# Web Programming 3 (420-5W6) Milestone 4 [Individual] (15%)

Objectives: **Document the creation of multiple microservices with ASP.NET Core.**

**Integrate Open API specifications and Swagger documenting tools to ASP.NET Core Microservices.**

Due Date: **December 19th**

For each chapter, submit on Lea 1 Zip folder containing all files and the solution file (.SLN).

* **IMPORTANT:** In Azure, delete the Fabric and all dependent services, in order to not get charged more. You need the credit in the following semester.
* **Attach a screenshot from Azure proving that you do not have any more services running (service fabric, key vault, resources).**

Late submissions are worth 0%.

## Task

For each microservice, use Swagger documenting tool to create API level documentation.

## Bonus 10% of this milestone

Deploy the application with Swagger documentation for all microservices on Azure and provide a how-to guide including the swagger URL. Provide proof of Swagger deployment with screenshots clearly showing the URL.

## Submissions:

You must submit your final source code in 1 ZIP file on Lea.

## Attendance

## Learning Microservices in the classroom is an active learning activity. There are lots of details and intricacies only explained in class.  Active learning of competencies requires hands on learning with interactive classroom work, which requires no more than 20% absences. Attendance is compulsory for this active learning activity. Milestone 3 (15%) grading has 20% worth of the points related to attendance. **For the rest of the semester, unexcused absences of more than 20% of any of these activities will result in a grade of 0 for the assessment related to that activity. If you have less than 20% of absences, it will be calculated based on the worth of 20% for each assessment.**

## Demo

There will be no demo.

Read next page 🡺

## Adding Swagger documentation tool to ASP.NET Core 3.1 Microservices Project

Example of Swagger Documentation:

<https://petstore.swagger.io/>

Read on how to incorporate Swagger/Open API with Swashbuckle:

<https://learn.microsoft.com/en-us/aspnet/core/tutorials/web-api-help-pages-using-swagger?view=aspnetcore-3.1>

Get started with Swashbuckle and ASP.NET Core

<https://learn.microsoft.com/en-us/aspnet/core/tutorials/getting-started-with-swashbuckle?view=aspnetcore-3.1&tabs=visual-studio>

This is an example for Ecommerce.Api.Products. Apply this example for all other projects.

1. In the Ecommerce.Api.Products project, add the NuGet package: Swashbuckle.AspNetCore   
   This adds all the necessary packages needed by Swagger:   
   Text

