

# Demo: Generating code completions using GitHub Copilot

---

GitHub Copilot can suggest the code when you type on the editor. When you create a class, function or declare a variable or constant, Copilot can suggest the autocompletions.

## Prerequisites

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

## Generating code completions and autosuggestions

1. Open the `book-api` Nodejs project in VS Code.
2. Open the `bookModel.js` file and start typing at the end of the class. Add a new method for retrieving the list of books using pagination. Start typing the function name and GitHub Copilot will suggest the autocompletions.

```
static getBooksByPage(page, limit, callback) {
```

3. Press the `Tab` key to accept the suggestions.

```
static getBooksByPage(page, limit, callback) {  
  const offset = (page - 1) * limit;  
  db.all('SELECT * FROM books LIMIT ? OFFSET ?', [limit, offset], (err, rows) => {  
    callback(err, rows);  
  });  
}
```

4. Open the `bookController.js` file from the `controllers` folder. Add a new method by typing the name of the async method.

```

async getBooksByPage(req, res) {
  try {
    const { page, limit } = req.query;
    this.bookModel.getBooksByPage(page, limit, (err, books) => {
      if (err) {
        res.status(500).json({ message: 'Error retrieving books', error: err });
      } else {
        res.status(200).json(books);
      }
    });
  } catch (error) {
    res.status(500).json({ message: 'Error retrieving books', error });
  }
}

```

5. Press **Tab** to accept the suggestions.

```

async getBooksByPage(req, res) {
  try {
    const { page, size } = req.params;
    const pageNumber = parseInt(page, 10) || 1;
    const pageSize = parseInt(size, 10) || 10;
    this.bookModel.getBooksByPage(pageNumber, pageSize, (err, books) => {
      if (err) {
        res.status(500).json({ message: 'Error retrieving books', error: err });
      } else {
        res.status(200).json(books);
      }
    });
  } catch (error) {
    res.status(500).json({ message: 'Error retrieving books', error });
  }
}

```

6. Open the **bookRoutes.js** file from **routes** folder and add a new route by typing the route configuration

```

router.post('/', bookController.createBook.bind(bookController));
router.get('/', bookController.getAllBooks.bind(bookController));
router.get('/:id', bookController.getBookById.bind(bookController));
router.put('/:id', bookController.updateBook.bind(bookController));
router.delete('/:id', bookController.deleteBook.bind(bookController));
router.get('/:page/:size', bookController.getBooksByPage.bind(bookController));

```

7. Complete the suggestion by pressing **Tab**.

```

router.post('/', bookController.createBook.bind(bookController));
router.get('/', bookController.getAllBooks.bind(bookController));
router.get('/:id', bookController.getBookById.bind(bookController));
router.put('/:id', bookController.updateBook.bind(bookController));
router.delete('/:id', bookController.deleteBook.bind(bookController));
router.get('/:page/:size', bookController.getBooksByPage.bind(bookController));

```

8. Run the application by running the following command.

```
npm start
```

