# LAB 5
# ASP.NET CORE API

## ❖ CONTENT

- Create API with ASP.NET Core
- Consume API with VueJS

## ❖ INSTRUCTION

1. Create new VueJS & ASP.NET Core full-stack web application

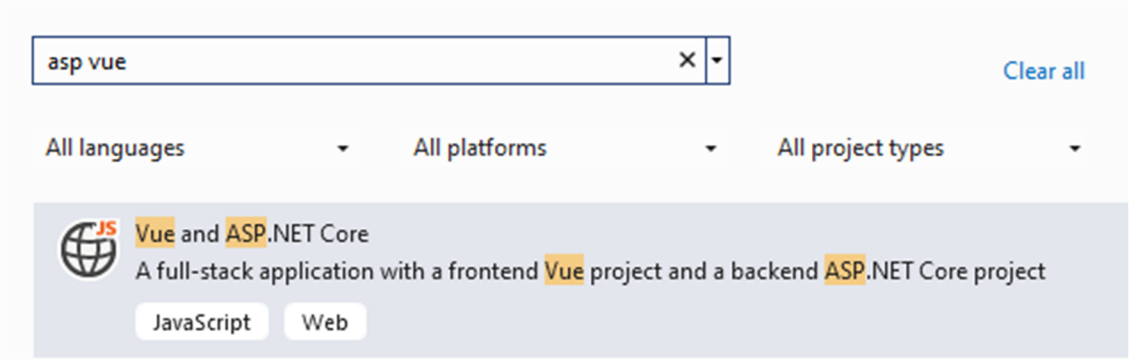


*Figure 1 - Create new project (1)*

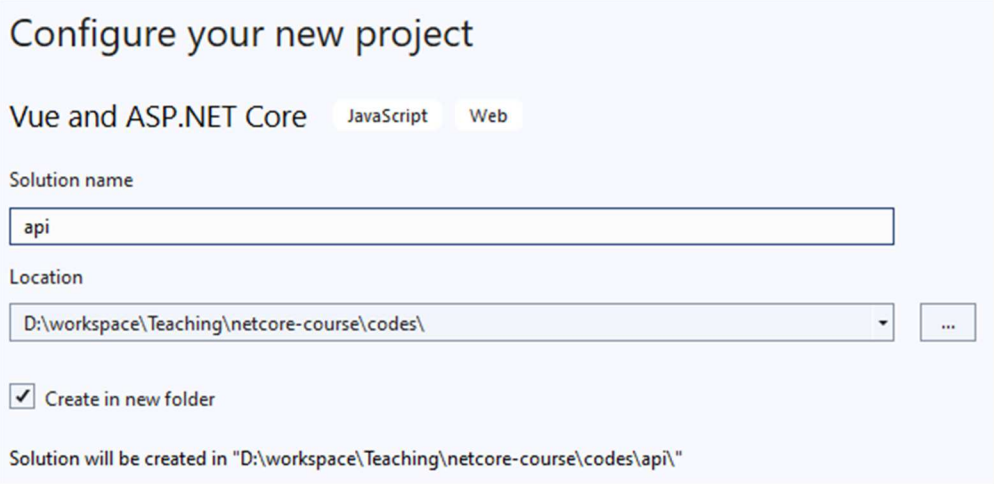


*Figure 2 - Create new project (2)*

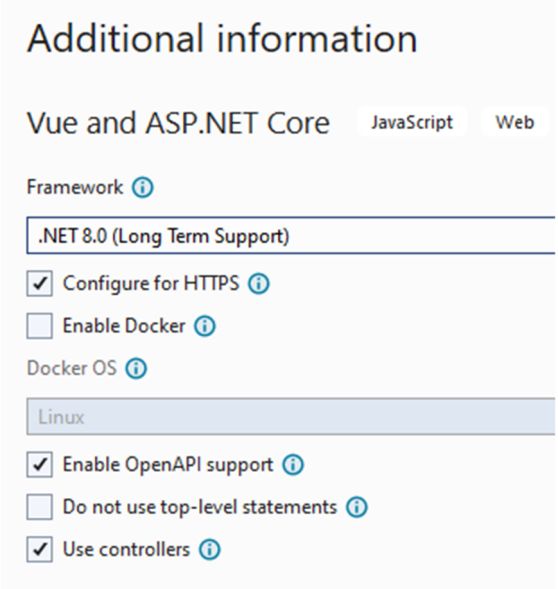


*Figure 3 - Create new project (3)*

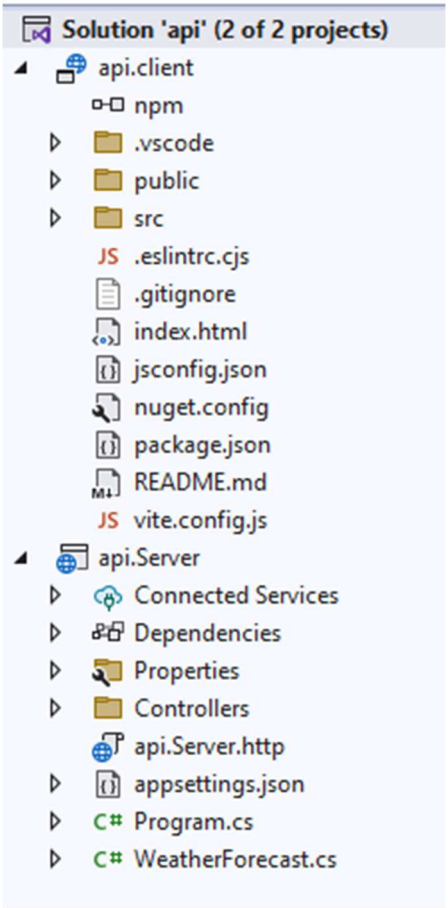


***Figure 4*** *– Web application structure (backend:* ***api.Server****, frontend:* ***api.client****)*

## 2. Create new model as table

```
public class Book
{
    public int BookId { get; set; }
    public required string BookTitle { get; set; } //required: compulsory field
    public double BookPrice { get; set; }
    public int BookQuantity { get; set; }
    public string? BookCover { get; set; } //? : nullable field
}
```

