Laboratory assignment №1

**Topic:** Introduction to ASP.NET Core

**Goal:** get acquainted with the basic principles of .NET, нlearn how to configure the development environment and install the necessary components, acquire skills in creating solutions and projects of various types, acquire skills in processing requests using middleware.

Workflow:

# Task 1. Installation of integrated development environment (IDE) and necessary workloads

***If you already have Visual Studio installed on your PC, please follow*** [***this link***](#_igpeyl484ojb)***.***

The most convenient way to work with ASP.NET Core, is to use the Visual Studio IDE. Currently, the latest version is Visual Studio 2022. To install, follow the link, scroll through the banner at the top of the page and click the “Download Visual Studio” button, then select **Community 2022** from the drop-down list (Fig. 1.1.)

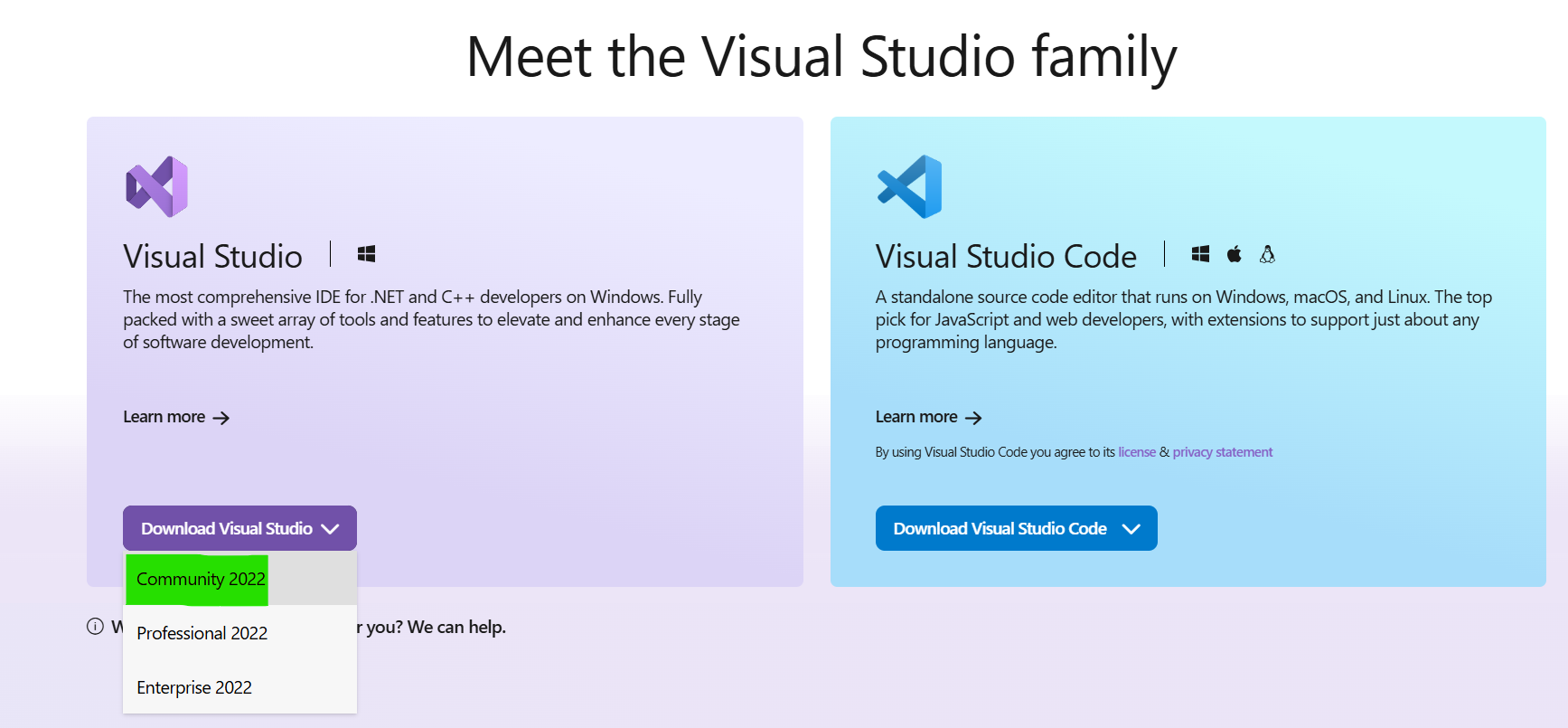


Fig. 1.1. Downloading Visual Studio Installer

Then the Visual Studio Installer will start downloading. After downloading is finished, run the downloaded exe file and wait for the installation to complete. After the Visual Studio Installer installation is complete, you will be able to select the necessary components. **Important: in the Web & Cloud section, you must select the component called ASP.NET and web development (Fig. 1.2.).** In the next step, the installer will offer to select individual components for installation (Individual Components tab), in this step we should leave everything by default.

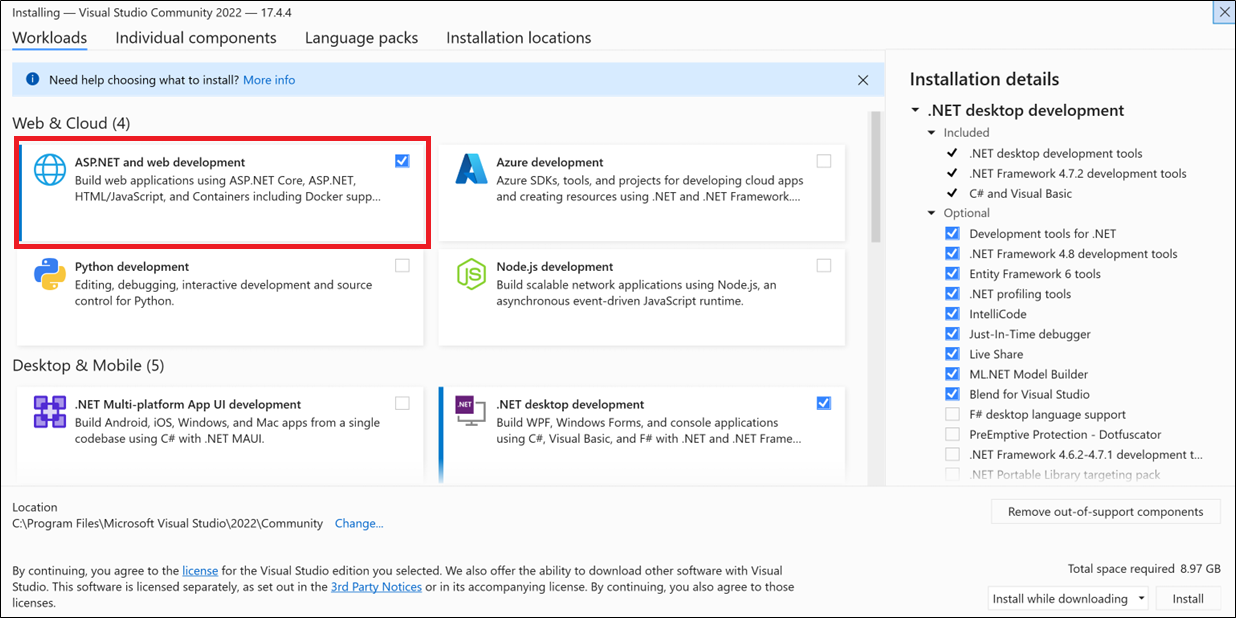


Fig. 1.2. Selection of necessary components to install

On the Language Packs tab installer provides an opportunity to choose desired language packs. **It is recommended to choose English (Fig. 1.3.).**

