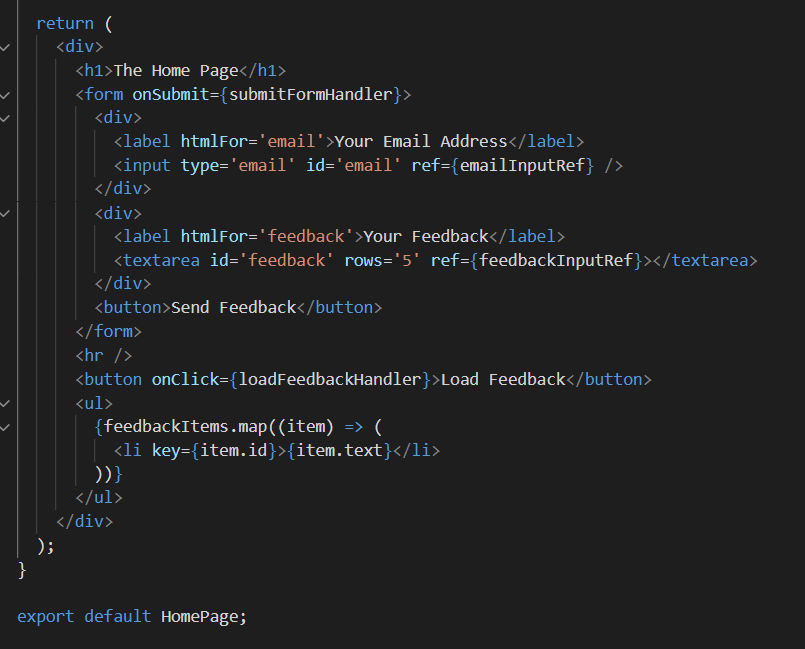


1. To create APIs, create “api” folder inside “pages”. It should be named as “api” only.

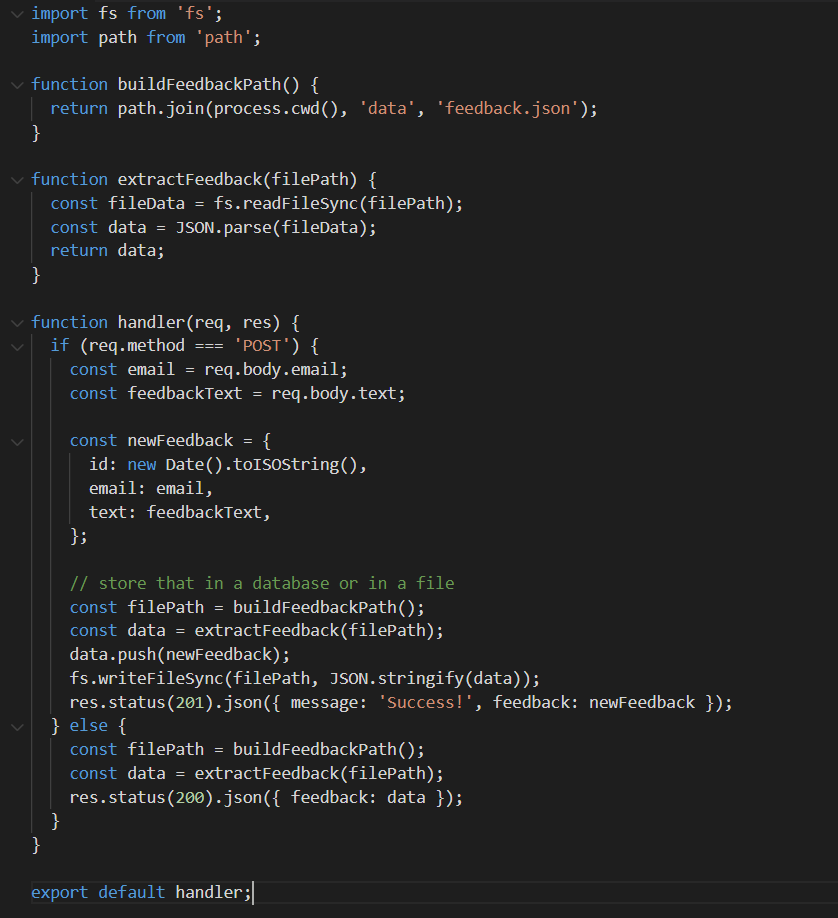
2. Code that we write inside this will be executed in Node.js environment. Just like “getStaticProps” method.

3. function name can be anything, but typically “handler” is given. And code should be similar to how we write in Express.

Left and blow one are index.js file, where are posting data from the form and getting data from the file.

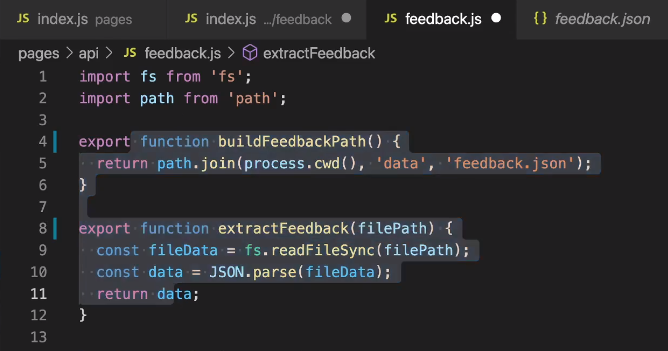
We use fetch to hit the APIs that we have created in “api” folder. On is post and other is get request.

