

Lesson 09 Demo 01

Demonstrating the Configuration Commands

Objective: To demonstrate the configuration commands in Express.js for effective application setup.

Tools Required: Visual Studio, Express.js, and Node.js

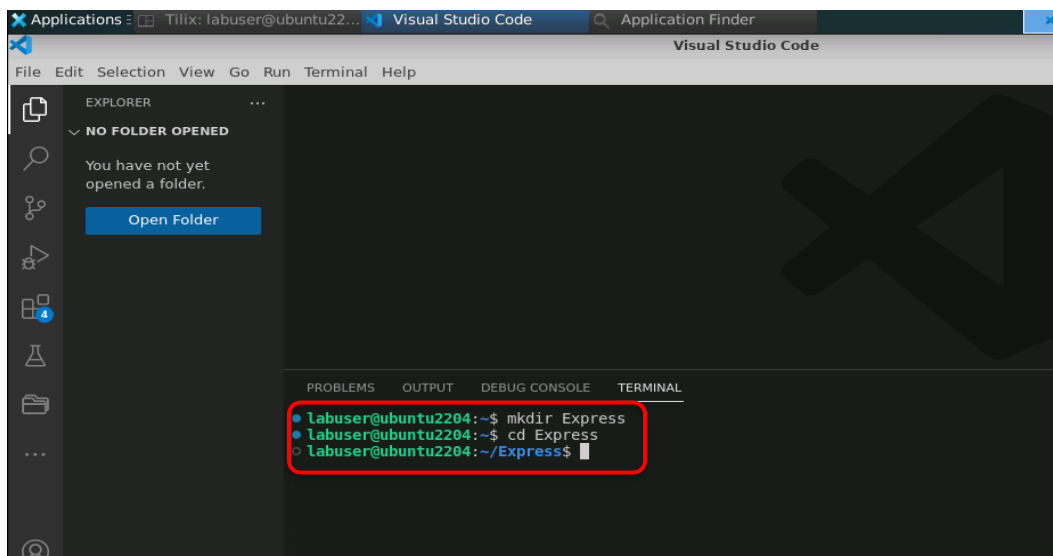
Prerequisites: Knowledge of JavaScript and Node.js

Steps to be followed:

1. Set up the working directory
2. Use `app.set()` and `app.get()` methods in Express.js
3. Use `app.enable()` and `app.disable()` methods in Express.js
4. Use `app.enabled()` and `app.disabled()` methods in Express.js

Step 1: Set up the working directory

- 1.1 Open the VS code and create a folder named **ExpressJS** using the command **mkdir Express** and then use **cd Express** to change the current working directory

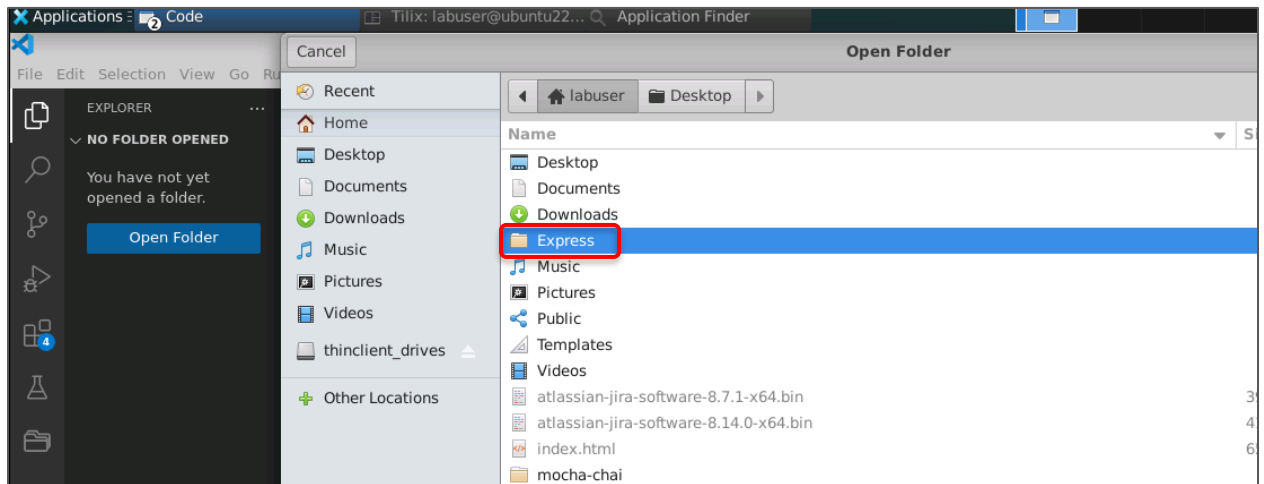


The screenshot shows the Visual Studio Code interface with a terminal window open at the bottom. The terminal displays the following commands and their output:

