

Demo: Using Smart actions to Explain and Fix the code

Smart actions are seamlessly integrated in your VS Code flow (for example, in Quick Fix and context menus) and they do not require you to write any prompt at all. The most powerful smart action is `/fix`.

Prerequisites

- VS Code
- GitHub Copilot Extension for VS Code
- GitHub Copilot subscription
- NodeJS 18 or later
- Book API application

Explain the code using `/explain` smart action

1. Open the `book-api` project in VS Code.
2. Open the `bookController.js` file in the editor.
3. Right click inside the file content and choose `Copilot > Explain` smart action.
4. GitHub Copilot will open the Chat window and shows the explanation of the code in the chat window.

sonusathyadas
@workspace /explain
JS bookController.js:1-89

Workspace
> Used 1 reference

The provided code defines a **(x) BookController** class in JavaScript, which is designed to handle various operations related to books in a web application. This class interacts with a **(x) bookModel** to perform CRUD (Create, Read, Update, Delete) operations on book data. Each method in the class corresponds to a specific operation and handles HTTP requests and responses.

The **(x) constructor** method initializes the **(x) BookController** with a **(x) bookModel** instance, which is used to interact with the data layer. The **(x) createBook** method handles the creation of a new book. It extracts book data from the request body and uses the **(x) bookModel** to create a new book. If an error occurs during this process, it sends a 500 status code with an error message; otherwise, it responds with a 201 status code and the newly created book.

The **(x) getAllBooks** method retrieves all books from the database. It calls the **(x) bookModel**'s **(x) getAllBooks** method and handles the response. If an error occurs, it logs

Can you explain error handling?

Ask Copilot
JS bookController.js:1-87 Current file
GPT-4o

Fix the error in Code using Copilot

1. Open the **book-api** project in VS Code.

2. Open the `bookController.js` file in the editor. Create some errors in the code by removing brackets, using incorrect variable names etc.
3. Right click on the file content and select `Copilot > Fix` from the Context menu.
4. It will try to detect the errors in the code and suggest the changes.

```
book-api > src > controllers > JS bookController.js > BookController > updateBook

1 class BookController {
52

  /fix
  /fix 'id' is declared but its value is never read., Could not find name 'id'. Did you mean 'id?', 'updatedBook' is declared but its value is never read., Could not find name 'updatedbook'. Did you mean 'updatedBook'?
  To fix the issues, you need to correct the destructuring of [id] to use [id] instead of [id], and ensure consistent naming for [updatedBook].

Ask Copilot
Accept

53 async updateBook(req, res) {
54   try {
55     const { id } = req.params;
56     const updatedData = req.body;
57     this.bookModel.updateBook(id, updatedData, (err, updatedBook) => {
58       if (err) {
59         res.status(500).json({ message: 'Error updating book', error: err });
60       } else if (updatedBook) {
61         res.status(200).json(updatedBook);
62       } else {
63         res.status(404).json({ message: 'Book not found' });
64       }
65     });
66   } catch (error) {
67     res.status(500).json({ message: 'Error updating book', error: error });
68   }
69 }
```

5. You can review the suggestions and accept the changes if required.