ROUTES

Defining your URL routes is important

It tells the server what the user wants, so it can respond correctly

You can define routes using methods of the Express app object that correspond to the usual HTTP methods we know

For example, you can use app.get() to handle GET requests

And you guessed it, you can use app.post() to handle POST requests

ROUTES

When you do this, you are also required to specify a callback function

The callback function you create will be executed (or "called") when the server receives a request to the specified URL and HTTP method you defined

In other words, your server will "listen" for requests that match a defined URL and methods, and when it finds a match, it calls the specified callback function

ROUTES

We can write all of our routes in our main express.js file

But the problem is that if your application becomes complex, maintaining your code will become a nightmare

Solution: divide your routes. In this project, we only have dogs so we'll define a separate routing file for all requests relating to our dogs

In other words, we will have our own API route file so maintenance will be a breeze

ROUTES

To define routes, we can use Router middleware given to us by Express

We don't have to use it. But the Router middleware makes it easy to define routes in a modular way

To use it, all we have to do is call the express.Router() function
The express.Router() function creates a new router object
We can then use this router object to define our various API (or URL) endpoints

ROUTES

Let me be clear: we don't have to use the Router() method in our mini project

But, I want to show you how easy it is to use

Enjoy