

Adding dynamic HTML to your Express server

Welcome to this Lab activity

In this lab activity you will learn how to make your html pages dynamic. Dynamic web pages have a very similar structure to static web pages but they have the ability to change their content while your web application is running.

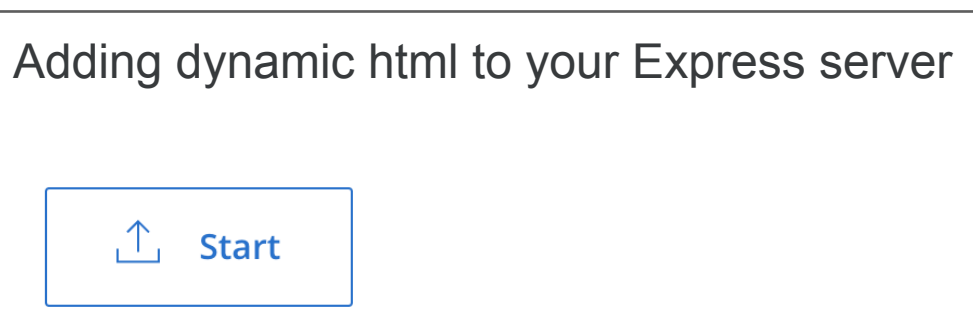
Reflection Point

Imagine you would like to change the title of your html page dynamically? For the purpose of this lab activity you will be using the EJS templating engine introduced in the previous lab.

Start the Lab environment application

It is simple to launch a lab exercise. You only need to click on the button “Start” below the activity title to enter a lab environment.

Let’s explore this lab activity. Go ahead and click on the “Start” button!



Task 1: Build a new Express web server with a dynamic html file

The folder structure has already been partially constructed for you and organised into different topics. For the purpose of this lab, you will be making changes inside the *topic3* folder structure. Let’s get started!

Please do not delete or move any existing folders/files inside the lab environment.

Use the Visual Studio Code Terminal and run the following commands:

- **cd topic3/htmlExpressDynamic**: type this command and press *Enter*. This command will change your working directory to be the *htmlExpressDynamic* folder.
- **npm init**: type this command and press *Enter*. This command will set up a new or existing npm package. You can skip all the npm initialisation questions by pressing *Enter*. Once the npm package file has been created, you can view it inside the *htmlExpressDynamic* directory (*package.json*). The *package.json* file contains a set of information regarding your current node project.
- **npm install express --save**: type this command and press *Enter*. This command will install the *Express* framework within your working directory so that it will be available for you to use.
- **npm install ejs --save**: type this command and press *Enter*. This command will install the *EJS* framework within your working directory so that it will be available for you to use.

Once you run the above Terminal commands, your *package.json* file located inside the *topic3/htmlExpressDynamic* folder should contain both the **Express** and **EJS** modules:



```
{ package.json x
1 {
2   "name": "htmlexpressdynamic",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "author": "",
10  "license": "ISC",
11  "dependencies": {
12    "ejs": "^3.0.1",
13    "express": "^4.17.1"
14  }
15 }
16 }
```

The very next step is to add some code to the *index.js* file located inside the *topic3/htmlExpressDynamic* folder in order to create your web server.

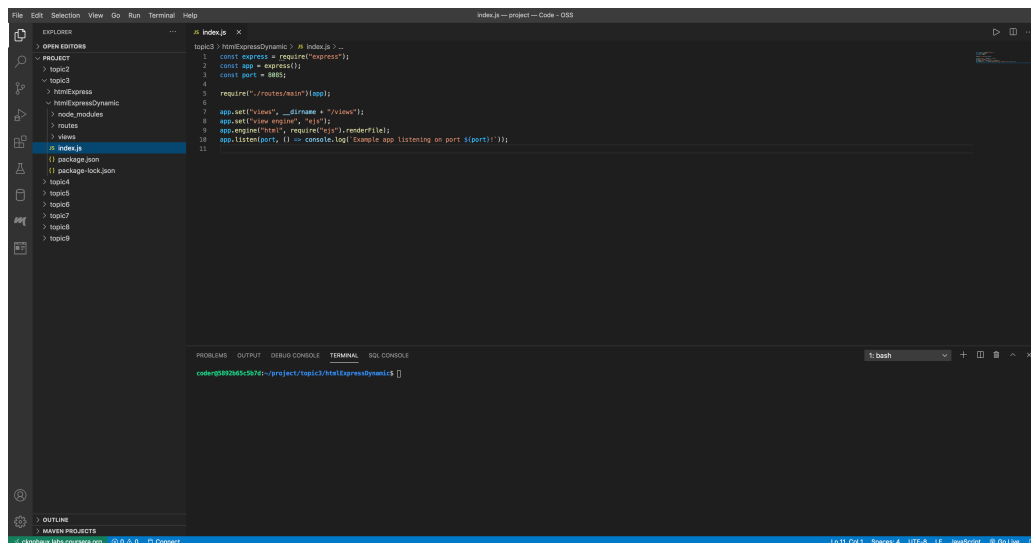
Add the following code to the *index.js* file and remember to save it:

```
const express = require ("express");
const app = express();
const port = 8085;

require("./routes/main")(app);

app.set("views",__dirname + "/views");
app.set("view engine", "ejs");
app.engine("html", require("ejs").renderFile);
app.listen(port, () => console.log( `Example app listening on port ${port}!` ));
```

