Binding Simple Types to a Request （一）

- For simple types such as int, string, and double, ASP.NET Core automatically tries to bind the value to a route parameter, or a query string value:
  - If the parameter name matches a route parameter name, binds to the associated route value.
  - If the parameter name doesn't match any route parameter name, bind to a query string value.

70

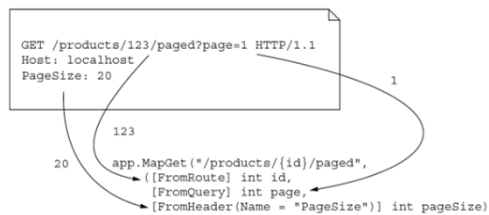### Binding Simple Types to a Request （二）

- Example
  - /products/123
  - /products?id=456

```
app.MapGet("/products/{id?}", (int id) => $"Received {id}");
```

71

Binding Simple Types to a Request （三）

- The [From*] attributes override ASP.NET Core's default logic and forces the parameters to load from a specific binding source.

```
GET /products/123/paged?page=1 HTTP/1.1
Host: localhost
PageSize: 20
                                              1
            123
    20   app.MapGet("/products/{id}/paged",
          ([FromRoute] int id,
           [FromQuery] int page,
           [FromHeader(Name = "PageSize")] int pageSize)   72
```

Binding Complex Types to the JSON Body

- Minimal APIs can bind the body of a request to a single complex type by deserializing the request from JSON, create the object.

```
app.MapPost("/product", (Product product) => $"Received {product}");
record Product(int Id, string Name, int Stock);

{ "id": 1, "Name": "Shoes", "Stock": 12 }
```

73

THANK YOU

ASP.NET MVC

1. ASP.NET Core MVC
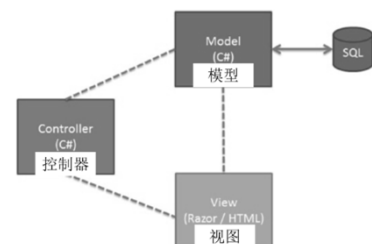
Structure of ASP.NET Core

- ASP.NET Core is Microsoft's web development platform.
- Most of your development time will be spent with the application frameworks.
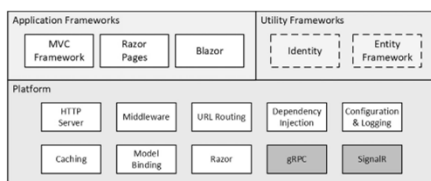
5

MVC Pattern（一）

- The MVC design pattern is a common pattern for designing apps that have UIs.

6

1

MVC Pattern（二）

- Three components make up the MVC pattern:
  - **Model**—The data that needs to be displayed.
  - **View**—The template that displays the data provided by the model.
  - **Controller**—Updates the model and provides the data for display to the view.
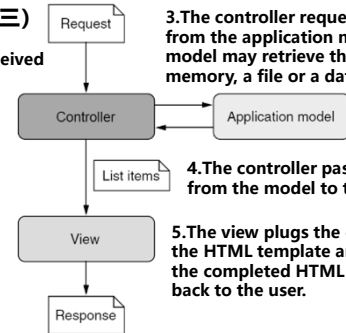
7

MVC Pattern（三）

1.Request is received from a user.

2.The controller handles the request.

3.The controller requests data from the application model. The model may retrieve them from memory, a file or a database.

4.The controller pass the data from the model to the view.

5.The view plugs the data into the HTML template and sends the completed HTML page back to the user.

8

2. Configuring an ASP.NET Core MVC Application

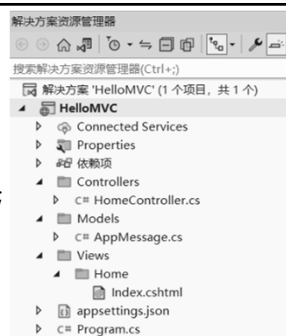## You First Application   CodeLab 3.1

- You are required to develop an application that receives a message from user and display it in the web page.


10

Project Structure

- The Program.cs file contains the code statements that are executed when the application is started.
  - Configure the ASP.NET platform;
  - Configure the individual frameworks it supports.

11

Services

- Services are objects that provide features in a web application.
- They are managed by the ASP.NET Core platform that can be shared by middleware components.

12
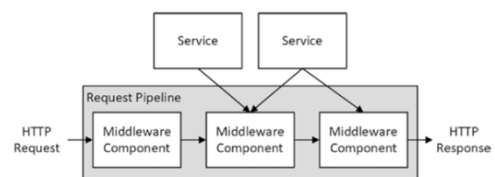
2

Configuration Service（一）

- Configuration is the set of external parameters provided to an application that controls the application's behavior in some way.
- It typically consists of a mixture of settings and secrets that the application loads at runtime.
  - A setting is value that changes the behavior of your application.
  - A secret is a special type of setting that contains sensitive data, such as a password.
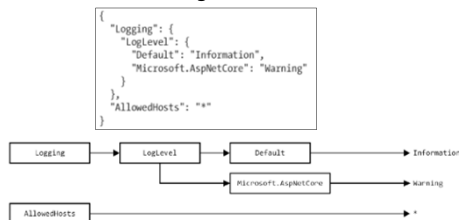
13

Configuration Service（二）

- One of the built-in features is access to the application's configuration settings, which is presented as a service.
- The main source of configuration data is the **appsettings.json** file.
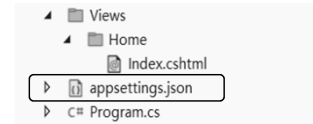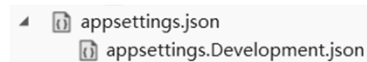

14

Configuration Service（三）

- The configuration service will process the JSON configuration file and create nested configuration sections.

15
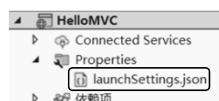
Configuration Service（四）

- Most projects contain more than one JSON configuration file, allowing different settings to be defined for different parts of the development cycle.
  - Development
  - Staging
  - Production

16

Launch Settings

- The launchSettings.json file contains the configuration settings for starting the ASP.NET Core platform:
  - The TCP ports that are used to listen for HTTP and HTTPS requests;
  - The environment used to select the additional JSON configuration files.

17