

# *Calculator Setup Challenge*

Before we start creating our Calculator website, we'll need to set up a new project. Follow the steps below using your Hyper Terminal to complete this challenge:

- Make a new folder called Calculator on your Desktop
- Change Directory to this new folder
- Inside the Calculator folder, create a new file called **calculator.js**
- Set up a new NPM package
- Open the project folder in Atom
- Using NPM install the **express** module
- Require express in your calculator.js
- Setup express
- Create a root route get method with app.get()
- Send the words **Hello World** from the root route as the response
- Spin up our server on port **3000** with app.listen
- Run server with **nodemon**

## **Let's start:**

Commands:

```
mkdir Calculator
```

```
cd calculator
```

```
nul > calculator.js
```

```
npm init
```

```
npm install express
```

```
npm install -g nodemon
```

```
//jshint esversion:6
```

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

```
const app = express();
```

```

app.get("/",function(req, res){
    //console.log(req);
    res.send("Hello, World!");
});

```

```

app.listen(3000, function(){
    console.log("Server started on port 3000.");
});

```

```

    pauseOnConnect: false,
    noDelay: true,
    keepAlive: false,
    keepAliveInitialDelay: 0,
    httpAllowHalfOpen: false,
    timeout: 0,
    maxHeadersCount: null,
    maxRequestsPerSocket: 0,
    _connectionKey: '6:::3000',
    [Symbol(IncomingMessage)]: [Function: IncomingMessage],
    [Symbol(ServerResponse)]: [Function: ServerResponse],
    [Symbol(kCapture)]: false,
    [Symbol(async_id_symbol)]: 5,
    [Symbol(http.server.connections)]: ConnectionsList {},
    [Symbol(http.server.connectionsCheckingInterval)]: Timeou
t {
  _idleTimeout: 30000,
  _idlePrev: [TimersList],
  _idleNext: [TimersList],
  _idleStart: 246,
  _onTimeout: [Function: bound checkConnections],
  _timerArgs: undefined,
  _repeat: 30000,
  _destroyed: false,
  [Symbol(refed)]: false,
  [Symbol(kHasPrimitive)]: false,
  [Symbol(asyncId)]: 4,
  [Symbol(triggerId)]: 1
},
  [Symbol(kUniqueHeaders)]: null
},
  parser: HTTPParser {
    '0': null,
    '1': [Function: parserOnHeaders],
    '2': [Function: parserOnHeadersComplete],
    '3': [Function: parserOnBody],
    '4': [Function: parserOnMessageComplete],
    '5': [Function: bound onParserExecute],
    '6': [Function: bound onParserTimeout],
    _headers: [],
    _url: '',
    socket: [Circular *1],
    incoming: [Circular *2],
    outgoing: null,
    maxHeaderPairs: 2000,
    _consumed: true,
    onIncoming: [Function: bound parserOnIncoming],

```

```
    },  
    [Symbol(kHeadersCount)]: 28,  
    [Symbol(kTrailers)]: null,  
    [Symbol(kTrailersCount)]: 0  
  }  
^C  
C:\Users\M.A Computer\Desktop\my-express-server>node server.js  
Server started on port 3000  
^C  
C:\Users\M.A Computer\Desktop\my-express-server>node server.js  
Server started on port 3000  
  
C:\Users\M.A Computer\Desktop\Calculator>cd..  
  
C:\Users\M.A Computer\Desktop> mkdir Calculator  
  
C:\Users\M.A Computer\Desktop>cd calculator  
  
C:\Users\M.A Computer\Desktop\Calculator> nul > calculator.js  
Access is denied.  
  
C:\Users\M.A Computer\Desktop\Calculator>npm init  
This utility will walk you through creating a package.json file.  
It only covers the most common items, and tries to guess sensible defaults.  
  
See `npm help init` for definitive documentation on these fields  
and exactly what they do.  
  
Use `npm install <pkg>` afterwards to install a package and  
save it as a dependency in the package.json file.  
  
Press ^C at any time to quit.  
package name: (calculator)  
version: (1.0.0)  
description: A simple calculator website  
entry point: (calculator.js)  
test command:  
git repository:  
keywords:  
author: Muhammad Ahmad  
license: (ISC)  
About to write to C:\Users\M.A Computer\Desktop\Calculator\package.json:
```

```
{
  "name": "calculator",
  "version": "1.0.0",
  "description": "A simple calculator website",
  "main": "calculator.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Muhammad Ahmad",
  "license": "ISC"
}
```

Is this OK? (yes)

C:\Users\M.A Computer\Desktop\Calculator> Atom .

C:\Users\M.A Computer\Desktop\Calculator> npm install express

added 57 packages, and audited 58 packages in 8s

7 packages are looking for funding  
run `npm fund` for details

found 0 vulnerabilities

C:\Users\M.A Computer\Desktop\Calculator>node calculator.js

Server started on port 3000.

```
<ref *2> IncomingMessage {
  _readableState: ReadableState {
    objectMode: false,
    highWaterMark: 16384,
    buffer: BufferList { head: null, tail: null, length: 0 },
    length: 0,
    pipes: [],
    flowing: null,
    ended: false,
    endEmitted: false,
    reading: false,
    constructed: true,
    sync: true,
    needReadable: false,
    emittedReadable: false,
    readableListening: false,
    resumeScheduled: false,
    errorEmitted: false,
```

```
^C
C:\Users\M.A Computer\Desktop\Calculator> npm install -g nodemon

changed 32 packages, and audited 33 packages in 8s

3 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\M.A Computer\Desktop\Calculator> node calculator.js
Server started on port 3000.
^C
C:\Users\M.A Computer\Desktop\Calculator>

C:\Users\M.A Computer\Desktop\Calculator> nodemon calculator.js
[nodemon] 2.0.19
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node calculator.js`
Server started on port 3000.
█
```