**Supporting Documentation for the Recipe API**

1. **Language Used C#, .Net Core 8.0**

**C#,** because of its high performance and low overhead, making it suitable for high-traffic applications.

C#, has also become a cross platform language with the evolution of Net Core and Asp.Net, allowing developers to build and run applications on Windows, Linux, and macOS.

C# and ASP.NET Core are well-suited for building microservices, with the ability to deploy in containerized environments using Docker and Kubernetes.

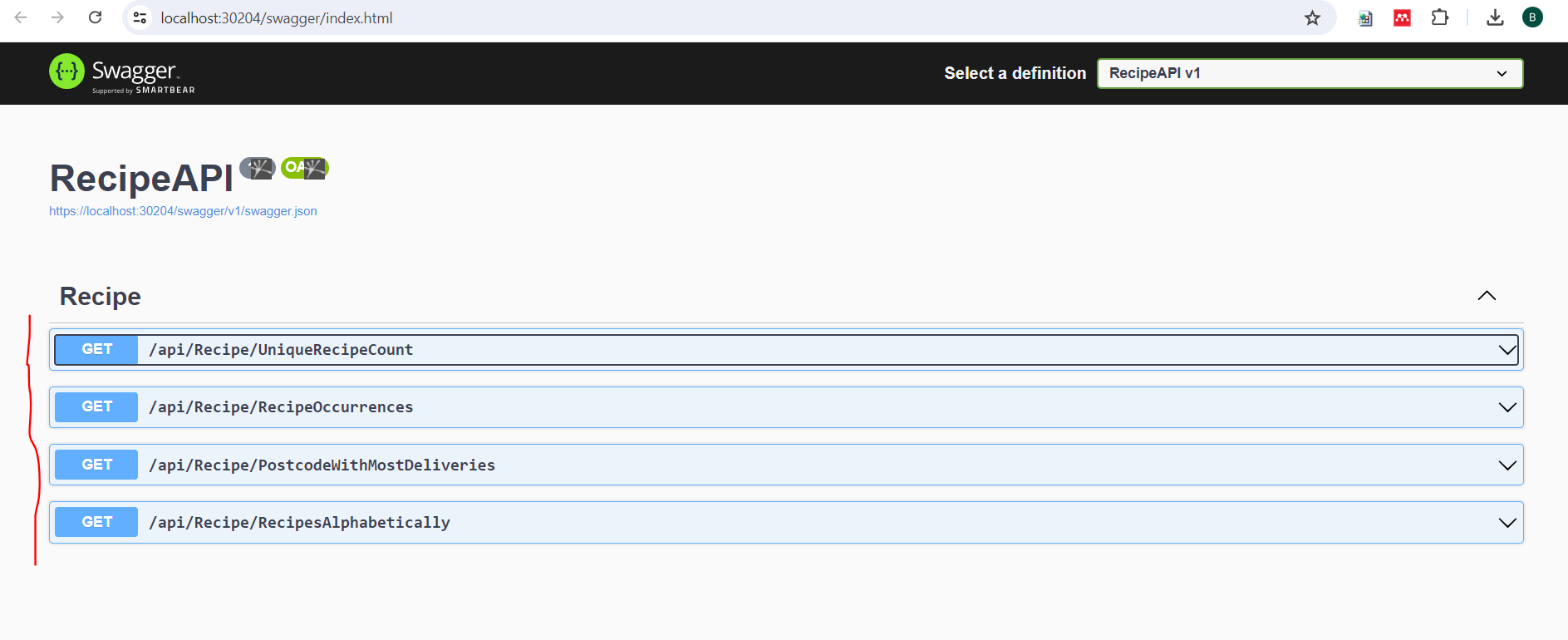
C# and .NET have a large and active developer community, which means plenty of tutorials, forums, and resources are available for troubleshooting and learning.

**Assumptions Made**

* **The data.json file is being read from the project folder.**
* **The only end points provided are the ones that were specified**

1. **Tests**

**List of end points before packaging the API with Docker**

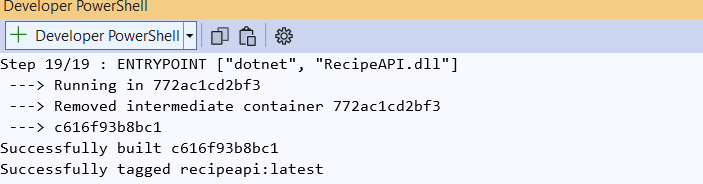


**Package the API Using Docker Container and Create its Image**

**Run the command below in the root dir of the project to create the Image for the API via the VS terminal**

docker build -t recipeapi:latest -f RecipeAPI/Dockerfile .

