## Demo: Using Smart actions to Explain and Fix the code

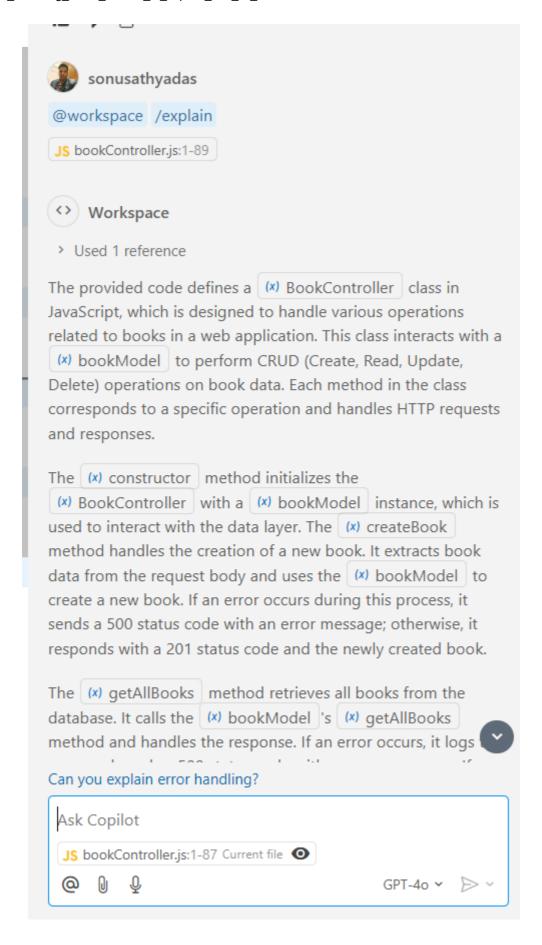
Smart actions are seamlessly integrated in your VS Code flow (for example, in Quick Fix and context menus) and they do not require you to write any prompt at all. The most powerful smart action is /fix.

## 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.



## 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 erros 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.

```
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'?, Could not find
         name 'updatedbook'. Did you mean 'updatedBook'?
      8 To fix the issues, you need to correct the destructuring of (w) id , use (w) id instead of (w) ld , and ensure consistent naming for (w) updatedBook
     Ask Copilot
     Accept Ø V
           async updateBook(req, res) {
53
               try {
const { id } } = req.params;
54
55
                    const updatedData = req.body;
56
                    this.bookModel.updateBook(id, updatedData, (err, updatedBook) => {
58
                               res.status(500).json({ message: 'Error updating book', error: err });
59
                         } else if (updatedBook) {
60
                             res.status(200).json(updatedBook);
61
62
                         -} else {
                            res.status(404).json({ message: 'Book not found' });
63
64
                         -}
                  ···});
66
                   res.status(500).json({ message: 'Error updating book', error });
67
68
69
```

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