If you have correctly followed all of the above steps, your environment should be similar to the one below:



Question

- What are the differences between this Express web server and the one you have created in your previous lab activity?

As you can see from the code above, there is no difference at all in the *index.js* file.

The structure you had in the previous lab activity was already configured to work with EJS but you never made use of it.

Let's add one route (the root route) inside the *topic3/htmlExpressDynamic/routes/main.js* file. Add the following code to the *main.js* file inside the *topic3/htmlExpressDynamic/routes/* folder:

```
module.exports = function(app) {  
  
  app.get("/",function(req, res){  
    res.render("index.html", {  
      title: "Dynamic title"  
    });  
  });  
  
}
```

Here you can see the first step towards dynamic pages:

```
res.render("index.html", {  
  title: "Dynamic title"  
});
```

The *res.render()* function has now two arguments: the first one still renders the “index.html” file located inside the “views” directory and the second one passes along an object to the “index.html” document. The object *title* variable, with the string value of “Dynamic title”, will be available inside the “index.html” document.

Let's see how you can now retrieve the *title* value using EJS inside the “index.html” document.

Create one html file inside the *topic3/htmlDynamicExpress/views* folder called *index.html* and add the following html content:

Add the following to the *index.html* file:

```
<!doctype html>  
<html>  
  <head>  
    <title> <%= title %> </title>  
  </head>  
  <body>  
    <h1> This is home page </h1>  
    <p>Homepage This is an example paragraph. Anything in the <strong>body  
    </strong> tag will appear on the page, just like this <strong>p</strong> tag  
    and its contents.</p>  
  </body>  
</html>
```

You have just written your first line of EJS code:

```
<title> <%= title %> </title>
```

The *title* of your “index.html” web page is no longer static. Its value in fact, will depend from the *title* value passed in by the root route object inside the “main.js” file.

In your case the title value is “Dynamic title”.

Now that you have correctly set up your route and html file to support EJS, you can access your application to check the output.

Running the index.js file

Run the *index.js* file with the following Terminal command:

- **node index.js:** type this command and press Enter. The node command, followed by the file name, tells Node.js to execute the content of the file.

The above command will start a web server running on port 8085.

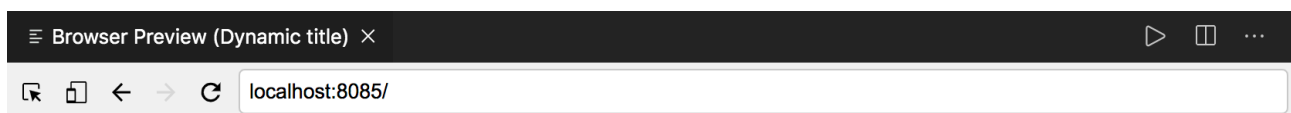
Task 2: Access your server via HTTP

Now that your code is running, serving your application on port 8085, you can access your web page from a browser!

Use the “Browser Preview” plugin to visualise your web application.

If you do not remember how to use the “Browser Preview” plugin, please refer to the “Creating my first Node.js web server” lab instructions.

Type *localhost:8085* on the “Browser Preview” tab and press *Enter* on your keyboard to visualise your web application “root” page:



As you can see the *title* of the page has the value of the root route object passed in. (Dynamic title)

Try to change the *title* value inside the *topic3/htmlExpressDynamic/routes/main.js* file to see if your html title page changes. Remember to save the file before viewing your new application version.

End of Section

Congratulations for completing this section. As long as you have saved your work, your files will remain when you close this lab activity so do not worry about about losing your data. You have successfully created a web application that serves dynamic html files according to your routes.