

Demo: Referencing codebase in Requests

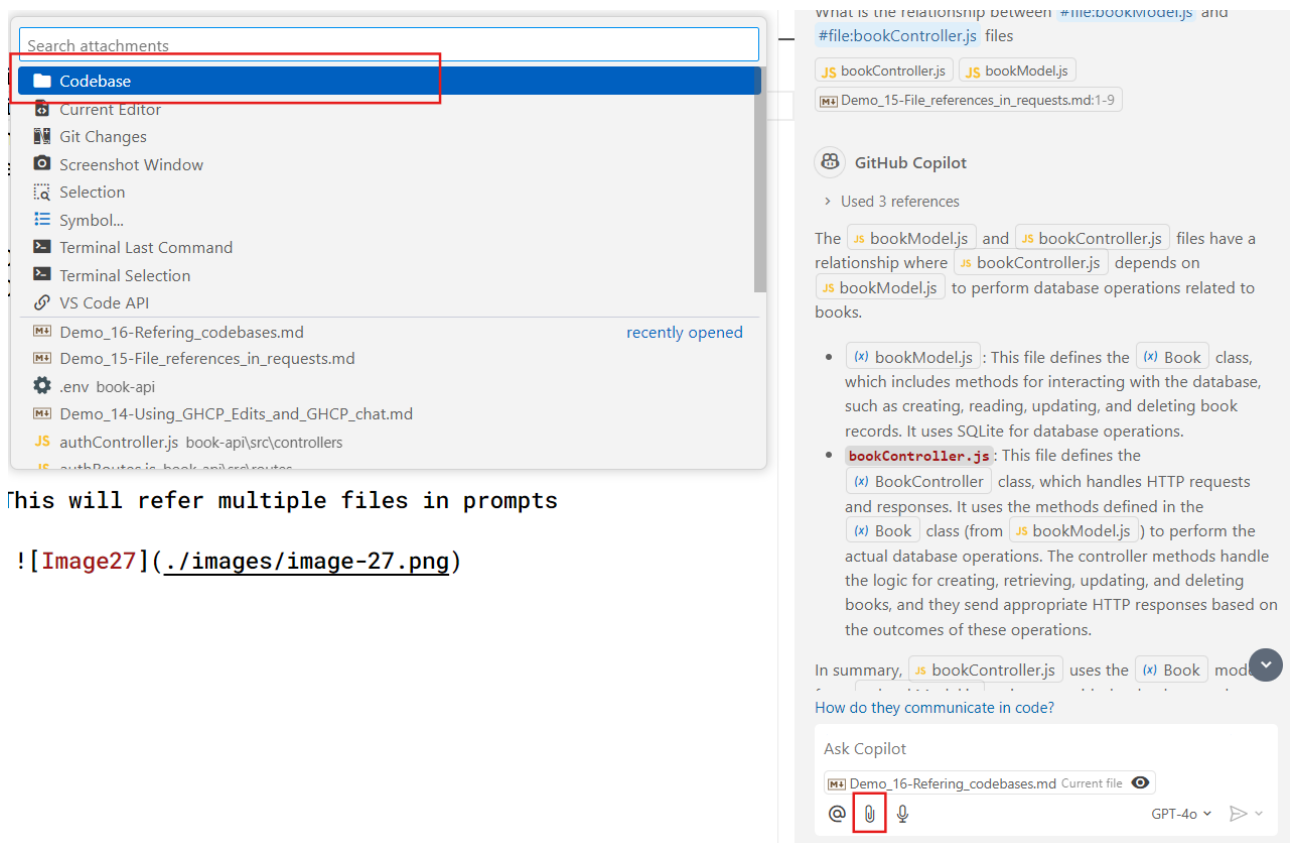
You can refer the entire code base (workspace) while making the requests in Chat or Copilot Edits. To refer the entire codebase in Chat or Edits use the **Attachment pin** icon in the Copilot Chat or Copilot Edits.

Prerequisites

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

Using codebase in Copilot requests

1. Open the **book-api** project in VS Code.
2. Open the Copilot Chat and click on the attachment pin icon and select **codebase** from the popup list.



The screenshot shows the VS Code interface. On the left, the 'Search attachments' dropdown menu is open, with 'Codebase' selected. The right panel shows the Copilot chat interface. The prompt is: 'what is the relationship between #file:bookModel.js and #file:bookController.js files'. The response is: 'The JS bookModel.js and JS bookController.js files have a relationship where JS bookController.js depends on JS bookModel.js to perform database operations related to books. ... In summary, JS bookController.js uses the (X) Book model. How do they communicate in code?'. The 'Attachment pin' icon is highlighted in the bottom right of the chat interface.



This will refer multiple files in prompts





! [Image27] (./images/image-27.png)


3. Write the below prompt in the chat and press Enter.

Check the project files are following best practices and naming conventions


4. It will refer all files in the code base and check for the best practices and naming conventions






**sonusathyadas**

Check the project files are following best practices and naming conventions

 Codebase

**GitHub Copilot**

> Used 22 references




✓ Searched codebase for **best practices naming conventions**

The **JS bookModel.js** and **JS bookController.js** files have a relationship where **JS bookController.js** depends on **JS bookModel.js** to perform database operations related to books.

- (x) bookModel.js**: This file defines the **(x) Book** class, which includes methods for interacting with the database, such as creating, reading, updating, and deleting book records. It uses SQLite for database operations.
- (x) bookController.js**: This file defines the **(x) BookController** class, which handles HTTP requests and responses. It uses the methods defined in the **(x) Book** class (from **JS bookModel.js**) to perform the actual database operations. The controller methods handle the logic for creating, retrieving, updating, and deleting books, and they send appropriate HTTP responses based on the outcomes of these operations.

In summary, **JS bookController.js** uses the **(x) Book** model from **JS bookModel.js** to interact with the database and

Ask Copilot



GPT-4o 