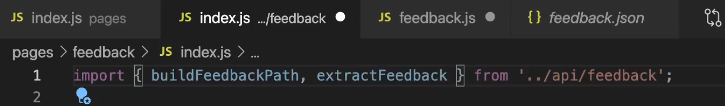
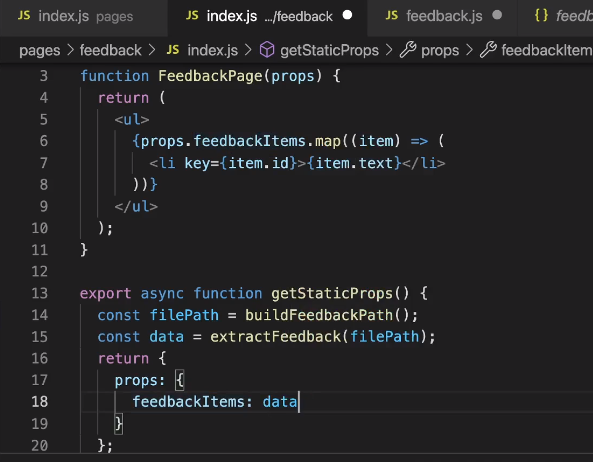


The function that we have exported will be called for any api request that go to “/api/feedback”, which is the file path actually.

We have to segregate the requests using req.method.

**Using API routes inside “getStaticProps()” method**



We have exported the helper functions from “api/feedback.js” and we use them by importing them

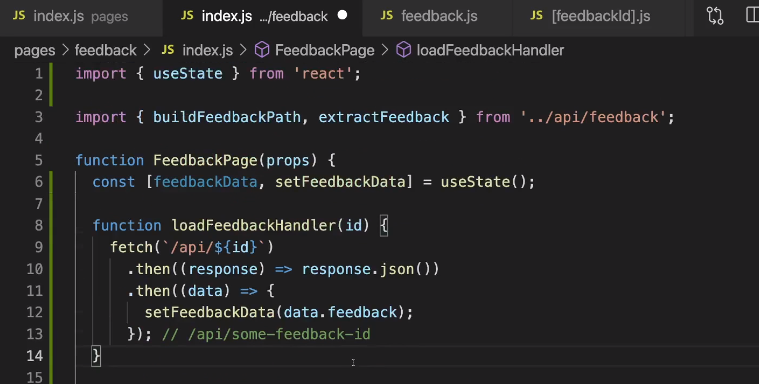
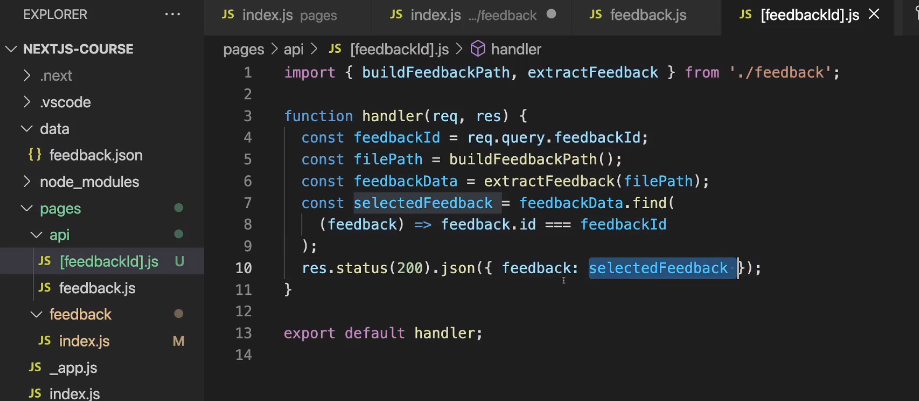
We have imported these functions in index.js. These functions will not be rendered to client side as they are used only in “getStaticProps()” method. So it’s safe to write server side keys.

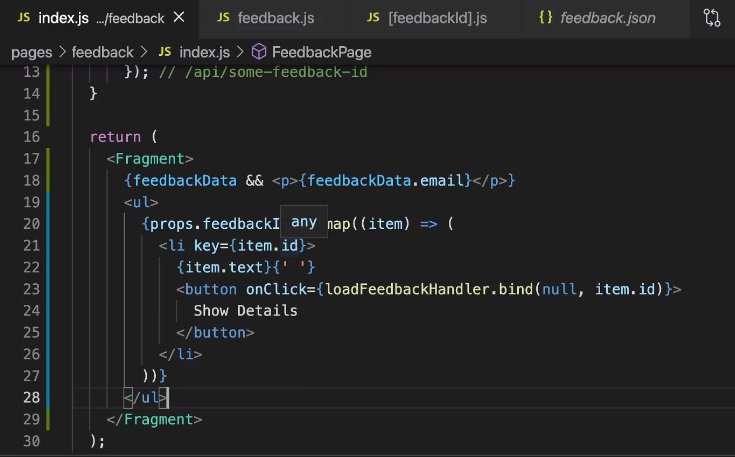
Here we are getting the props via API request and sending them as props to the react method.

**Dynamic API routes**

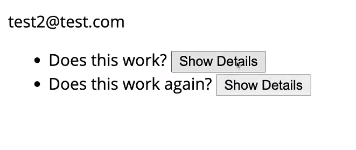
Inside “api” folder also we can create files with [someid].js as file name. This is called when “api/<anything>” is called. We can get the query param using “req.query.someid” and do our logic.

If “api/feedback” is called then the handler inside “feedback.js” will called. “SPECIFIC to GENERIC” is the order.





On click of “Show Details” button, api request is made and the email is displayed on the screen.



Other ways of folder structure

1st image: instead of creating “feedback.js” file, we can create “feedback” folder and create index.js file.

2nd image: [feedbackId].js is moved inside api/feedback folder. Now request to particular id should be made as “api/feedback/some-id”. This is better way of organizing

3rd image: spread operator can be used. “api/something/something” can be used now.

