

# Blockchain

## 02 Build the Blockchain – Blocks:

### Set Up the Blockchain Application:

```
C:\Users\Dell>cd C:\Users\Dell\Desktop\BlockChain

C:\Users\Dell\Desktop\BlockChain>cd course II

C:\Users\Dell\Desktop\BlockChain\Course II>npm init -y
Wrote to C:\Users\Dell\Desktop\BlockChain\Course II\package.json:

{
  "name": "course-ii",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

C:\Users\Dell\Desktop\BlockChain\Course II>npm i nodemon --save-dev

added 32 packages, and audited 33 packages in 2s

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

found 0 vulnerabilities

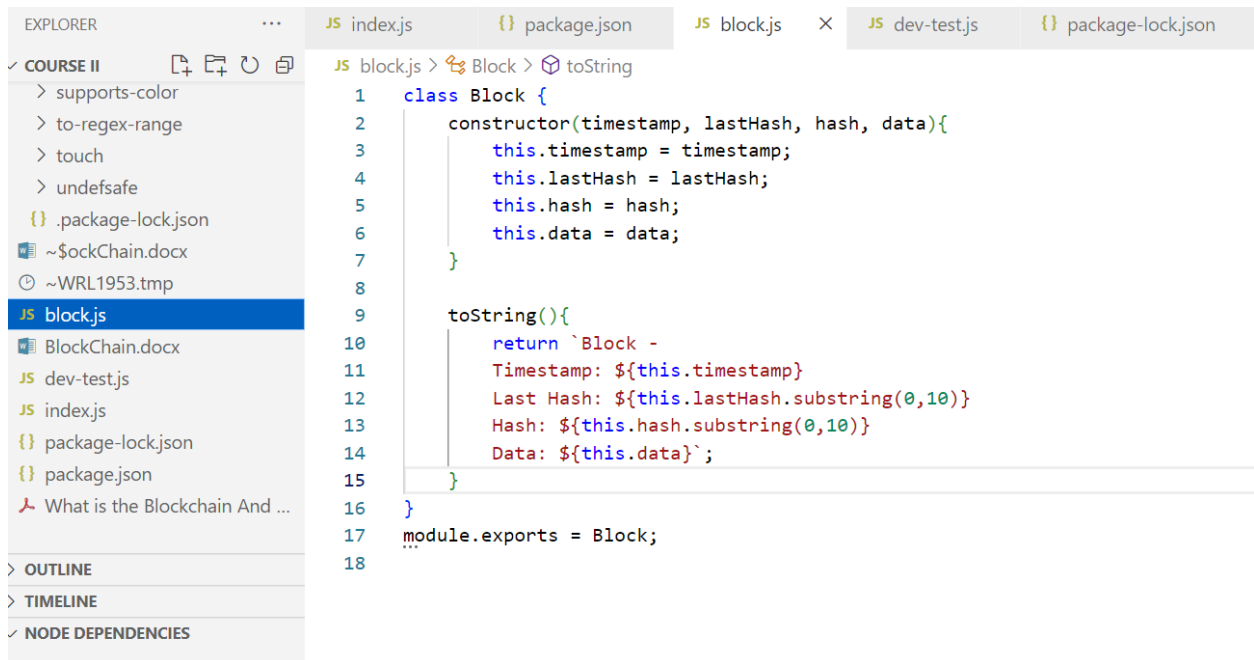
C:\Users\Dell\Desktop\BlockChain\Course II>_
```

# Blockchain

## Create a Block:

### Block

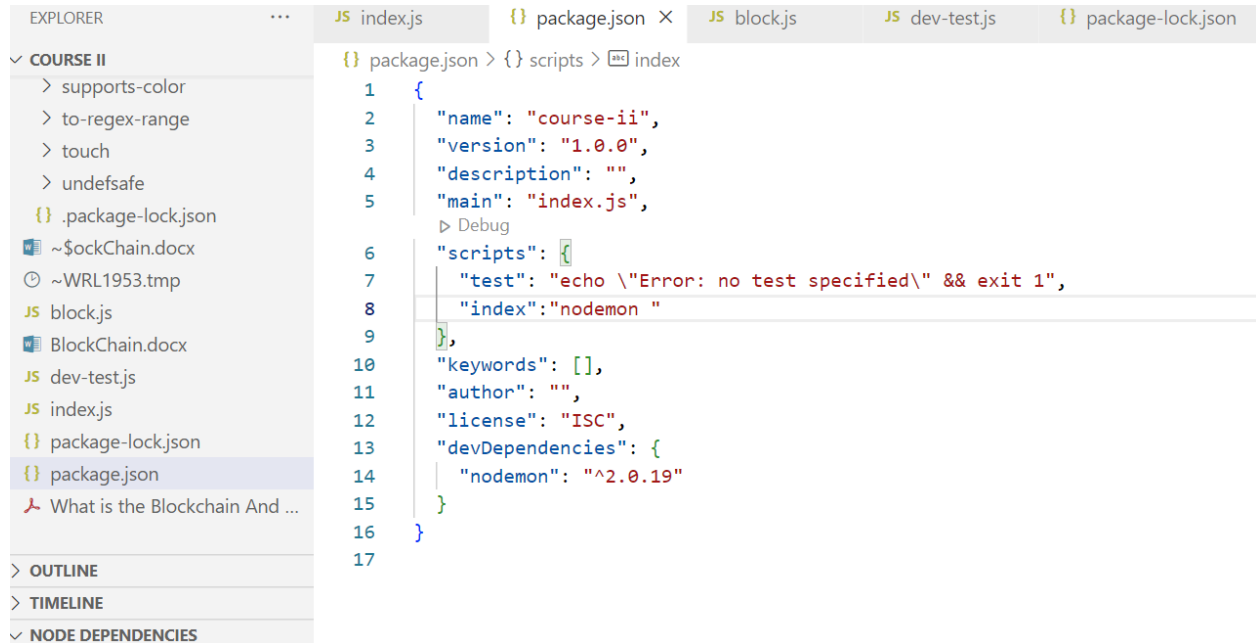
- Timestamp in milliseconds.
- lastHash - the hash of the block before it.
- hash - based on its own data.
- The data to store.