   Description automatically generated

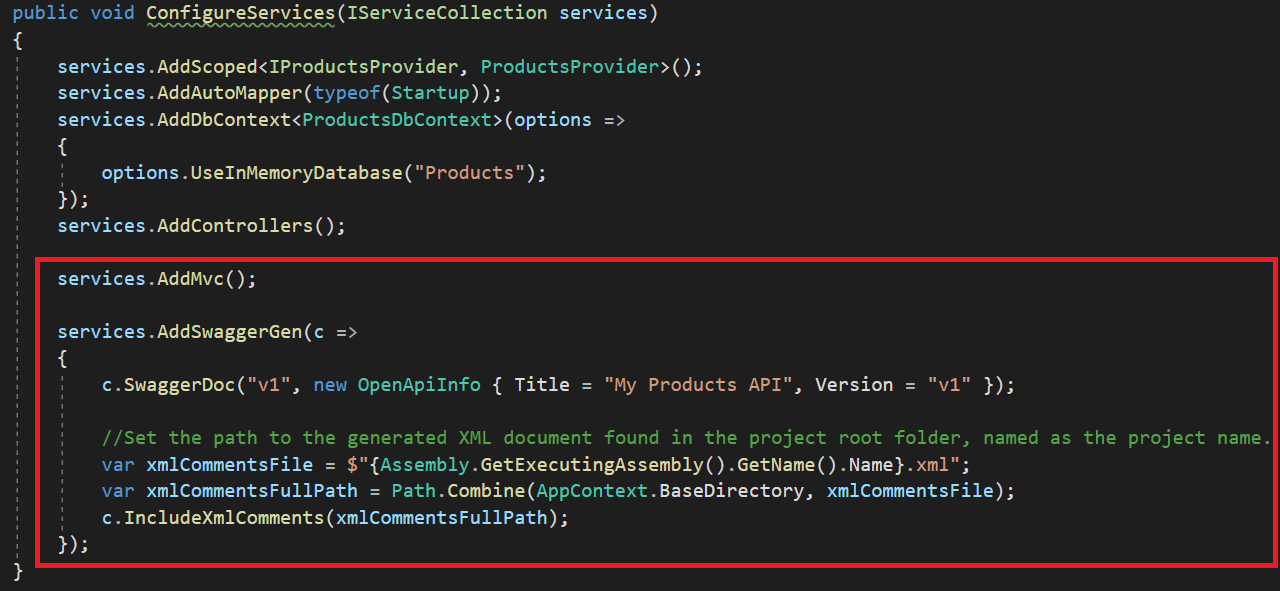
Graphical user interface, application

Description automatically generated

1. Read the official Swashbuckle.AspNetCore Getting Started page:

<https://github.com/domaindrivendev/Swashbuckle.AspNetCore>

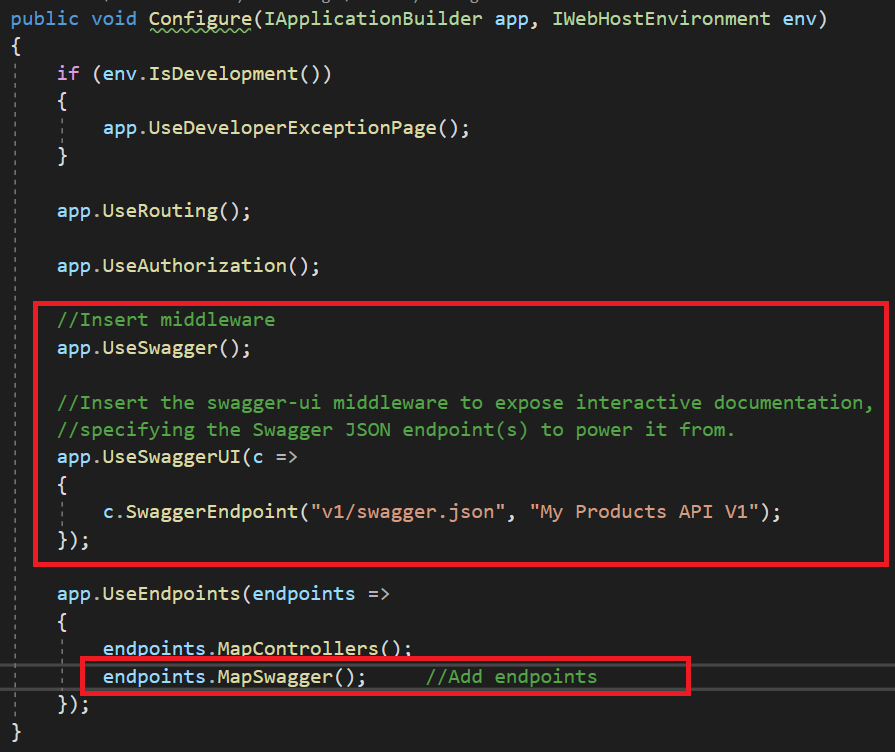
1. The following steps are based on the above link:
2. In the *ConfigureServices* method of *Startup.cs*, register the Swagger generator, defining one or more Swagger documents. Add these lines and any extra libraries



