) and do search.



* Header.js

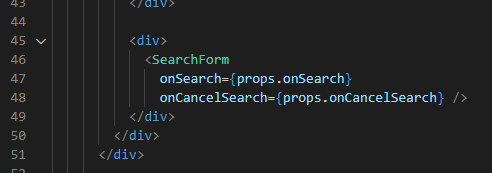
It will be display the header on the page.

Import icons and SearchForm component.

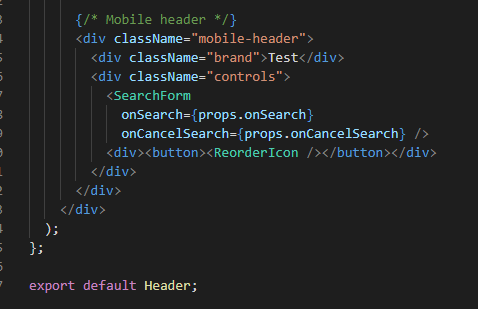


Render SearchFom.

When trigger search event in search form, emit to parent(App component) so can get products by search.



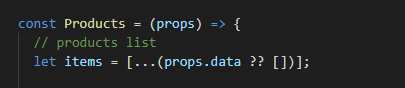
Mobile header for responsive.



* Product.js

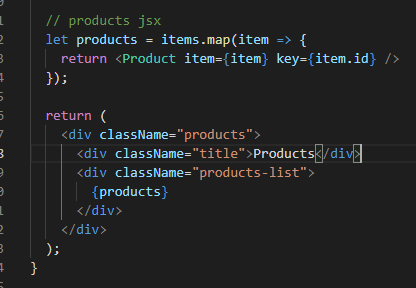
Show Products

`items` is products



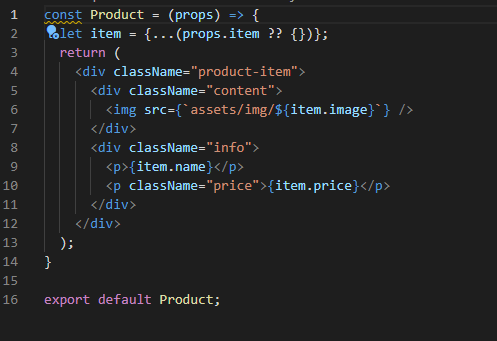
`products` is jsx of product to display products.

Doing for loop of items(products), get jsx of product based on each item(product).



* Product.js

Show product



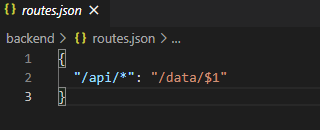
1. Backend

* Route.js

Define route for get data

e.x:

<http://localhost:3000/api/something> to http://localhost:3000/data/something



Data

If url is <http://localhost:3000/data>, return all data.

And to search, localhost/data?{field}\_like=value

`field` is data field for ex, id, name, price, image …

`value` is search value

You can data crud on here

