Blockchain Technology

LAB 2-
To create a blockchain and implement replay attacks on blockchain

Outline

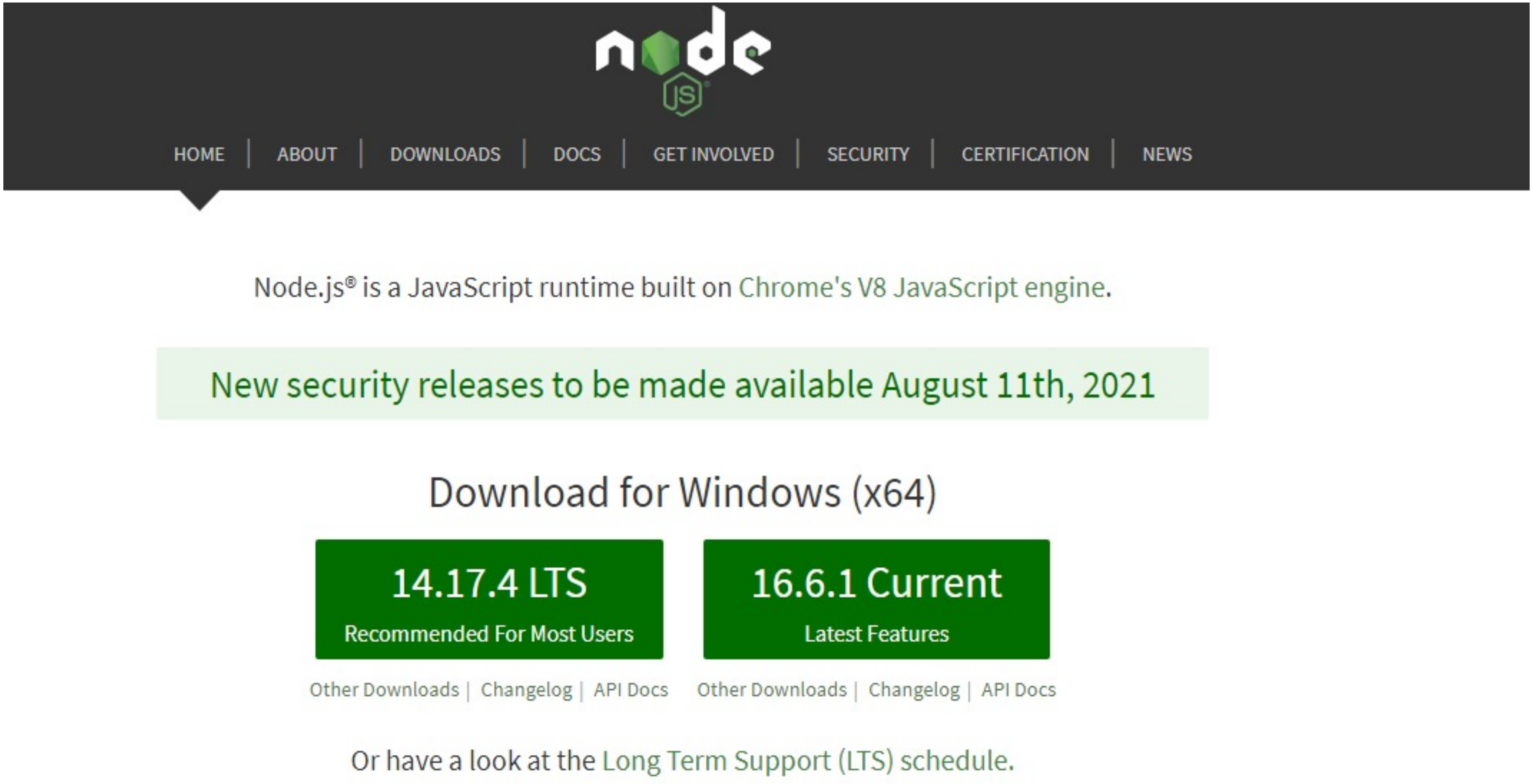
- Prerequisites
- What is Blockchain?
- How To Create A Blockchain
- Testing and Integrity Verification of the Blockchain
- Implement replay attacks on Blockchain
- Conclusion

PREREQUISITES

PREREQUISITES

- To implement this lab, you’ll need to have the following:
- Node.js installed on your machine. You can download it from (https://nodejs.org/en/)
- A code editor, such as Visual Studio Code (https://code.visualstudio.com/download)

(https://nodejs.org/en/)



HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

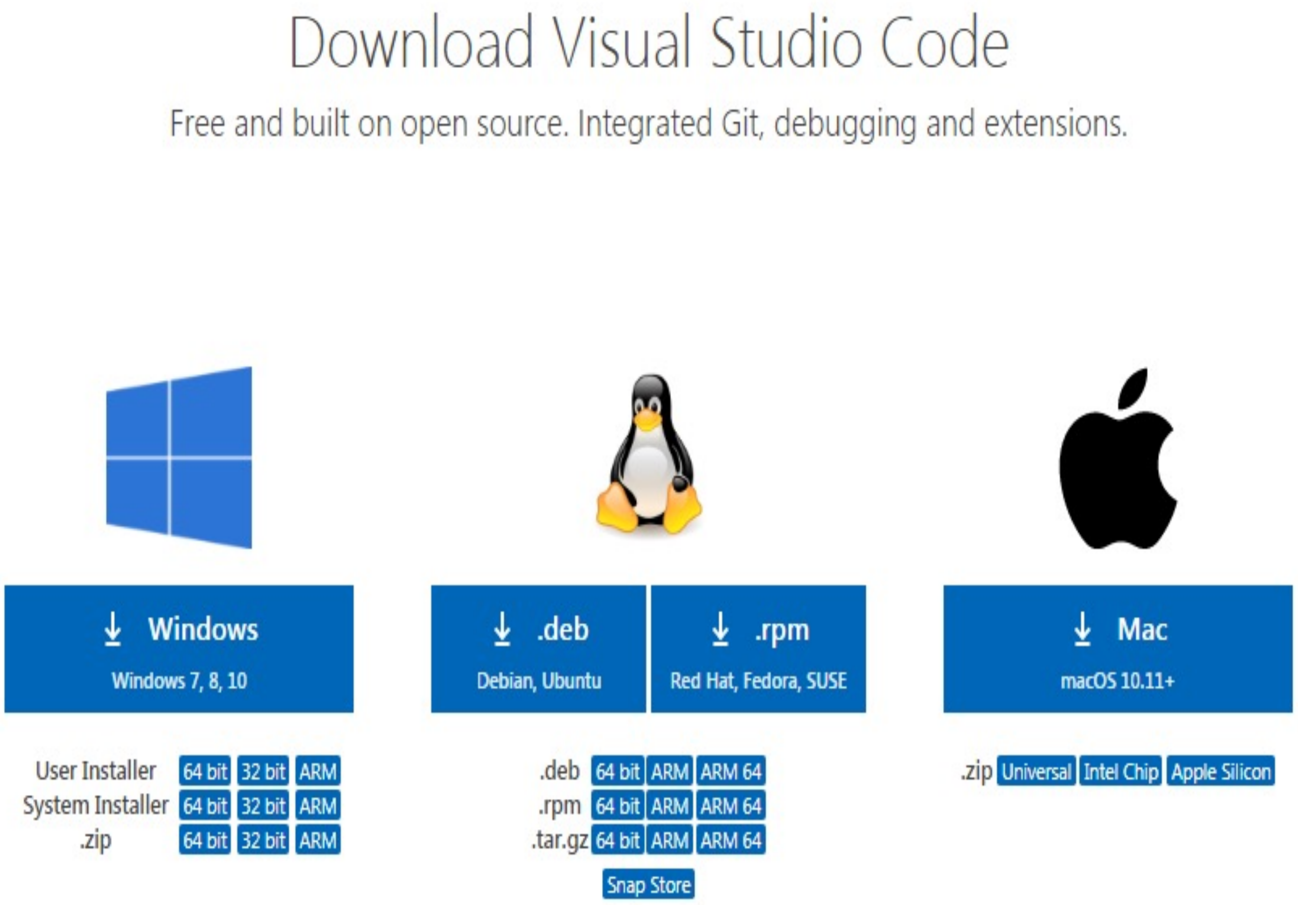
New security releases to be made available August 11th, 2021

Download for Windows (x64)

14.17.4 LTS
Recommended For Most Users
16.6.1 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs

Or have a look at the Long Term Support (LTS) schedule.