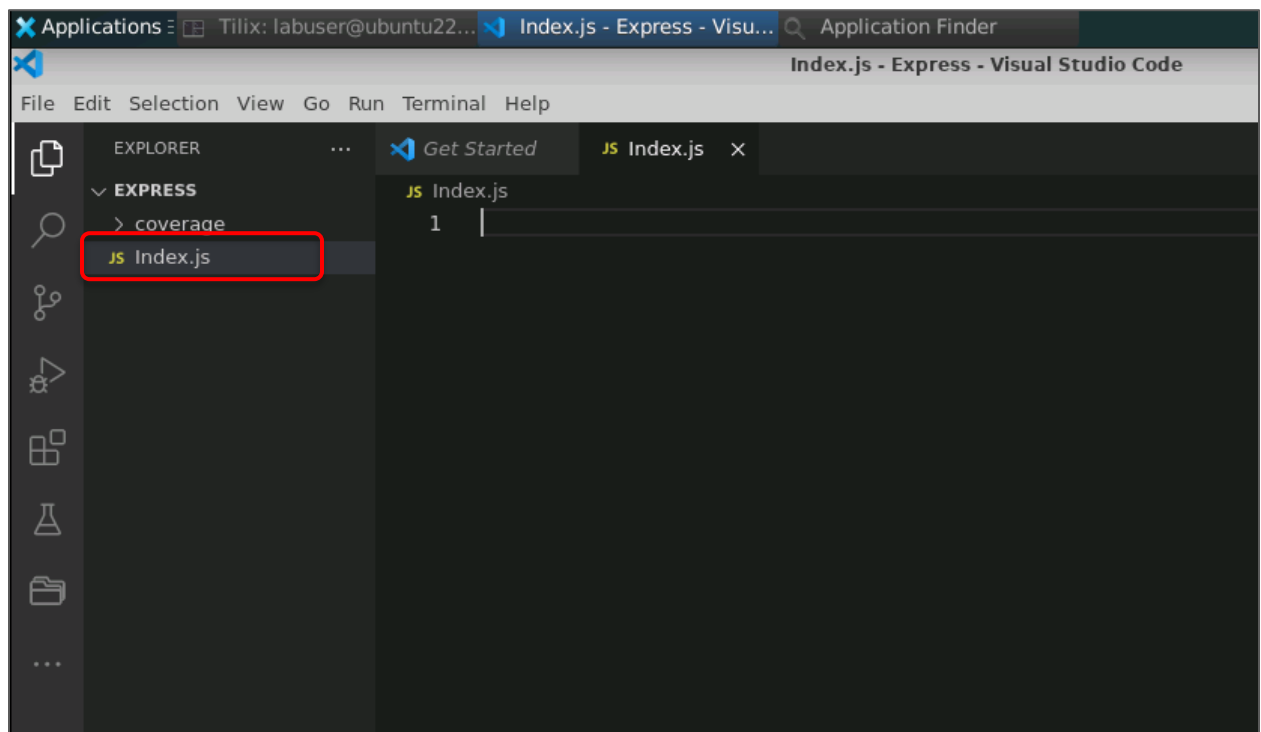
```
labuser@ubuntu2204:~$ mkdir Express
labuser@ubuntu2204:~$ cd Express
labuser@ubuntu2204:~/Express$
```

The terminal output is highlighted with a red box. The Explorer panel on the left shows "NO FOLDER OPENED" and a button to "Open Folder".

1.2 Open the **Express** folder in VS code



1.3 Create an **index.html** file in that folder



Step 2: Use app.set() and app.get() methods in Express.js

2.1 Add the following code in the **index.js** file:

```
var express = require('express');
var app = express();

console.log(app.get('title')) // undefined

app.set('title', 'Configuration Commands.')

console.log(app.get('title'))
```



```
JS index.js x
JS index.js > ...
1  var express = require('express');
2  var app = express();
3
4  console.log(app.get('title')) // undefined
5
6  app.set('title', 'Configuration Commands.')
7
8  console.log(app.get('title'))
9  |
```