The screenshot shows the Visual Studio Code editor with the file explorer on the left and the code editor on the right. The file explorer shows a project named 'COURSE II' with various files and folders. The code editor shows the 'block.js' file with the following code:

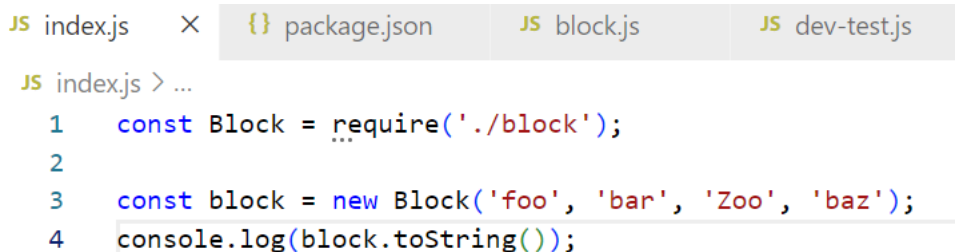
```
1 class Block {
2   constructor(timestamp, lastHash, hash, data){
3     this.timestamp = timestamp;
4     this.lastHash = lastHash;
5     this.hash = hash;
6     this.data = data;
7   }
8
9   toString(){
10    return `Block -
11      Timestamp: ${this.timestamp}
12      Last Hash: ${this.lastHash.substring(0,10)}
13      Hash: ${this.hash.substring(0,10)}
14      Data: ${this.data}`;
15  }
16 }
17 module.exports = Block;
18
```

# Blockchain



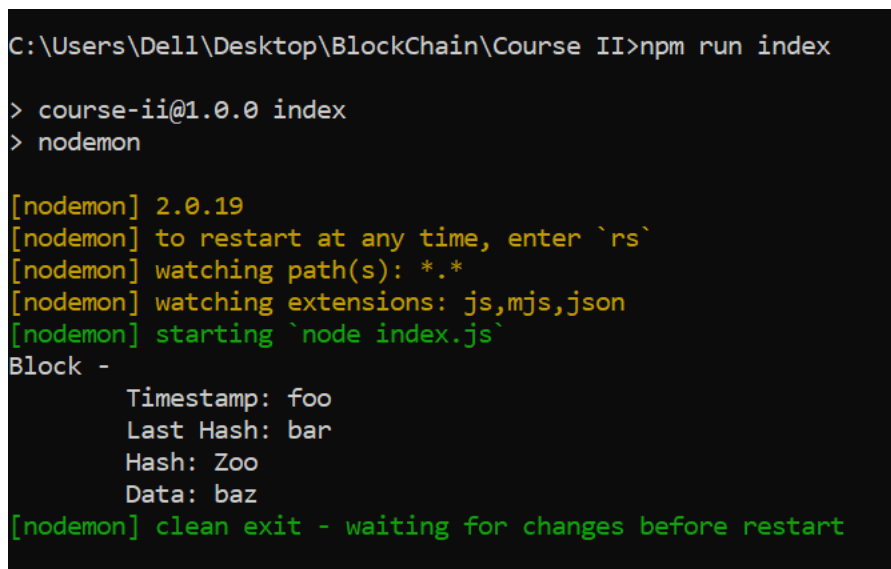
The screenshot shows the VS Code interface. The Explorer sidebar on the left lists files for 'COURSE II', including 'package.json'. The main editor area displays the content of 'package.json'.

```
package.json > {} scripts > index
1  {
2    "name": "course-ii",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1",
8      "index": "nodemon "
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "devDependencies": {
14     "nodemon": "^2.0.19"
15   }
16 }
```



The screenshot shows the VS Code editor with the 'index.js' file open. The code defines a 'Block' class and creates an instance 'block'.

```
1  const Block = require('./block');
2
3  const block = new Block('foo', 'bar', 'Zoo', 'baz');
4  console.log(block.toString());
```