(https://code.visualstudio.com/download)



Download Visual Studio Code

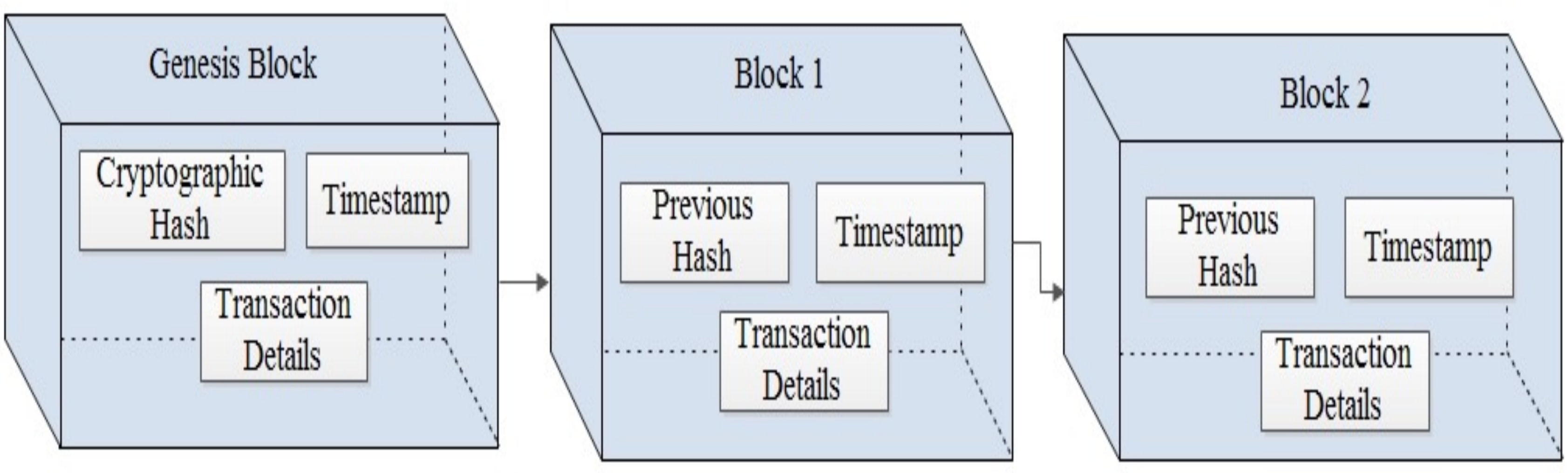
Free and built on open source. Integrated Git, debugging and extensions.

WHAT IS A BLOCKCHAIN?

What is Blockchain?

- Blockchain is the technology that powers digital currencies, such as Bitcoin and Ethereum.
- It is an innovative distributed public ledger technology that maintains a continuously growing list of records, referred to as blocks, which are connected securely using cryptography.

What is Blockchain?



Block Components

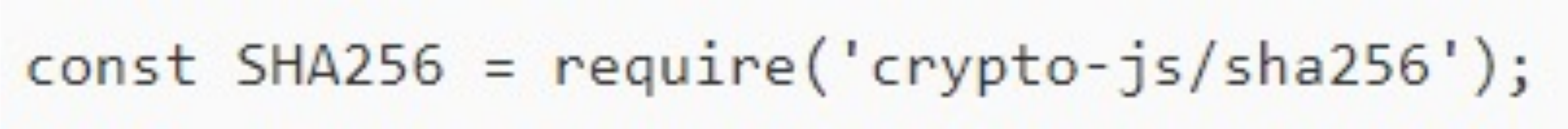
- Index: It’s a unique number that tracks the position of every block in the entire blockchain.
- Timestamp: It keeps a record of the time of occurrence of each completed transaction.
- Data: It provides data about the completed transactions, such as the sender details, recipient’s details, and quantity transacted.
- Preceding Hash: It points to the hash of the preceding block in the blockchain, something important in maintaining the blockchain's integrity.

Summary: How To Create A Block(CryptoBlock class)

- Import the crypto-js JavaScript library and used its crypto-js/sha256 module to calculate the hash of each block.
- To add the crypto-js library to your project, go the terminal and run the following command to install it using npm:
- **npm install --save crypto-js**
- Create the CryptoBlock class
- Add the constructor() method to it just like it’s done in any other JavaScript class.
- Then, to initialize its properties, assign the relevant parameters to the constructor method
- Use the computeHash method to calculate the hash of the block based on its properties, as given in the data above.
- This module returns a number object used the toString() method to convert it into a string.

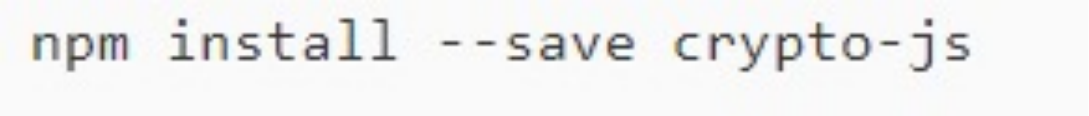
How To Create A Block(CryptoBlock class)

- Import the crypto-js JavaScript library and used its crypto-js/sha256 module to calculate the hash of each block.