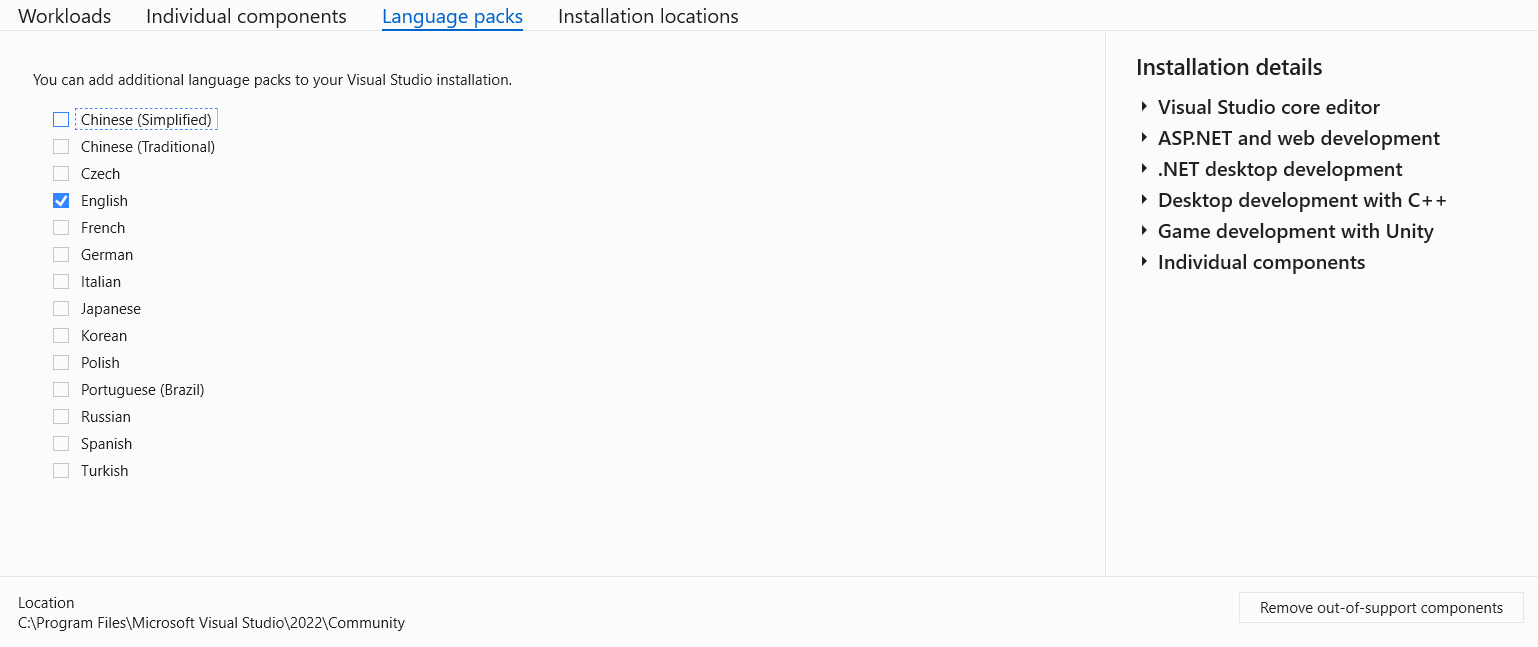


Fig. 1.3. Selection of language packs

On the Installation locations tab you can choose the desired folder where Visual Studio will be installed, the rest of the options can be left as default (Fig. 1.4.)