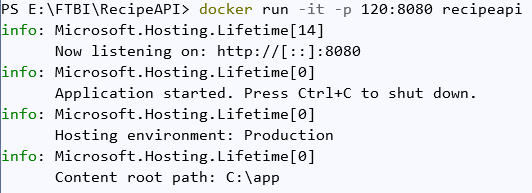
**Result**



**Run the command below to run the created API Image using the VS terminal**

docker run -it -p 120:8080 recipeapi

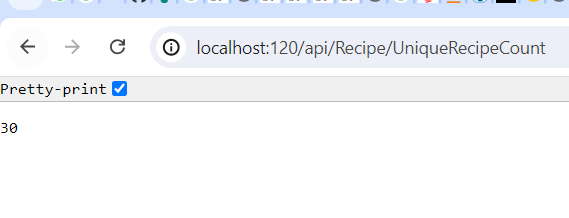
**Result**



**Test the List of end points after packaging the API with Docker via the browser or postman #I used the browser**

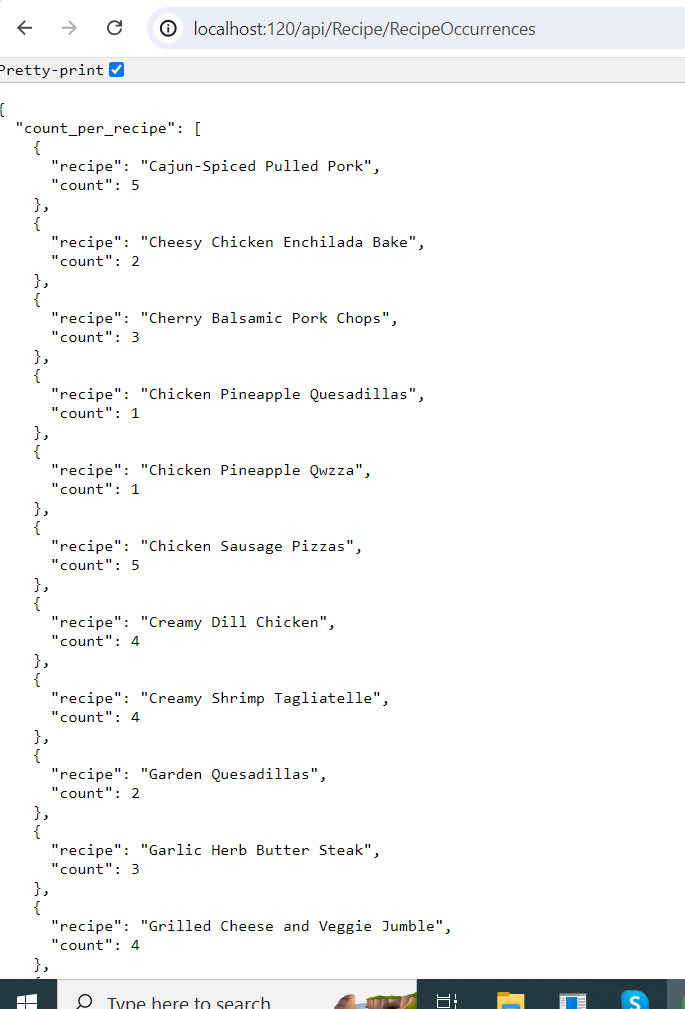
1. [**http://localhost:120/api/Recipe/UniqueRecipeCount**](http://localhost:120/api/Recipe/UniqueRecipeCount)

**Result**



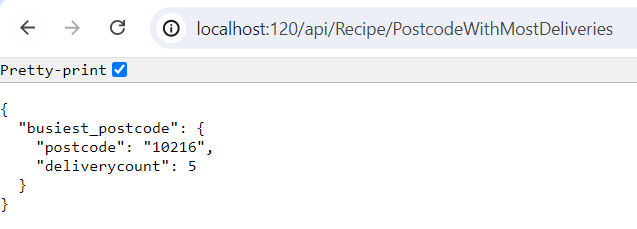
1. http://localhost:120/api/Recipe/RecipeOccurrences

**Result**



1. <http://localhost:120/api/Recipe/PostcodeWithMostDeliveries>

**Result**



1. <http://localhost:120/api/Recipe/RecipesAlphabetically>

**Result**

