

Web Frameworks and Full-Stack

Ramiro Gonzalez

Delivery Date : February 15, 2019

At the end of this project one should be able to use express framework, define what a web application and web APIs (Application Programming Interface) is, and understand what constitutes full-stack development.

Background

We will be working with the Linux Command Line and VIM or NANO (text editor). It is assumed that you are running a Ubuntu Distro (Linux Distribution) and have installed node. Express.js runs within node. Express allows client to send request to server only, it is not bidirectional.

Material

- ☐ Computer
- ☐ Internet Connection
- ☐ Ubuntu Linux (Linux Distribution)

1 Tasks

Complete the following tasks. Note that different operating systems may require different steps.

Task 1

This task will introduce to frameworks. You will be installing Express.js a web framework for Node.js. Use the Express.js website for reference.

- Create a new folder where your files will be stored.

1. Create a folder named playground

```
1 $ mkdir playground
2 $ cd playground
```

2. Create .json file. This is needed to save our dependencies.

```
1 $ npm init
```

3. Install <https://expressjs.com/>. The -s saves it to the json file as a dependency.

```
1 $ npm install -s express
```

4. Create a server. Name it server.js

```
1 $ vim server.js
2 var express = require('express')
3 var app = express()
4 app.listen(8080)
```

5. Run server.js and open a browser <http://localhost:8080> or local-Address:8080

```
1 $ nodemon server.js
```

The server is not serving any content.

Task 2

The client-server relationship is where a client request a services or content and a server provides it. Static content as the name implies does not have to be generated, or modified by the server. Static content is CSS, images, or scripts.

- In the playground folder, create an html file named index.html. This is the main page, first file to be served by the server.

```

1 <!doctype html>
2 <head>
3   <title> Sample Page </title>
4 </head>
5 <!-- This is a comment -->
6 <body>
7   <h1> Hello World</h1>
8 </body>
9 </html>

```

- To serve the static file (index.html), we must use the *app.use()* function. Add this to server.js

```

1 var express = require('express')
2 var app = express()
3 app.use(express.static(__dirname))
4 app.listen(8080)

```

`__dirname` is always the directory in which the currently executing.

- Learn the difference between static and dynamic content.

Static web page is just like a poster, while a dynamic web page changes perpetually or when the a client (user) interacts with it.

- Run server.js and open a browser <http://localhost:8080> or localIPaddress:8080

```
1 $ nodemon server.js
```

The server is serving static content (index.html page)

Task 3

Frontend development uses HTML, CSS, and JavaScript as technologies for development, this is the graphical component and user interaction.

HTML - Hyper Text Markup Language is a markup language it is based on the **DOM - Document Object Model**. HTML provides information on how a page will look.

- (Optional) Copying to index.html if using non-gui ubuntu server.

- Download or Open WinSCP
- Find the servers local ip address using ifconfig
- Connect using hostname (localip), username, password
- Open index.html with WinSCP
- Copy and paste

- Open index.html, and add bootstrap by copying the style sheet <link> Go to <https://getbootstrap.com/docs/4.3/getting-started/introduction> , under CSS. It should be as follows.

```
1 <link rel="stylesheet" href="https://stackpath.
    bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css
    " integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/
    j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous
    ">
```

- JQuery is a widely deployed JavaScript library. Some components such as buttons, navigation, and display require [jQuery](#), [cdnjs.com](#) , and [bootstrapCDN](#). Add them to the index.html document.

```
1 <script src="https://code.jquery.com/jquery-3.3.1.slim.
    min.js" integrity="sha384-q8i/X+965
    Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
    abtTE1Pi6jizo" crossorigin="anonymous"></script>
2 <script src="https://cdnjs.cloudflare.com/ajax/libs/
    popper.js/1.14.7/umd/popper.min.js" integrity="sha384-
    U02eT0CpHqdSjQ6hJty5KVphtPhzWj9W01clHTMGa3JDZwrnQq4sF86dIHNDz0W1
    " crossorigin="anonymous"></script>
3 <script src="https://stackpath.bootstrapcdn.com/bootstrap
    /4.3.1/js/bootstrap.min.js" integrity="sha384-
    JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/
    nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></
    script>
```

Copied from <https://getbootstrap.com/docs/4.3/getting-started/introduction> under JS.

- The index.html document should be as follows.

```
1 <!doctype html>
2 <-- Insert Style Sheet <link> here -->
3 <--Insert Scripts here -->
4 <head>
```

```

5     <title> Sample Page </title>
6 <head>
7 <body>
8     <h1> Hello World</h1>
9 </body>
10 </html>

```

Task 4

- (Optional) The following line refreshes the webpage. Add inside <head>

```

1     <meta http-equiv="refresh" content="20"/>

```

- Open a browser localip:8080 or localhost:8080
- Open the index.html on a document editor, every time it is saved the changes will show on the browser every 20 seconds.

- Create a container. Add to index.html inside the <body>
 ”A jumbotron indicates a big box for calling extra attention to some special content or information.”. Add a heading, input box, and button.

```

1 <body>
2 <div>
3     <br>
4     <div class="jumbotron">
5         <h1 class="display-4">How may I help you? </h1>
6         <br>
7         <input class="form-control" placeholder="Enter
8             Something"> </input>
9         <br>
10        <button class="btn btn-danger"> Send Message </
11            button>
12    </div>
13 </div>
14 </body>

```

Changing *line 9* to "btn btn-success" would change the color of the button from red to green.

- Under the jumbotron div (*line 10*) add a new <div>

```
1 <div id="messages">
2 </div>
```

- Add a `<footer>` below *Line 12* `</body>` and add the following script. Generally scripts should be added at the end, so that needed elements are loaded.

```
1 <footer>
2 <script>
3     $((() => {
4         addMessages({name: 'MyName',message : 'Welcome'})
5     })
6     function addMessages(message){
7         $("#messages").append('<h4> ${message.name} </h4>
8         <p> ${message.message} </p>')
9     }
10 </script>
11 </footer>
```

Task 5

Consider the following definitions.

- **Backend development** refers to tasks done by a server, such as calculations and data processing. Programming language such as Python, Ruby or SQL are essential to backend development.
- **Fullstack development** to fullstack development one must understand the technologies used in frontend and backend. Must be proficient at server, networks, user interface, and the business side of things.

Summary

A framework makes functionalities accessible to the programmer, such functionalities may be extended or selectively changed but not modified. `express.js` is a framework designed to build web applications. JavaScript libraries are used to change how elements in an html page are displayed. Refer to this link [Frontend vs. Backend](#)