1. First, navigate to the root directory of your project in your terminal. Then, run the following command:

npm init

This will prompt you with a series of questions to fill in the details of your **package.json** file. Here's an example of how you might answer these questions:

After answering these questions, npm will generate a package.json file with the provided details in the root directory of your project.

If you need to add dependencies or scripts later, you can manually edit the package.json file or use npm install <package-name> to install dependencies and npm run <script-name> to run scripts defined in the scripts section of the package.json file.

1. const fs = require('fs');

// Information about Node.js architecture

const nodejsArchitectureInfo = `

Node.js is built on the V8 JavaScript runtime and uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

`;

// Write information to the file

fs.writeFile('nodejs\_architecture.txt', nodejsArchitectureInfo, (err) => {

if (err) {

console.error('Error writing to file:', err);

} else {

console.log('Node.js architecture information has been saved to nodejs\_architecture.txt');

}

});

1. const fs = require('fs');

// Read the content of nodejs\_architecture.txt file

fs.readFile('nodejs\_architecture.txt', 'utf8', (err, data) => {

if (err) {

console.error('Error reading file:', err);

} else {

console.log('Content of nodejs\_architecture.txt:');

console.log(data);

}

});

1. const fs = require('fs');

// Delete the nodejs\_architecture.txt file

fs.unlink('nodejs\_architecture.txt', (err) => {

if (err) {

console.error('Error deleting file:', err);

} else {

console.log('File Deleted Successfully');

}

});

8. const EventEmitter = require('events');

// Create an instance of EventEmitter

const eventEmitter = new EventEmitter();

// Subscribe to the custom "subscribe" event

eventEmitter.on('subscribe', () => {

console.log('User has subscribed');

});

// Simulate triggering the "subscribe" event

eventEmitter.emit('subscribe');

9. const EventEmitter = require('events');

// Create an instance of EventEmitter

const eventEmitter = new EventEmitter();

// Define an event handler

const eventHandler = () => {

console.log('Event handler has been triggered');

};

// Subscribe to the custom "myEvent" event

eventEmitter.on('myEvent', eventHandler);

// Call the event handler by emitting the "myEvent" event

eventEmitter.emit('myEvent');

// Remove the event handler

eventEmitter.removeListener('myEvent', eventHandler);

// Try calling the event handler again

eventEmitter.emit('myEvent');