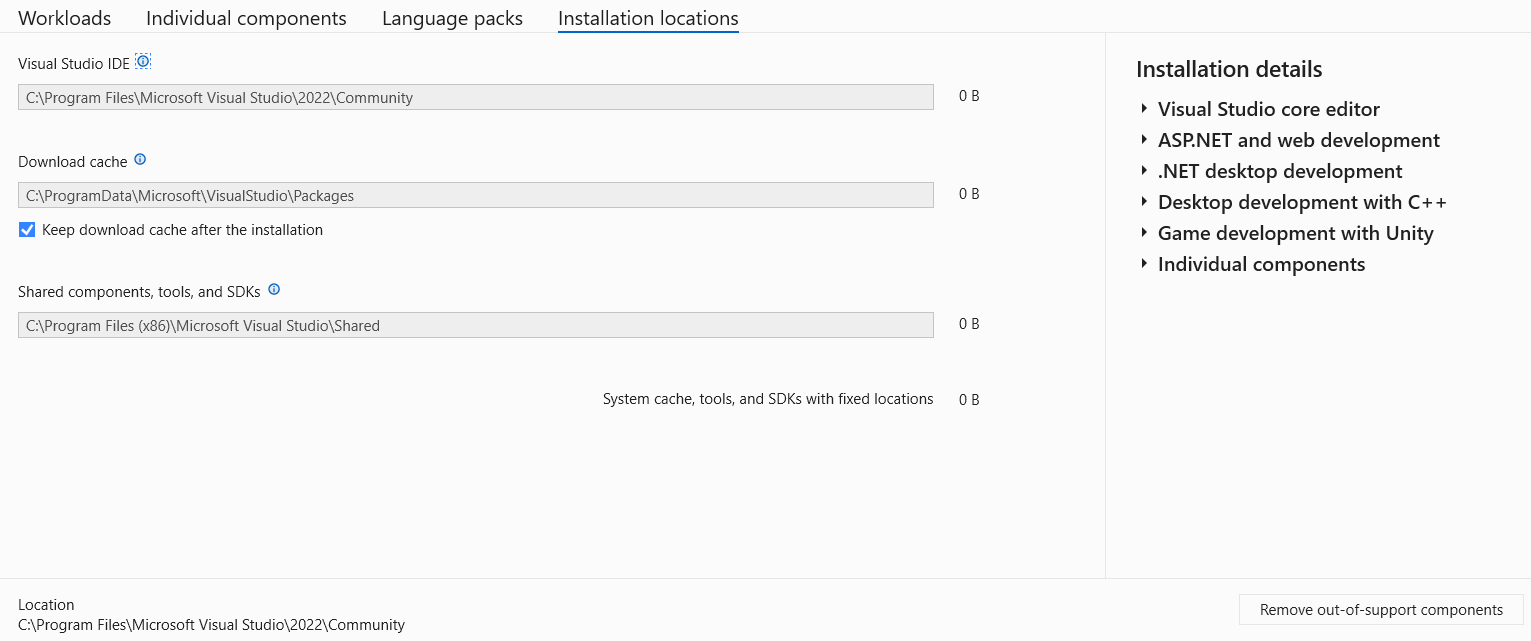


Рис. 1.4. Вибір директорії для встановлення

After that, the process of installing Visual Studio and all the necessary components for developing on ASP.NET Core will begin.

If desired, you can configure Visual Studio Code to work with .NET, detailed instructions are available [at the link](https://dotnet.microsoft.com/en-us/learn/aspnet/blazor-tutorial/install).

After the installation is complete, you can launch Visual Studio in any convenient way, the initial window is shown in Figure 1.5.