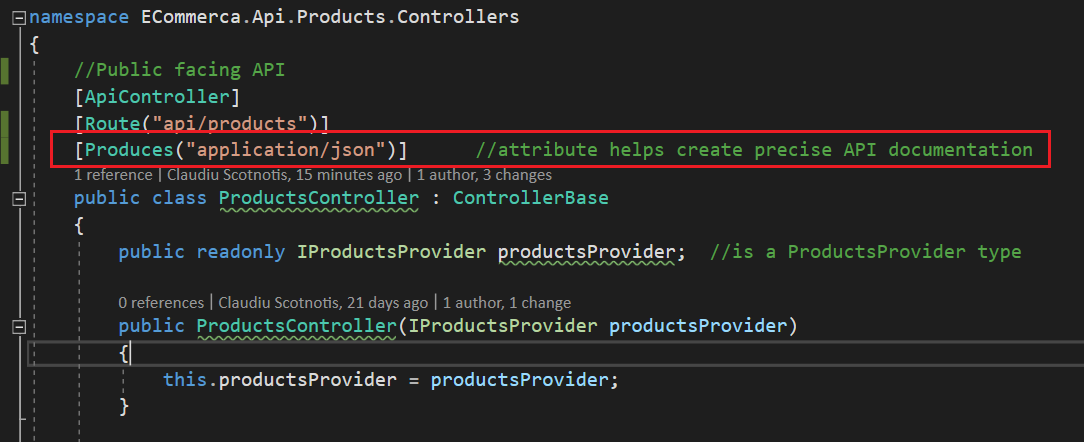
Graphical user interface, text

Description automatically generated

1. Also in Startup.cs, add in Configure method the following:



1. In the Controller:
   1. add at class level 

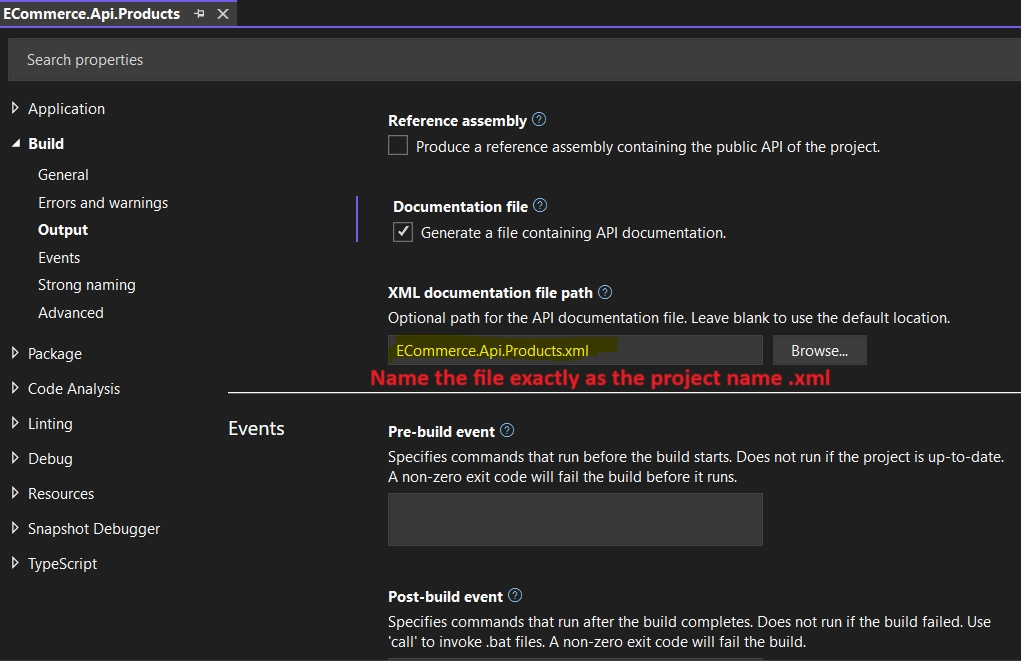


* 1. add **/// XML comments** to each API method
  2. add associated data annotation **[ProducesResponseType(StatusCodes…)]**
  3. Add the necessary 

Text

Description automatically generated

1. Setup where to generate the XML document containing the comments:
   1. Right click on the Ecommerce.Api.Products Project 🡺 Properties 🡺 Build
   2. Name the XML file **exactly as the project name** .xml.   
      Example: Ecommerce.Api.Products.xml
   3. Save the changes
   4. Do this for all 4 services.

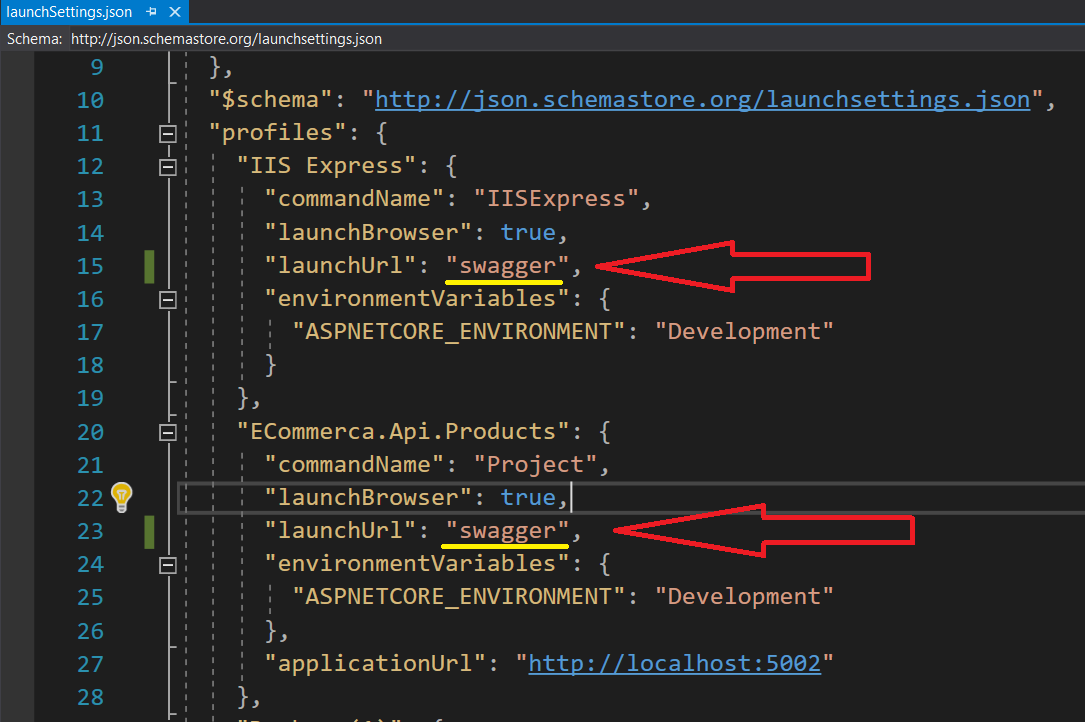


1. Run the application locally exactly as such:

Graphical user interface, application

Description automatically generated

1. For all 4 Microservices, modify the launchSettings.json to open automatically to swagger for both IIS and Kestrel.



1. Run the application and you should see the Swagger documentation:
   1. <URL:port/swagger>

Graphical user interface, text, application, email

Description automatically generated

1. For each API call, add **Sample Request** code to the XML to look like this.Text

   Description automatically generated with low confidence
2. **Congratulations** you have added Swagger documentation to your Microservices.
3. **~~Bonus 5%:~~** ~~Deploy the application with Swagger documentation for all microservices on Azure and provide a how-to guide.~~
4. **IMPORTANT:** In Azure, delete the Fabric and all dependent services, in order to not get charged more. You need the credit in the following semester.
5. **Attach a screenshot from Azure proving that you do not have any more services running on Lea.**