# ASP.NET Core in Action

## *CHAPTER 1*
## *Getting started with ASP.NET Core*

**1. What is ASP.NET Core primarily used for?**
a) Building mobile applications
b) Creating desktop applications
c) Building dynamic, server-side rendered applications and HTTP APIs
d) None of the above
Answer: c) Building dynamic, server-side rendered applications and HTTP APIs

**2. Which of the following is NOT a characteristic of ASP.NET Core?**
a) Cross-platform
b) Closed source
c) Open source
d) Web application framework
Answer: b) Closed source

**3. What does ASP.NET Core provide to developers?**
a) Ready-made applications
b) Basic HTML templates
c) Structure, helper functions, and a framework for building applications
d) Hosting services
Answer: c) Structure, helper functions, and a framework for building applications

**4. In ASP.NET Core, where does the business logic typically reside?**
a) Directly within ASP.NET Core framework code
b) In the HTML templates
c) Within separate handlers that call methods in the application's business logic
d) In the database
Answer: c) Within separate handlers that call methods in the application's business logic

**5. What is one of the purposes of ASP.NET Core's business logic?**
a) Directly interact with ASP.NET Core framework code
b) Serve as a template engine
c) Handle requests from the client
d) Interact with other services such as databases or remote APIs
Answer: d) Interact with other services such as databases or remote APIs

**6. Why might building a static web application without a web framework limit its capabilities?**

a) It will consume more server resources
b) It will be less secure
c) It will require less coding effort
d) It will lack features such as security and dynamism

Answer: d) It will lack features such as security and dynamism

**7. What is one advantage of using ASP.NET Core for web development?**
a) It requires extensive coding from scratch
b) It doesn't support dynamic web pages
c) It makes writing web applications faster, easier, and more secure
d) It limits the ability to provide user authentication
Answer: c) It makes writing web applications faster, easier, and more secure

**8. What is a common functionality provided by ASP.NET Core libraries?**
a) Creating desktop applications
b) Generating static web pages
c) Logging requests made to a web app
d) Handling server hardware management

Answer: c) Logging requests made to a web app

**9. What is the key feature of a dynamic web page?**
a) It displays static content only
b) It relies heavily on client-side scripting
c) It requires no user interaction
d) It may display different data depending on factors like user login status

Answer: d) It may display different data depending on factors like user login status

**10. Why are dynamic frameworks essential for websites like Amazon, eBay, and Stack Overflow?**
a) They enhance server performance
b) They simplify frontend development
c) They provide advanced security features
d) They enable personalized content and user authentication

Answer: d) They enable personalized content and user authentication

**11. What were the main goals behind the development of ASP.NET Core?**
a) To focus exclusively on Windows development
b) To create a heavy and monolithic framework
c) To achieve cross-platform compatibility, modular architecture, open-source development, and alignment with current web development trends
d) To prioritize compatibility with legacy web technologies

Answer: c) To achieve cross-platform compatibility, modular architecture, open-source development, and alignment with current web development trends

**12. What platform does ASP.NET Core run on?**

a) Windows only
b) Linux only
c) macOS only
d) Windows, Linux, and macOS

Answer: d) Windows, Linux, and macOS

**13. What is the purpose of .NET 5.0 in relation to ASP.NET Core?**

a) .NET 5.0 is a successor to ASP.NET Core
b) .NET 5.0 is a lightweight platform for building web applications
c) .NET 5.0 provides a runtime and framework for ASP.NET Core to run cross-platform
d) .NET 5.0 is not related to ASP.NET Core

Answer: c) .NET 5.0 provides a runtime and framework for ASP.NET Core to run cross-platform

**14. How does .NET Core differ from .NET Framework?**

a) .NET Core is heavier and more complex than .NET Framework
b) .NET Core implements more features than .NET Framework
c) .NET Core is a fork of .NET Framework
d) .NET Core is more modular and implements only a subset of the features of .NET Framework

Answer: d) .NET Core is more modular and implements only a subset of the features of .NET Framework

**15. How does ASP.NET Core relate to console applications?**

a) ASP.NET Core and console applications are entirely separate and unrelated
b) Console applications are built on top of ASP.NET Core
c) ASP.NET Core is an additional layer on top of console applications for converting them into web applications
d) Console applications are a prerequisite for using ASP.NET Core

Answer: c) ASP.NET Core is an additional layer on top of console applications for converting them into web applications

**16. What is Microsoft's recommendation regarding the use of ASP.NET Core for new .NET web development?**

a) Microsoft advises against using ASP.NET Core for new projects
b) Microsoft recommends using ASP.NET Core only for small-scale applications
c) Microsoft recommends that all new .NET web development should use ASP.NET Core
d) Microsoft suggests using ASP.NET Core only for legacy applications

Answer: c) Microsoft recommends that all new .NET web development should use ASP.NET Core