2.2 Run the **node index.js** command in the terminal and check the output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  SQL CONSOLE
● labuser@ubuntu2204:~/Expressjs$ node index.js
undefined
Configuration Commands.
○ labuser@ubuntu2204:~/Expressjs$
```

Step 3: Use app.enable() and app.disable() methods in Express.js

3.1 Replace the following code in the **index.js** file:

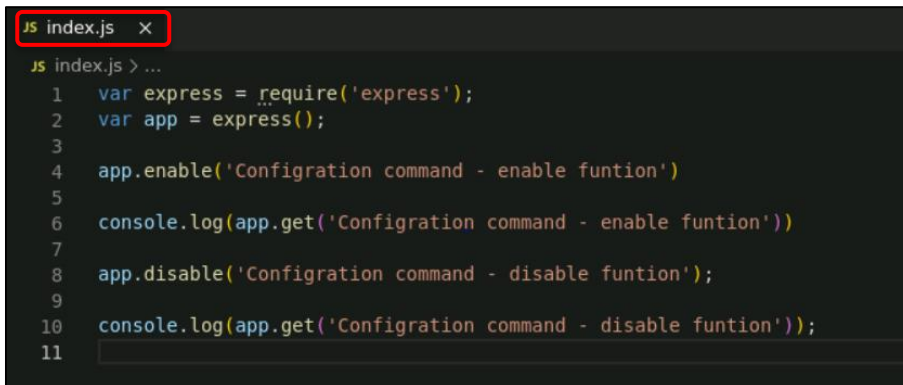
```
var express = require('express');
var app = express();

app.enable('Configuration command - enable funtion')

console.log(app.get('Configuration command - enable funtion'))

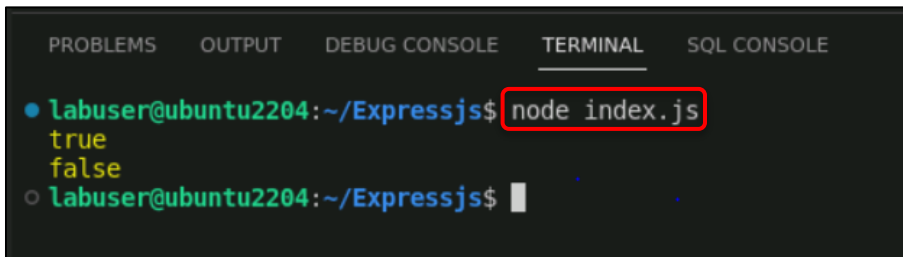
app.disable('Configuration command - disable funtion');
```

```
console.log(app.get('Configuration command - disable funtion'));
```



```
JS index.js x
JS index.js > ...
1  var express = require('express');
2  var app = express();
3
4  app.enable('Configuration command - enable funtion')
5
6  console.log(app.get('Configuration command - enable funtion'))
7
8  app.disable('Configuration command - disable funtion');
9
10 console.log(app.get('Configuration command - disable funtion'));
11
```

3.2 Run the **node index.js** command in the terminal and check the output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  SQL CONSOLE
• labuser@ubuntu2204:~/Expressjs$ node index.js
true
false
○ labuser@ubuntu2204:~/Expressjs$
```

Step 4: Use app.enabled() and app.disabled()

4.1 Replace the following code in the **index.js** file:

```
var express = require('express');
var app = express();
```

```
console.log(app.disabled('Configuration command - disabled funtion')) // true
```

```
console.log(app.enabled('Configuration command - enabled funtion')) // false
```

```
app.enable('Configuration command - disabled funtion')
```

```
console.log(app.disabled('Configuration command - disabled funtion')) // false
```

```
app.enable('Configuration command - enabled funtion')
```

```
console.log(app.enabled('Configuration command - enabled funtion')) // true
```



```
1  var express = require('express');
2  var app = express();
3
4  console.log(app.disabled('Configuration command - disabled funtion')) // true
5
6  console.log(app.enabled('Configuration command - enabled funtion')) // false
7
8  app.enable('Configuration command - disabled funtion')
9
10 console.log(app.disabled('Configuration command - disabled funtion')) // false
11
12 app.enable('Configuration command - enabled funtion')
13
14 console.log(app.enabled('Configuration command - enabled funtion')) // true
15
```

4.2 Run the command **node index.js** and check the output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  SQL CONSOLE
labuser@ubuntu2204:~/Expressjs$ node index.js
true
false
false
true
labuser@ubuntu2204:~/Expressjs$
```

By following these steps, you have successfully demonstrated and implemented key configuration commands in Express.js, enhancing your understanding of application setup and management.