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

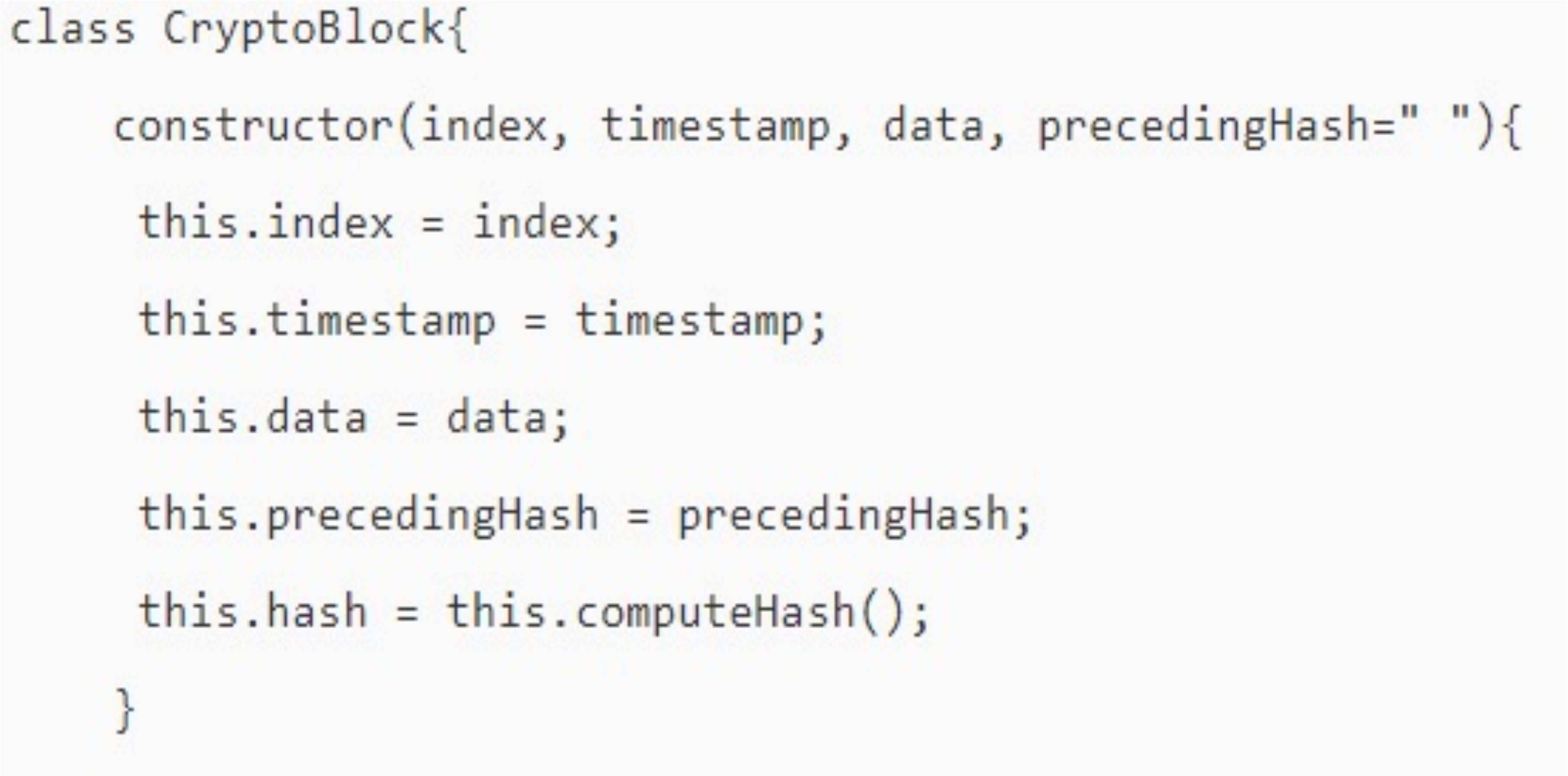
- To add the crypto-js library to your project, go the terminal and run the following command to install it using npm:



```
npm install --save crypto-js
```

How To Create A Block(CryptoBlock class)