**17. What is a consideration for developers or companies before switching to ASP.NET Core?**

a) ASP.NET Core requires extensive prior knowledge of web development

b) ASP.NET Core is not compatible with modern web development practices
c) Switching to ASP.NET Core involves a significant learning curve and investment
d) ASP.NET Core offers no advantages over other web development frameworks
Answer: c) Switching to ASP.NET Core involves a significant learning curve and investment

**18. What topics are covered in the section discussing when to choose ASP.NET Core?**
a) Basic syntax of ASP.NET Core
b) The history of web development
c) Building applications with ASP.NET Core
d) What sort of applications you can build, highlights of ASP.NET Core, why to consider using it, and considerations for converting existing ASP.NET applications

Answer: d) What sort of applications you can build, highlights of ASP.NET Core, why to consider using it, and considerations for converting existing ASP.NET applications

**19. What is one of the highlights of ASP.NET Core mentioned in the section?**
a) Limited compatibility with third-party libraries
b) Lack of support for cross-platform development
c) Increased security features
d) Easy integration with legacy ASP.NET applications

Answer: c) Increased security features

**20. What does Microsoft's recommendation imply about the future of ASP.NET Core?**
a) ASP.NET Core is likely to be deprecated soon
b) ASP.NET Core will remain the primary web development framework for .NET
c) ASP.NET Core will be replaced by another framework in the near future
d) ASP.NET Core will become obsolete within a few years

Answer: b) ASP.NET Core will remain the primary web development framework for .NET

**21. What is HTTP primarily used for?**
a) Sending emails
b) Transferring files between servers
c) Powering the web through stateless request-response interactions
d) Hosting websites

Answer: c) Powering the web through stateless request-response interactions

**22. What does a typical HTTP request consist of?**
a) Only a verb indicating the type of request
b) Only a path indicating the resource to interact with
c) Only headers indicating key-value pairs
d) A verb indicating the type of request, a path indicating the resource to interact with, headers, and optionally a body

Answer: d) A verb indicating the type of request, a path indicating the resource to interact with, headers, and optionally a body

**23. What information does an HTTP response contain?**

a) Only the contents of the requested resource
b) Only headers indicating key-value pairs
c) Only a status code indicating whether the request was successful
d) A status code indicating whether the request was successful, optionally headers, and optionally a body

Answer: d) A status code indicating whether the request was successful, optionally headers, and optionally a body

**24. What is the term used for the "type" of request in an HTTP request?**

a) Verb
b) Noun
c) Predicate
d) Adjective

Answer: a) Verb

**25. How does HTTP handle state?**

a) It maintains session information between client and server
b) It is a stateful protocol
c) It is a stateless protocol
d) It relies on cookies to maintain state

Answer: c) It is a stateless protocol

**26. What is ASP.NET Core primarily recommended for?**

a) Legacy projects
b) Existing, well-established projects
c) New, "green-field" projects
d) Small-scale personal projects

Answer: c) New, "green-field" projects

**27. Can legacy technologies such as WCF Server and Web Forms be used with ASP.NET Core?**

a) Yes
b) No
c) Only with additional plugins
d) It depends on the version of ASP.NET Core

Answer: b) No

**28. What platform does ASP.NET Core run on?**

a) .NET Framework
b) .NET Core only
c) .NET 5.0 only
d) .NET 5.0 with the Windows Compatibility Pack

Answer: d) .NET 5.0 with the Windows Compatibility Pack

**29. What is .NET 5.0 in relation to .NET Core?**
a) A separate framework from .NET Core
b) A downgrade from .NET Core
c) An earlier version of .NET Core
d) The next version of .NET Core after .NET Core 3.1

Answer: d) The next version of .NET Core after .NET Core 3.1

**30. How does ASP.NET Core handle fetching web pages?**
a) It doesn't handle fetching web pages
b) By directly fetching web pages from the server
c) By sending an HTTP request and receiving an HTTP response
d) By using JavaScript only

Answer: c) By sending an HTTP request and receiving an HTTP response

# *CHAPTER 2*
# *Your first application*

**1. What is the purpose of the "dotnet restore" command?**
a) Compiling the application
b) Running the application
c) Ensuring NuGet dependencies are copied to the project folder
d) Generating project files

Answer: c) Ensuring NuGet dependencies are copied to the project folder

**2. Where are dependencies of ASP.NET Core projects listed?**
a) In a separate text file
b) In the project's source code
c) In the project's .csproj file
d) In a configuration file

Answer: c) In the project's .csproj file

**3. What happens during the "dotnet build" command?**
a) NuGet packages are restored
b) The application is run
c) Any errors in the application are checked

d) NuGet packages are downloaded

Answer: c) Any errors in the application are checked

**4. When might it be useful to run "dotnet restore" explicitly?**
a) Before writing code for the application
b) After running "dotnet build"
c) In continuous-integration build pipelines
d) Only when using Visual Studio

Answer: c) In continuous-integration build pipelines

**5. How can you view the full list of available commands for the .NET CLI?**
a) Run "dotnet --all"
b) Run "dotnet list"
c) Run "dotnet --help"
d) Run "dotnet commands"

Answer: c) Run "dotnet --help"

