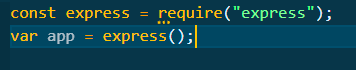
**Lecture 01 (What is express)**

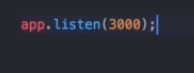
* Express is a node framework
* Is allows to structure our code using nodejs
* Express focuses on web application solely since the scope of nodejs is very broad.
* Its made to help coders write less repetitive code when we are building web applications.

**Lecture 02 (Creating Our First Sever with Express)**

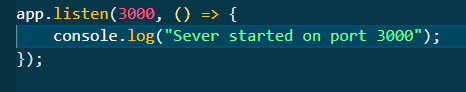
* First you have to initialize npm using npm init
* Then run the command **npm install express**.
* You can add expressjs in your project using the following commands.



* Listen function allows us to listen to an http request on a specific port as shown in the screenshot below



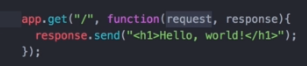
* A port is just a channel that you program your server to listen to, like we choose certain frequencies on radio.
* You can also add a callback function to perform certain tasks when a request is set up.



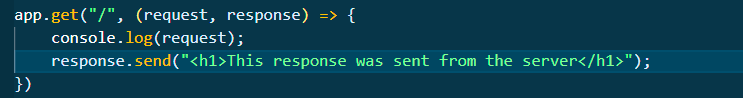
* Localhost3000 is basically the root/homepage of the website.

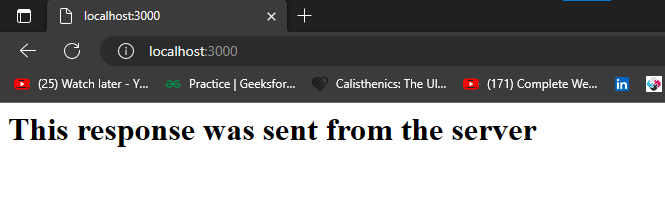
**Lecture 03 (Handling Requests and Responses)**

* Normally when our website is requested it asks to access the home page, expressjs provides the get() function to send responses when a request is made to the server

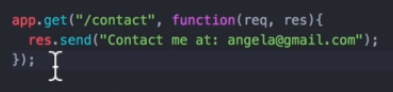


* Here “/” denotes that a request is made to the home of our website and the other argument in the function is a callback function with 2 paraments(objects).
* The first one is the request that contains all the info about the request that was made.
* The other argument is the respond object that makes us able to send a response back using the send() method.
* You can send simple html in the send method and it will be displayed by the browser.





* People also use req for request and res for response.
* The get function basically means that when “/” route is requested you must respond with the callback function.
* If you want to set up a route for /contacts then you can do so by

****

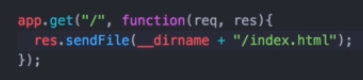
**Lecture 05 (Understanding and working with Routes)**

**Lecture 06 (What we’ll make a Calculator)**

**Lecture 08 (Calculator Setup Challenge Solution)**

**Lecture 09 (Responding to Requests with HTML Files)**

* To send a file in response to an HTTP request we use the sendFile method provided by the expressjs



* In the function above we have given the absolute path. “\_\_dirname” provides us the absolute path it took to reach the current directory and we append the name of the file we want to send in it.
* The value of \_\_dirname for the tutorial we are following is given below.



* We do this cuz when we host our project on the cloud/server we don’t know what will be the exact name of the file directories over there thus \_\_dirname provides us a good and clean way of parsing file paths correctly.

**Lecture 10 (Processing Post Requests with Body Parser)**

* You handle post requests using the app.post() method.
* First arguments is the location where you want to receive the request (determined in the form tag of the HTML file), while the other argument is a callback function that gets executed when a post request is made at the aforementioned path.
* In order to tap into the data sent by the browser we use another npm package called the body parser.
* Body-parser works with express js.
* There are a number of modes of body parser
* When we want to receive the data from an HTML form we use the urlencoded() mode of the body-parser module as shown in the picture below.



* The extended:true is something that body parser requires you to explicitly declare.

**Lecture 12 (Solution to BMI Routing Challenge)**