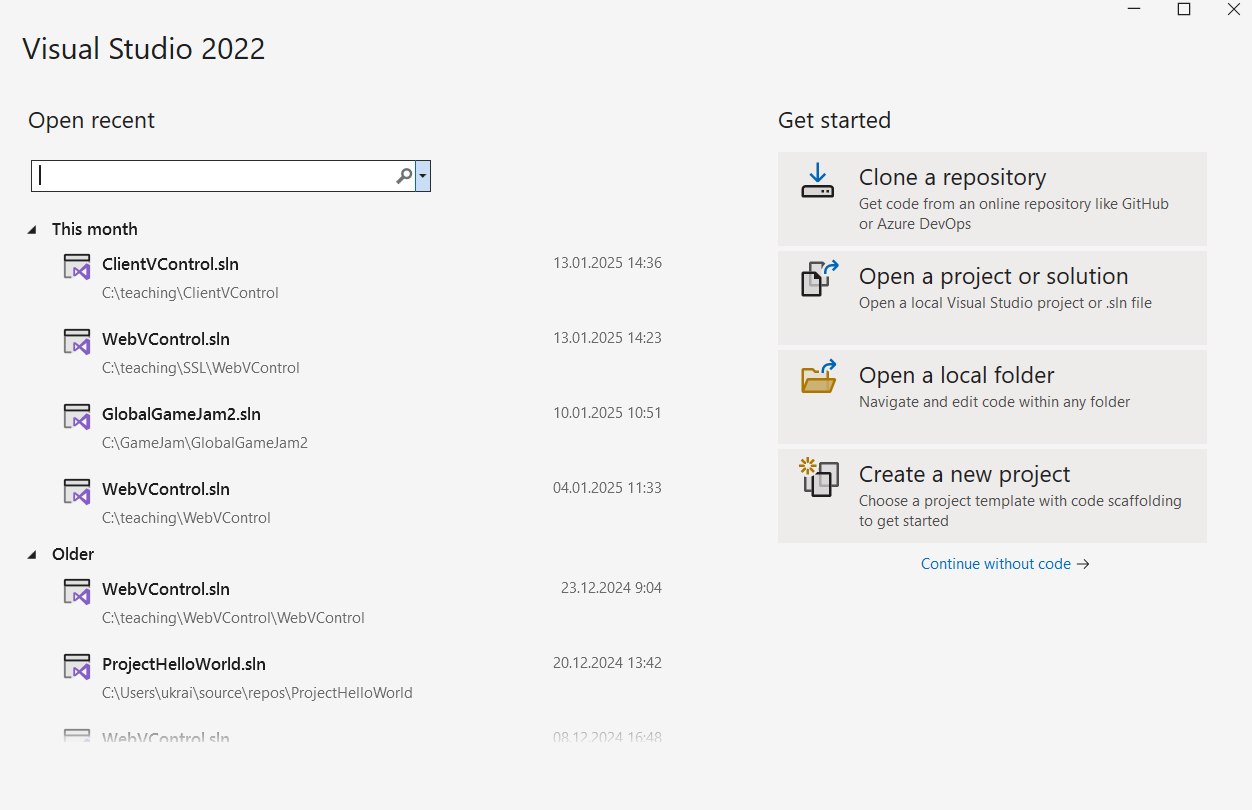


Fig. 1.5. The initial window of the Visual Studio

## How to act if the Visual Studio is already installed on your PC

Open the Visual Studio Installer. Then, on the Visual Studio installation, click the “Modify” button (Fig. 1.6.).

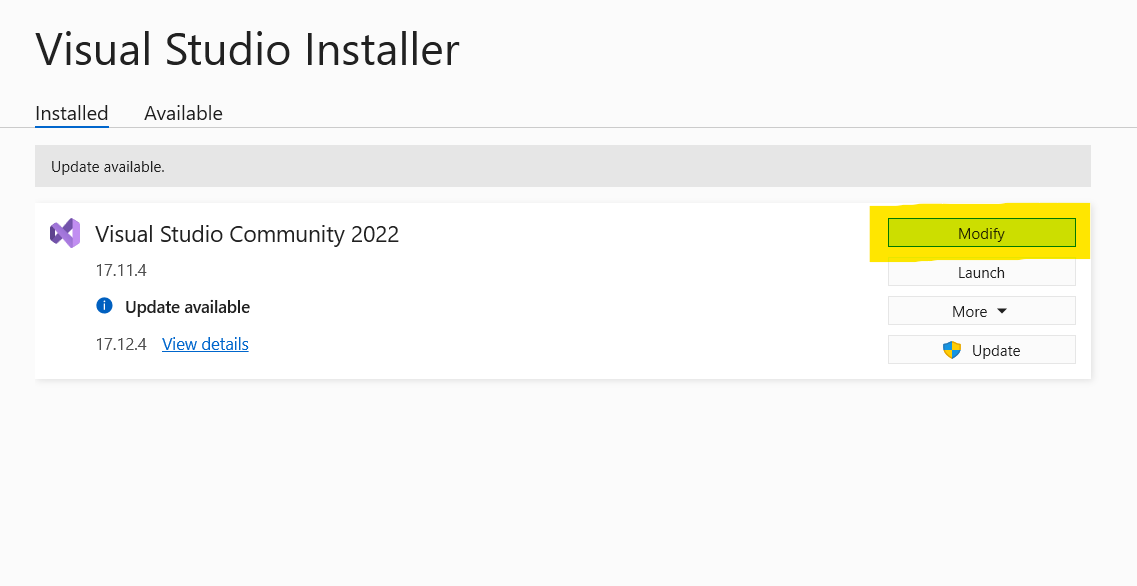


Fig. 1.6. Installation modification

After that, the component selection window will open. In the Web & Cloud section, you must select the workload called **ASP.NET and web development** if it is not already selected (Fig. 1.7).