*Figure 5 – api.Server/Models/Book.cs*

## 3. Generate API Controller by adding new scaffolded item

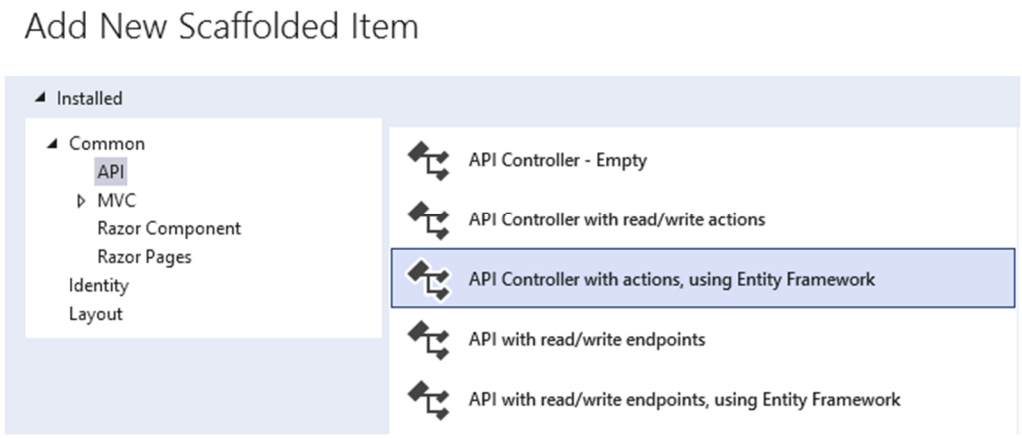


*Figure 6 - Generate API Controller with Scaffolding*

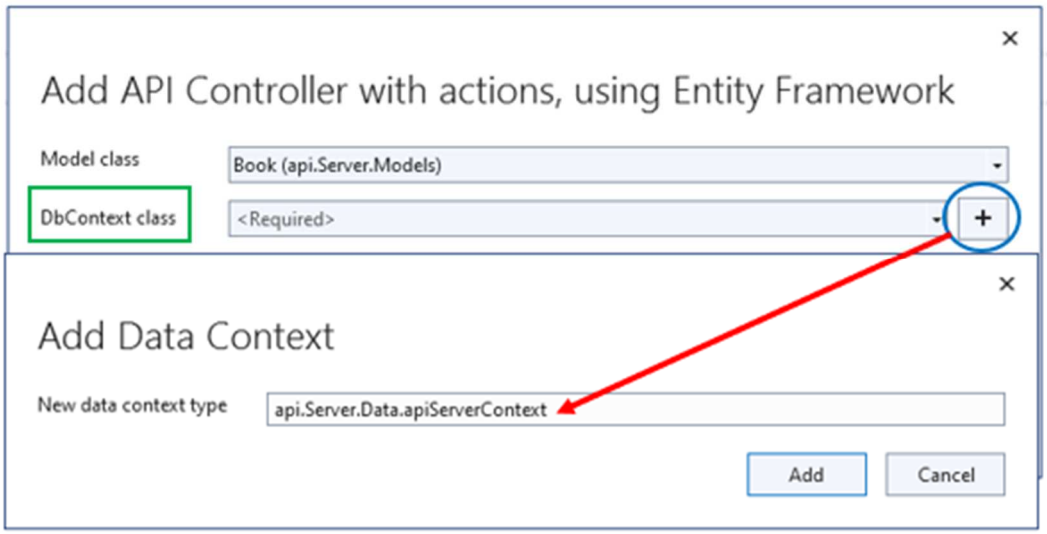


*Figure 7 - Add DBContext class (1)*

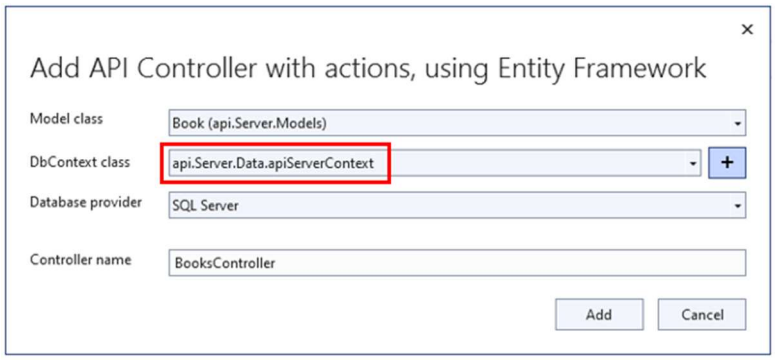


*Figure 8 - Add DBContext class (2)*

## 4. Seed book data when running project

```
public apiServerContext(DbContextOptions<apiServerContext> options)
    : base(options)
{
}

public DbSet<Book> Book { get; set; }

protected override void OnModelCreating(ModelBuilder builder)
{
    base.OnModelCreating(builder);
    SeedBook(builder);
}
```

*Figure 9 – **api.Server/Data/apiServerContext.cs** (1)*

```
private void SeedBook(ModelBuilder builder)
{
    builder.Entity<Book>().HasData(
        new Book
        {
            BookId = 1,
            BookTitle = "Clean Code: A Handbook of Agile Software Craftsmanship",
            BookPrice = 30,
            BookQuantity = 15,
            BookCover = "https://m.media-amazon.com/images/I/41jEbK-jG+L._SX258_BO1,204,203,200_.jpg"
        },
        new Book
        {
            BookId = 2,
            BookTitle = "The Pragmatic Programmer: Your Journey To Mastery",
            BookPrice = 35,
            BookQuantity = 20,
            BookCover = "https://m.media-amazon.com/images/I/41as+WafrFL._SX218_BO1,204,203,200_QL40_FMwebp_.jpg"
        },
        new Book
        {
            BookId = 3,
            BookTitle = "You Don't Know JS Yet: Scope & Closures",
            BookPrice = 20,
            BookQuantity = 10,
            BookCover = "https://m.media-amazon.com/images/I/81kqrwS1nNL.jpg"
        }
    );
}
```

*Figure 10 – **api.Server/Data/apiServerContext.cs** (2)*

```
PM> Add-Migration First
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> Update-Database
Build started...
Build succeeded.
```

*Figure 11 - Use **PMC** to add migration & update database*

5. Config backend API endpoint

```
server: {
    proxy: {
        '^/api': {
            target,
            secure: false
