Node.js Backend Template

⚙️ Setup the project 🛠️

1. *Initialize the project*

```
npm init
```

2. *Install dependencies*

```
npm i express ejs express-ejs-layouts
```

3. *Install development dependencies*

```
npm i --save-dev nodemon
```

4. *In package.json change to below* 👇

```
"main": "server.js",
"scripts": {
  "start": "node server.js",
  "devStart": "nodemon server.js"
 }
```

🔌 Set up the server 🗄️

1. *Create file* `"server.js"`

2. *Import Express, Express Layouts, setup app*

```
const express = require('express');
const expressLayouts = require('express-ejs-layouts');
const app = express();
```

3. *Set 👁‍🗨 view engine, create & set path for "views" directory*

```
app.set('view engine', 'ejs');
app.set('views', __dirname + '/views');
```

4. *Create "layouts" folder for template HTML & "public" folder for static files*

```
app.set('layout', 'layouts/layout');
app.use(expressLayouts);
app.use(express.static('public'));
```

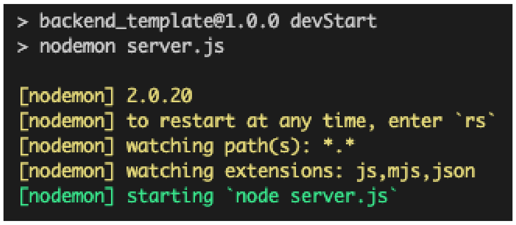
5. *Set app to 🎧 Listen to pull port from "env" when deploy OR port:5500 for development*

```
app.listen(process.env.PORT || 5500);
```

6. *Run 🏃 server to check*

`npm run devStart` or `nodemon server.js`

7. *Server is running! 🏃💨*



```
> backend_template@1.0.0 devStart
> nodemon server.js

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
```

🎛️ Set up Routes or Controller in "*MVC*" 🕹️

1. *Create "routes" folder to which contains controllers & Create first route "*index.js*"

2. Import `express.Router()` from Express & set "router" variable

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

3. Create Home route & send message with response

```
Router.get('/', (req, res) => {
   res.send('Hello from the server!');
});
```

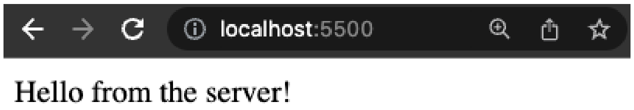
4. Export "Router" to be used in "server.js"

```
module.exports = Router;
```

5. Import "Router" to the server from "index.js" in "routes" folder

```
const indexRouter = require('./routes/index');
```

6. ✅ Check on `http://localhost:5500` 👌 🥳



Hello from the server!

🛣️ Intrigating "routes" with "views" 👁️‍🗨️

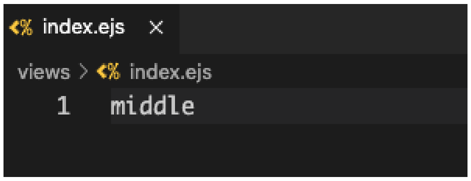
1. Create "layout" folder inside "views" folder & create "layout.ejs"

2. Add HTML boilerplate in "layout.ejs" by command `! + tab` in vsCode

3. Add "EJS" template inside the body tag for other pages to use this template and add content in "ejs" tag

```
<body>
   Bofore
   <br>
      <%- body %>
   <br>
   After
</body>
```

4. In "views" folder Create "index.ejs" & put just `middle`

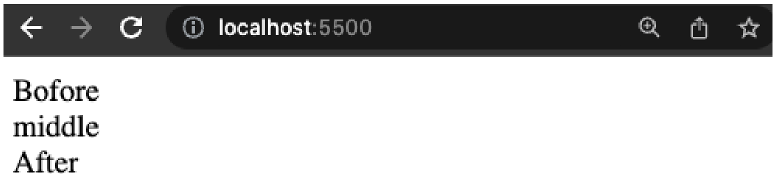


```
middle
```

5. In "index.js" render 'index' page

```
Router.get('/',(req, res) => {
   res.render('index')
});
```

6. Check on `http://localhost:5500` 👏



Before
middle
After

🔐 Create 'dotenv' file to store secret info like password, API keys... 🔑

1. Install "dotenv"

```
npm i --save -dev dotenv
```

2. Create '.env' file & add info

```
DATABASE_URL=mongodb+srv://username:password@yourCluster.xc3cs9i.mongodb.net/project?retryWrites=true&w=majority
```

3. Import ".env" to the server (check if the server is in dev mode before loading ".env" variable)

```
if (process.env.NODE_ENV !== > 'production'){
   require('dotenv').config();
}
```

🗃️ Intrigate Model (MongoDB) to the server 🗄️

1. Install "Mongoose"

```
npm i mongoose
```

2. Setup connection to the Database

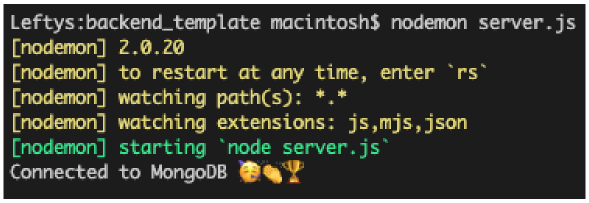
> NOTE: *use "dotenv" file to keep secret infomation outside of the exposed files*

```
const mongoose = require('mongoose');
mongoose.connect(process.env.DATABASE_URL)
```

3. Test connection to the Database

```
const db = mongoose.connection;
db.on('error', error => console.log(error))
db.once('open', () => console.log('Connected to MongoDB 🥳👏🏆 '));
```

4. Run the server and check the log 🤞🤩



```
Leftys:backend_template macintosh$ nodemon server.js
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
Connected to MongoDB 🥳👏🏆
```