

Demo: Generating unit tests using GitHub Copilot

You can generate unit test cases for your code using GHCP. It can generate unit tests for different languages including C#, Java, JavaScript, Python etc. It can produce code for various unit test frameworks.

Prerequisites

- VS Code
- GitHub Copilot extension for VS Code
- GitHub Copilot subscription
- NodeJS 18 or later
- Book API project

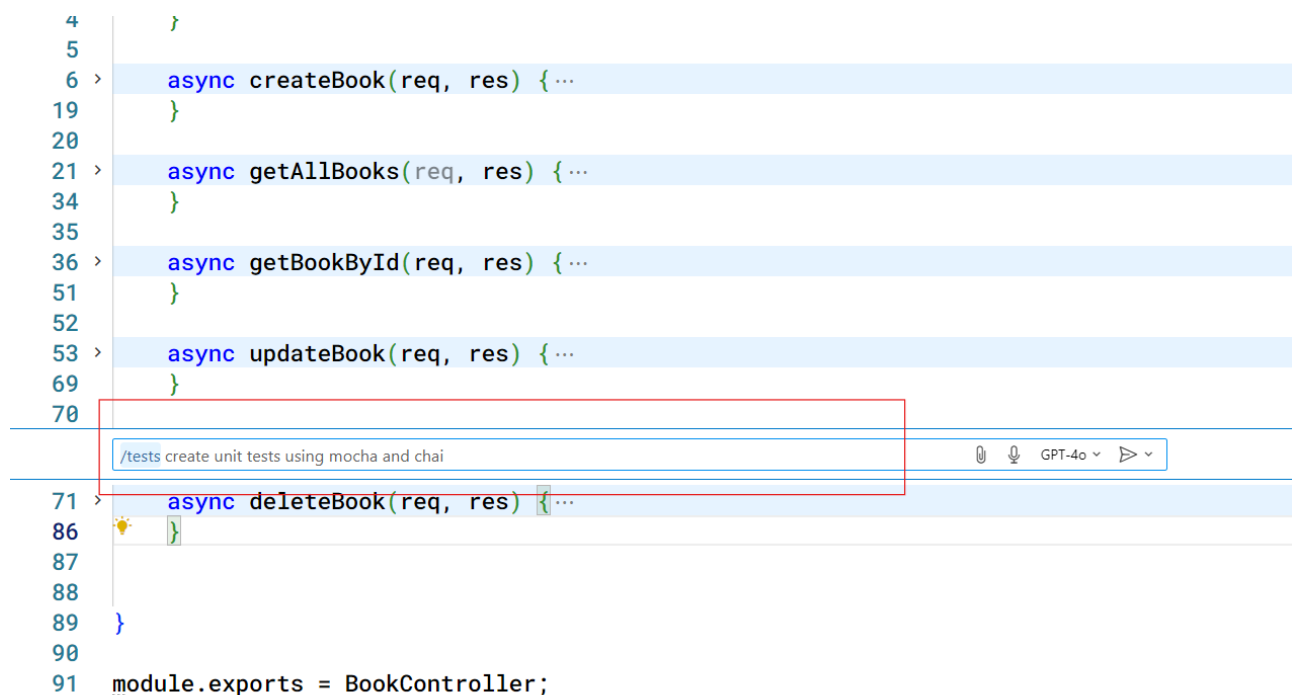
Generating unit tests for NodeJS application

1. Open the `book-api` project in VS Code.
2. Install the testing framework `mocha` and assertion library `chai` in the application

```
npm install -D mocha chai sinon
```

3. Open the file `bookController.js` in the editor.
4. Press `Ctrl+I` to open the Inline chat and use the `/tests` command with the following prompt.

```
/tests Create unit tests using mocha and chai
```



5. This will generate a new file with all test cases. Save the file in `/test/bookController.test.js` file.

```

1  const chai = require('chai');
2  const sinon = require('sinon');
3  const BookController = require('./bookController');
4
5  const expect = chai.expect;
6
7  describe('BookController', () => {
8    (property) Sinon.SinonSandbox.stub: sinon.SinonStubStatic
9    <any[], any>() => sinon.SinonStub<any[], any> (+2 overloads)
10   Creates an anonymous stub function
11
12     deleteBook: sinon.stub()
13   };
14   req = {
15     params: {
16       id: '1'
17     }
18   };
19   res = {
20     status: sinon.stub().returnsThis(),
21     json: sinon.stub(),
22     send: sinon.stub()
23   };
24   bookController = new BookController(bookModelMock);
25 });
26

```

Ln 59, Col 4 Spaces: 4 UTF-8 CRLF JavaScript Go Live default

6. Open the `package.json` file and add a script to run the tests

```



"scripts": {
  "start": "node src/app.js",
  "test": "mocha ./test/*.test.js"
},

```

7. Run the test cases

```
npm test
```

If required update the paths in the import statements

book-api > test >  bookController.test.js >  BookController

```
1  const chai = require('chai');
2  const sinon = require('sinon');
3  const BookController = require('../src/controllers/bookController');
4
5  const expect = chai.expect;
6
7  describe('BookController', () => {
8    let bookModelMock, req, res, bookController;
9
10    beforeEach(() => {
11      bookModelMock = {
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS


AZURE

COMMENTS

POLYGLOT NOTEBOOK

SQL HISTORY

TASK MONITOR

 pwsh - book-api +

• ERROR sonusathyadas@Sonu-XP360 ~ > Desktop > Upcoming > Demos > book-api  test

```
> express-rest-api@1.0.0 test
> mocha ./test/*.test.js
```

```
BookController
  deleteBook
    ✓ should delete a book and return 204 status
    ✓ should return 404 if book not found
    ✓ should return 500 if there is an error deleting the book
```

3 passing (19ms)