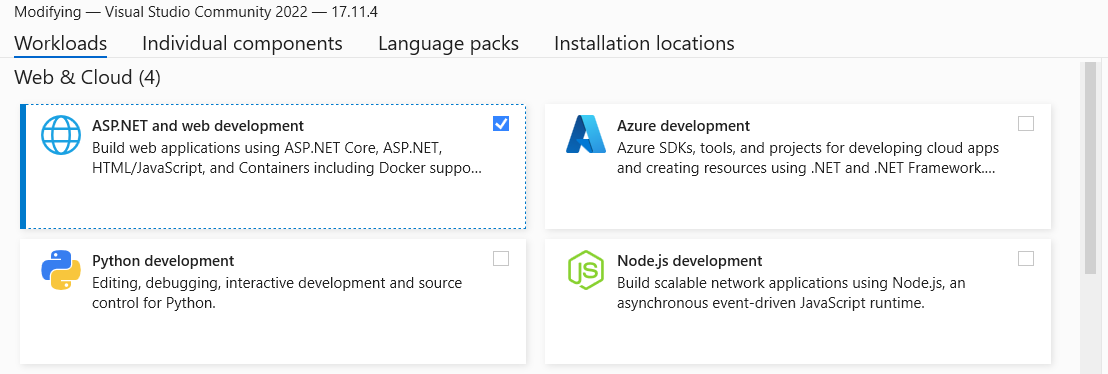


Fig. 1.7. Installation of necessary workloads

After that, click the “Modify” button in the lower right corner and wait for the installation to complete.

# Task 2. Project creation

**Part 1.**

1. Create a project for a console application called ConsoleToWeb using the [dotnet CLI](https://learn.microsoft.com/en-us/dotnet/core/tools/dotnet-new)
2. Convert the created console application into a web application
3. Describe the steps taken in the report

**Part 2.**

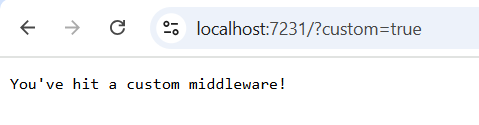
1. Create an ASP.NET WebAPI project without authorization named WebFromCli using [dotnet CLI](https://learn.microsoft.com/en-us/dotnet/core/tools/dotnet-new)
2. Implement a GET endpoint “/who” that will return your first and last name
3. Implement a GET endpoint “/time” that will return the current time on the server
4. Provide a listing of implemented handlers in the report

**Part 3.**

1. Create an ASP.NET MVC project in any convenient way
2. Implement a controller named LabController
3. In the created controller, implement the /info handler, which will return a View with data about the laboratory work number, topic, purpose, and the name and surname of the performer in a table.
4. Data for table rendering should be transferred from the controller

# Task 3. Middlewares

1. Learn about [the concept of middleware](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/middleware/?view=aspnetcore-9.0) in ASP.NET
2. Create an ASP.NET WebAPI project named MiddlewareSandbox
3. Implement the following middlewares
   1. Implement a middleware that counts the number of requests to the server and returns this number in the response, for example: The amount of processed requests is X.
   2. Implement a middleware that parses the query string parameters. If the query string contains the parameter “custom”, then return “You’ve hit a custom middleware!”, otherwise skip the request. An example of a request URL is shown in Figure 3.1.

  
Figure. 3.1. Request example

* 1. Implement a middleware that logs all requests to the console. The logs should show the request method (GET, POST, etc.) and its path (/user, /product, etc.). You can use Postman or other similar utility to test POST and other request methods.
  2. Implement middleware that checks for the presence of an API key in the request headers. To do this, check whether the request contains the X-API-KEY header. If the key is incorrect or missing, return 403 Forbidden, do not pass the request on for further execution. If the key in the request header matches the one specified on the server, pass the request on. You can use Postman or similar utilities to add the corresponding request.

# Self-checkup questions.

1. What is the .NET platform? How is the diversity of languages ​​and cross-platformness of applications developed on .NET are achieved?
2. What were the stages of development of ASP.NET technology? How is ASP.NET different from ASP.NET Core?
3. What types of applications can be developed using ASP.NET?
4. What is a REST API?
5. Describe how the middleware works.