        }
    },
    port: 5173,
    https: {
        key: fs.readFileSync(keyFilePath),
        cert: fs.readFileSync(certFilePath),
    }
}
```

*Figure 12 – **api.client/vite.config.js***

6. Import a CSS framework to decorate web UI *(optional)*

```
<title>ASP.NET Core - VueJS Web App</title>
<link rel="stylesheet" type="text/css" href="semantic/dist/semantic.min.css">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
      rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
      crossorigin="anonymous">
```

*Figure 13 – **api.client/ index.html***

7. Install axios library to work with API through terminal: **npm install axios**

```
"dependencies": {
  "axios": "^1.7.8",
  "vue": "^3.5.12"
},
```

*Figure 14 - **api.client/package.json***

8. Add methods & views to retrieve (GET) & delete (DELETE) books

```
<script>
  import axios from 'axios';

  var api = "/api/books"

  export default {
    data() {
      return {
        books: []
      }
    },
    methods: {
      async fetchBooks() {
        const response = await axios.get(api);
        this.books = response.data;
        console.log(this.books);
      },
      async deleteBook(id) {
        if (confirm('Are you sure you want to delete this book?')) {
          await axios.delete(`${api}/${id}`);
          this.fetchBooks();
        }
      }
    },
    created() {
      this.fetchBooks();
    }
  }
</script>
```

*Figure 15 - **api.client/src/App.vue (1)***

```
<template>
  <div class="container text-center mt-5">
    <table class="table table-hover table-primary">
      <thead class="thead-dark">
        <tr>
          <th colspan="6" class="text-danger text-uppercase">
            <h3>Book List</h3>
          </th>
        </tr>
        <tr>
          <th class="text text-success">Id</th>
          <th class="text text-success">Title</th>
          <th class="text text-success">Price</th>
          <th class="text text-success">Quantity</th>
          <th class="text text-success">Cover</th>
          <th class="text text-success">Menu</th>
        </tr>
      </thead>
```

