**Serverless Route Handlers**

Next.js allows applications to be full stack which means running the application both on Font-end and Back-end using the same file based routing system. Next.js allows us to handle HTTP requests and develop back-end functionality without requiring an external server.

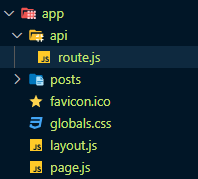
Next.js covers all the features that found in traditional backend server which includes, middleware, parsing, authentication checks and even severless functions that simplify the deployment and scaling of API routes.

In next.js there are 2 different ways to define a rout Handler:

1. The first one is to create file based route handlers. Right within the API folder within the app directory.
2. Second approach is to create a direct route Handler within the app directory itself but there’s a caveat, with this approach you’ve to create a special ‘**route.js’** file that’s going to act as an backend API route but if you want to create a route that’s going to start with / like our page.js does present at the root directory. So, these two (page.js & route.js) can’t interfere in the same fashion if you wanted to create an API route for /posts you wouldn’t be able to have a route.js within the post next to the page.js because next.js won’t know does page.js have to be a regular front-end page or a backend API route. It’s going class between   
   /posts -> going to render a regular page  
   /posts -> that should be an API route.

So In this case we won’t be able to create file structure like this within the app directory.

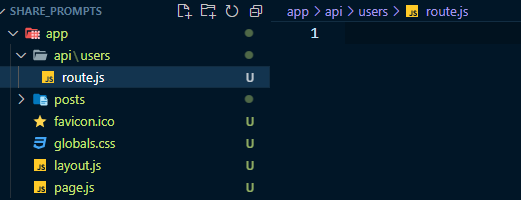
So it’s highly recommend the first approach of creating the API routes meaning don’t create routes right within the app folder rather to keep our code clean and understandable keep all the backend related logic and API endpoints within the API folder.



This separation makes it clear that right here is the backend of your application and everything else in the app is the frontend.

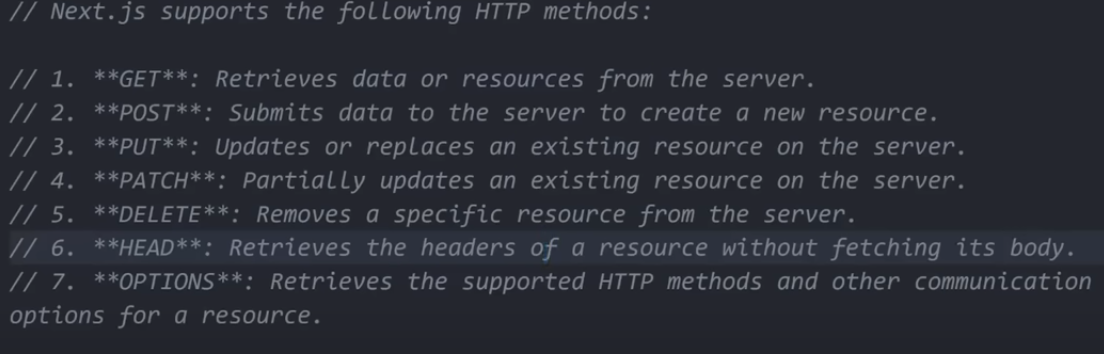
Ex:

Process of creating a simple users API route would look like this:



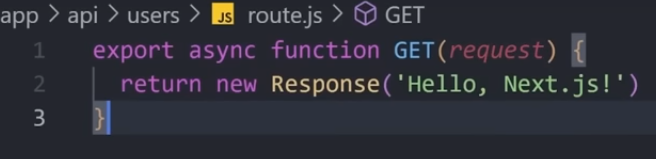
This is now going to act as an API BackEnd route. It will give route like this: *http://localhost:3000/api/users*

**Next.js supports the following HTTP methods:**

****

To create an HTTP methods inside of the route.js file, the process is incredibly straightforward.

You simply need to write like this the require methods to to implementing your backend logic.

****

There is no need to set up any extra additional Express configuration.

**Note:**

Just like, loading, error, route.js is also a special file name and just for that reason it allows us to create those back-end routes.