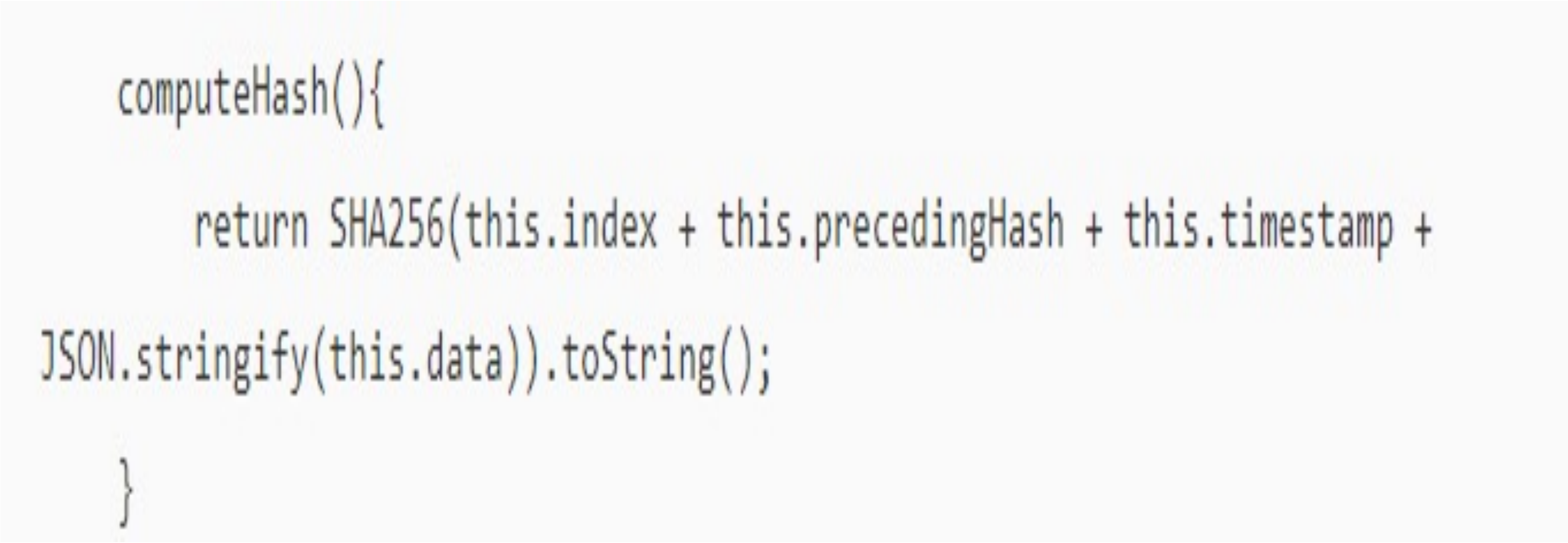
- Create the CryptoBlock class



```
class CryptoBlock{

    constructor(index, timestamp, data, precedingHash=" "){

     this.index = index;

     this.timestamp = timestamp;

     this.data = data;

     this.precedingHash = precedingHash;

     this.hash = this.computeHash();

    }
```

How To Create A Block(CryptoBlock class)

- Use the computeHash method to calculate the hash of the block based on its properties, as given in the data above.



```
    computeHash(){

        return SHA256(this.index + this.precedingHash + this.timestamp +

JSON.stringify(this.data)).toString();

    }
```

- This module returns a number object used the toString() method to convert it into a string.

How To Create A Blockchain?

How To Create A Blockchain?

1. USE OF CONSTRUCTOR METHOD

This method instantiates the blockchain. Inside the constructor, created the blockchain which refers to an array of blocks. Passed it to the startGenesisBlock() method, which creates the initial block in the chain.

2. CREATING THE GENESIS BLOCK

Use the startGenesisBlock() method to create the genesis block. You can create it using the CryptoBlock class and pass the index, timestamp, data, and precedingHash parameters.

How To Create A Blockchain?

3. OBTAINING THE LATEST BLOCK

Getting the latest block in the blockchain assists in ensuring the hash of the current block points to the hash of the previous block thus maintaining the chain’s integrity.

4. ADDING NEW BLOCKS

Use The addNewBlock() method to add a new block to the chain. To accomplish this, we can set the previous hash of the new block to be equal to the hash of the last block in the chain to ensure the chain is tamper-proof.

Testing and Intergity Verification of the Blockchain

Implement Replay Attacks on Blockchain

Conclusion

- That’s how you can build a simple blockchain using Node.js

Thank You