The screenshot shows a terminal window with the following output:

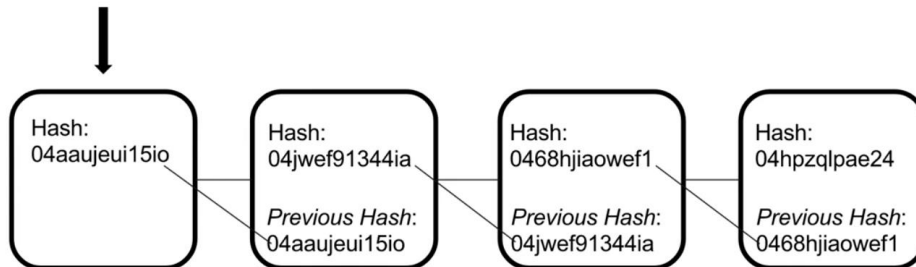
```
C:\Users\Dell\Desktop\BlockChain\Course II>npm run index
> course-ii@1.0.0 index
> nodemon

[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 index.js`
Block -
  Timestamp: foo
  Last Hash: bar
  Hash: Zoo
  Data: baz
[nodemon] clean exit - waiting for changes before restart
```

# Blockchain

## Genesis Block:

### Genesis Block



```
static genesis(){  
  return new this('Genesis time', '-----', 'f1r57-h45h', []);  
}
```

```
const block = new Block('foo', 'bar', 'Zoo', 'baz');  
console.log(block.toString());  
console.log(Block.genesis().toString());
```

```
C:\Users\Dell\Desktop\BlockChain\Course II>npm run index
```

```
> course-ii@1.0.0 index  
> nodemon
```

```
[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 index.js`  
Block -  
  Timestamp: foo  
  Last Hash: bar  
  Hash: Zoo  
  Data: baz  
Block -  
  Timestamp: Genesis time  
  Last Hash: -----  
  Hash: f1r57-h45h  
  Data:  
[nodemon] clean exit - waiting for changes before restart
```

# Blockchain

## Mine Blocks:

```
static mineBlock(lastBlock, data){  
  const timestamp = Date.now();  
  const lastHash = lastBlock.hash;  
  const hash = 'todo-hash';  
  
  return new this(timestamp, lastHash, hash, data);  
}
```

```
const fooBlock = Block.mineBlock(Block.genesis(), 'foo');  
console.log(fooBlock.toString());
```

```
[nodemon] restarting due to changes...  
[nodemon] starting `node index.js`  
Block -  
  Timestamp: 1661751440935  
  Last Hash: f1r57-h45h  
  Hash: todo-hash  
  Data: foo  
[nodemon] clean exit - waiting for changes before restart
```

# Blockchain

## SHA256 Hash Function:

### Block Hashes and SHA-256

- The hash is generated from the timestamp, lastHash, and stored data.
- We'll use an algorithm called SHA-256.
  - Produces a unique 32-byte (256 bit) hash value for unique data inputs.
  - One-way hash.
- Useful for block validation.

```
C:\Users\Dell\Desktop\BlockChain\Course II>npm i crypto-js --save
added 1 package, and audited 34 packages in 848ms
3 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

```
const SHA256 = require('crypto-js/sha256');
```

```
static mineBlock(lastBlock, data){
  const timestamp = Date.now();
  const lastHash = lastBlock.hash;
  const hash = Block.hash(timestamp, lastHash, data);

  return new this(timestamp, lastHash, hash, data);
}
```

```
static hash(timestamp, lastHash, data){
  return SHA256(`${timestamp}${lastHash}${data}`).toString();
}
```

```
[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 index.js`
Block -
  Timestamp: 1661752741755
  Last Hash: f1r57-h45h
  Hash: 0ca2b0c81a
  Data: foo
[nodemon] clean exit - waiting for changes before restart
```

# Blockchain

## Test the Block:

```
C:\Users\Dell\Desktop\BlockChain\Course II>npm i jest --save-dev

added 275 packages, and audited 309 packages in 14s

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

found 0 vulnerabilities

C:\Users\Dell\Desktop\BlockChain\Course II>_
```

```
Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        0.735 s, estimated 1 s
Ran all test suites.
```

Watch Usage: Press w to show more.

```
"scripts": {
  "test": "jest --watchAll",
  "index": "nodemon "
},
```

JS block.test.js > ...

```
1  // const { expect } = require('@babel/types');
2  // const { it } = require('node:test');
3  // const { describe } = require('yargs');
4  const Block = require('./block');
5
6  describe('Block', () => {
7
8      let data, lastBlock, block;
9
10     beforeEach(() => {
11         data = 'bar';
12         lastBlock = Block.genesis();
13         block = Block.mineBlock(lastBlock, data);
14     });
15     it('sets the `data` to match the input', () => {
16         expect(block.data).toEqual(data);
17     });
18
19     it('sets the `lastHash` to match the hash of the last block', () => {
20
21         expect(block.lastHash).toEqual(lastBlock.hash);
22
23     });
24 });
```

# Blockchain