



SNS COLLEGE OF ENGINEERING



An Autonomous Institution

Coimbatore-107

CS8651-INTERNET PROGRAMMING

UNIT-1

JAVASCRIPT AND BASICS OF MERN STACK

Need for MERN - Server-Less Hello
World - Server Setup - nvm - Node.js - npm.



Server-Less Hello World

- Open up your favorite editor and create an HTML file with a head and body as follows,



Listing 2-1. index.html: Server-less Hello World

```
<!DOCTYPE HTML>
<html>

<head>
  <meta charset="UTF-8" />
  <title>Pro MERN Stack</title>
  <script src=
    "https://cdnjs.cloudflare.com/ajax/libs/react/15.2.1/react.js">
  </script>
  <script src=
    "https://cdnjs.cloudflare.com/ajax/libs/react/15.2.1/react-dom.js">
  </script>
  <script src=
    "https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.23/browser.min.js">
  </script>
</head>
```



```
<body>
  <div id="contents"></div><!-- this is where our component will appear -->
  <script type="text/babel">
    var contentNode = document.getElementById('contents');
    var component = <h1>Hello World!</h1>; // A simple JSX component
    ReactDOM.render(component, contentNode); // Render the component inside ←
                                             the content Node
  </script>
</body>

</html>
```



- Save it as index.html anywhere on your file system. Open it in your browser and check it out.





Server Setup

- **nvm**
- This is the Node Version Manager that makes installation and switching between multiple versions of Node.js easy. This is not a mandatory step.
- If you are using Mac OS or any Linux-based distribution, follow the instructions on nvm's GitHub page at <https://github.com/creationix/nvm>.
- Windows users can follow nvm for Windows (please search for it in your favorite search engine), or install Node.js directly without nvm.



- Node.js
- let's install Node.js using nvm.
- Visit the Node.js website to find out which version of Node.js fits the requirement: the Long Term Support (LTS) or the latest stable version. If you like to use the latest version, just type
- `$ nvm install node`



- don't forget to make the just installed version of node the default. For example,
- `$ nvm alias default 4.5`
- Confirm the version of the node that's been installed as your default by typing the following in a new shell / terminal:
- `$ node --version`



- Note that installing Node.js via nvm will also install npm, the package manager. If you are using Windows and installing Node.js directly, ensure that you have installed a compatible version of npm as well. Confirm this by noting down the version of npm that was installed along with Node.js:
- `$ npm -version`
- With Node.js 4.5, you are likely to see 2.15.9 as the npm version.



- **Project**
- Before we install any third-party packages with npm, it's a good idea to initialize the project.
- With npm, even an application is considered a package. A package defines various attributes of the application.



- But to start with, we need at least a placeholder where these things will be saved. For that, we need to initialize the package.
- Create a directory, say pro-mern-stack, navigate to it in your shell, and type the following:
 - `$ npm init`



- **npm**
- To install anything using npm, the command that you use is `npm install <package>`.
- To start off, and because we need an HTTP server, let's install Express using npm.
- Installation of Express is as simple as
- **`$ npm install express`**



- Go ahead and install it. Once done, you will notice that it prints a tree of other packages Express depends upon. Now, uninstall and install it again with the --save options:
- `$ npm uninstall express`
- `$ npm install express --save`



- **Express**
- It is the best way to run an HTTP server in the Node.js environment.
- Express is a framework that does minimal work by itself; instead, it gets most of the job done by functions called middleware.
- A middleware is a function that takes in an HTTP request and response object, plus the next middleware function in the chain.
- The function can look at and modify the request and response objects, respond to requests, or decide to continue with middleware chain by calling the next function.



- The `express.static` generator function generates one such middleware function.
- This middleware responds to a request by trying to match the request URL with a file under a directory specified by the parameter to the generator function.
- If a file exists, it returns the contents of the file as the response; if not, it chains to the next middleware function.
- The middleware is mounted on the application using the application's `use()` method.



Listing 2-2. server.js: Express Server

```
const express = require('express');

const app = express();
app.use(express.static('static'));

app.listen(3000, function () {
  console.log('App started on port 3000');
});
```




- Let's create a directory named static under the project directory and move index.html (which we created in the previous section) into this directory.
- Now we can start the web server and serve index.html. To start the web-server, do this on the console:
 - **\$ npm start**
- You should see a message saying the application has started on port 3000. Open up your browser and type `http://localhost:3000` in the URL bar. You should see the same Hello World page.



Thank You