*Figure 16 - **api.client/src/App.vue (2)***

```
      <tbody>
        <tr v-for="book in books" :key="book.bookId">
          <th>{{ book.bookId }}</th>
          <td>{{ book.bookTitle }}</td>
          <td>${{ book.bookPrice }}</td>
          <td>{{ book.bookQuantity }}</td>
          <td>
            <img :src="book.bookCover" width="150" height="200" />
          </td>
          <td>
            <button @click="deleteBook(book.bookId)"
                    class="btn btn-danger">
              Delete
            </button>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
</template>
```

*Figure 17 - **api.client/src/App.vue (3)***

## 9. Check for final results

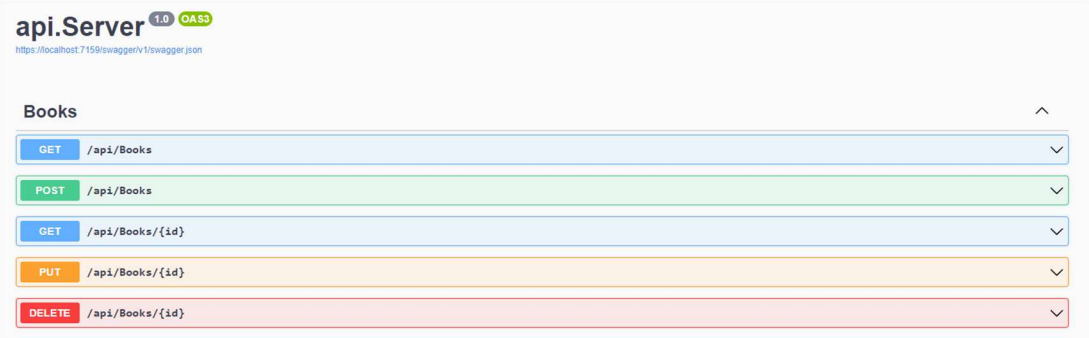


*Figure 18 - Test API with Swagger*

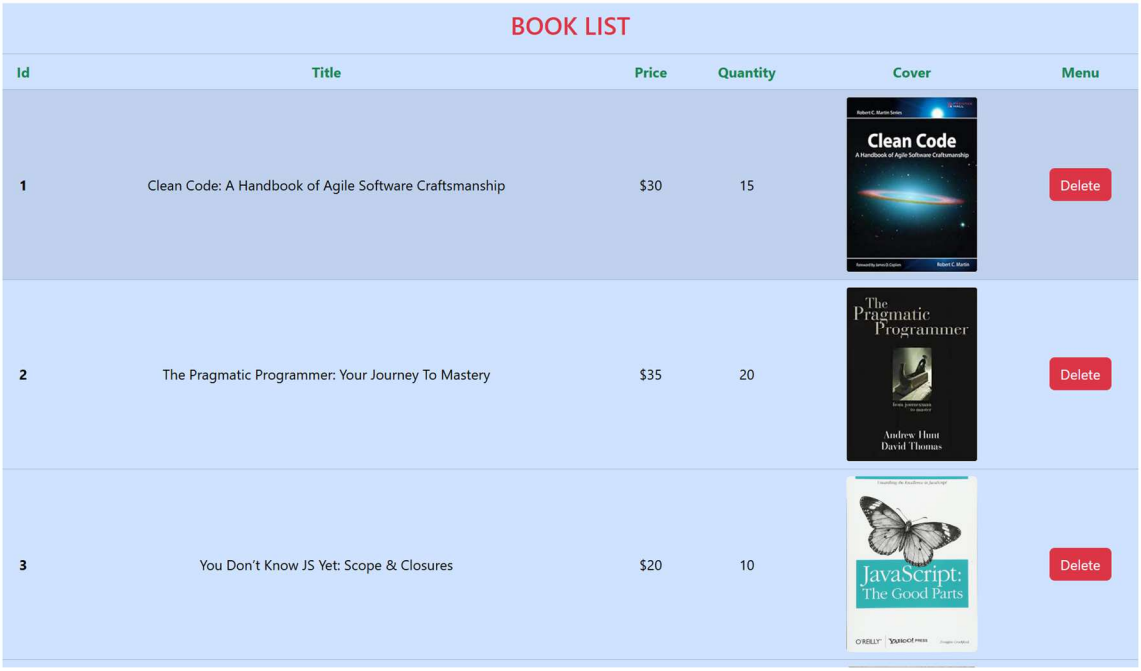


*Figure 19 - Consume API in Vue*