**6. What is the main responsibility of the Startup class in an ASP.NET Core application?**
a) Handling HTTP requests
b) Managing database connections
c) Configuring service registration and middleware
d) Generating HTML templates

Answer: c) Configuring service registration and middleware

**7. What is service registration in the context of the Startup class?**
a) Registering users for the application
b) Configuring endpoint routes
c) Registering classes that provide functionality for the application
d) Configuring database migrations

Answer: c) Registering classes that provide functionality for the application

**8. Where is service registration typically configured in the Startup class?**
a) In the ConfigureServices method
b) In the ConfigureEndpoints method
c) In the Configure method
d) In the ConfigureMiddleware method

Answer: a) In the ConfigureServices method

**9. What is middleware configuration in the context of the Startup class?**
a) Configuring authentication and authorization policies
b) Registering service dependencies

c) Handling and responding to HTTP requests
d) Configuring database connections

Answer: c) Handling and responding to HTTP requests

**10. In which method of the Startup class is middleware configuration typically performed?**
a) ConfigureServices
b) ConfigureEndpoints
c) ConfigureServices
d) Configure

Answer: d) Configure

# *CHAPTER 3*

## *Handling requests with the middleware pipeline*

**1. What is middleware in the context of ASP.NET Core?**
a) A piece of hardware used for network communication
b) A software component responsible for generating HTML pages
c) C# classes that handle HTTP requests or responses
d) A type of database management system

Answer: c) C# classes that handle HTTP requests or responses

**2. What are some functions that middleware can perform in ASP.NET Core?**
a) Generating HTML pages and API responses
b) Logging each request and adding security headers to the response
c) Compiling C# code and managing database connections
d) Handling user authentication and managing session state

Answer: b) Logging each request and adding security headers to the response

**3. What is the most important piece of middleware in most ASP.NET Core applications?**
a) AuthenticationMiddleware
b) LoggerMiddleware
c) EndpointMiddleware
d) RoutingMiddleware

Answer: c) EndpointMiddleware

**4. What is the arrangement of middleware components called in ASP.NET Core?**
a) Pipeline
b) Assembly
c) Bundle
d) Stack

Answer: a) Pipeline

**5. What are some common cross-cutting concerns that middleware can handle?**
a) Generating dynamic HTML content
b) Managing database transactions
c) Logging each request and adding security headers to the response
d) Handling user authentication and authorization

Answer: c) Logging each request and adding security headers to the response

**6. What is the benefit of having highly focused middleware components in ASP.NET Core?**
a) They make the application more complex
b) They handle multiple concerns simultaneously
c) They are easier to reason about and maintain
d) They restrict the flexibility of the application

Answer: c) They are easier to reason about and maintain

**7. How are multiple middleware components combined in ASP.NET Core?**
a) By nesting them within each other
b) By chaining them together in a sequence called a pipeline
c) By merging their functionalities into a single component
d) By grouping them into separate assemblies

Answer: b) By chaining them together in a sequence called a pipeline

**8. What does each middleware component have access to in the ASP.NET Core pipeline?**
a) Only the original request
b) Only the final response
c) Both the original request and any changes made by previous middleware
d) Only the changes made by previous middleware

Answer: c) Both the original request and any changes made by previous middleware

**9. What can middleware components do with the response in the ASP.NET Core pipeline?**
a) They can only inspect the response
b) They can modify the response before sending it to the user
c) They cannot interact with the response
d) They can only send the response back to the client

Answer: b) They can modify the response before sending it to the user

**10. How does composing multiple middleware components benefit ASP.NET Core applications?**
a) It increases application complexity
b) It decreases application flexibility
c) It allows for the creation of complex application behaviors from small, focused components
d) It reduces the number of components required

Answer: c) It allows for the creation of complex application behaviors from small, focused components

**11. What is the purpose of HTTP response status codes?**
a) To indicate the size of the response body
b) To specify the language of the response
c) To provide information about the client's device
d) To communicate the result of the request handling

Answer: d) To communicate the result of the request handling

**12. What does a status code starting with "2xx" indicate?**
a) Informational response
b) Client error
c) Redirection
d) Success

Answer: d) Success

**13. In which class of status codes would you find a response indicating that the browser must follow a provided link?**
a) 1xx
b) 2xx
c) 3xx
d) 4xx

Answer: c) 3xx

**14. What type of error is indicated by a status code starting with "5xx"?**
a) Client error
b) Success
c) Server error
d) Redirection

Answer: c) Server error

**15. When might a browser automatically handle a 301 response?**
a) When the request is successful
b) When the client sends invalid data

c) When the resource requested couldn't be found
d) When the server encounters an error

Answer: c) When the resource requested couldn't be found

**16. In ASP.NET Core, how is the design philosophy regarding features like error handling described?**
a) Features are automatically enabled by default
b) Features must be explicitly enabled
c) Features are enabled based on user preferences
d) Features are only available in the premium version

Answer: b) Features must be explicitly enabled

**17. What is one common type of error that can occur in an application?**
a) NullReferenceException
b) AuthenticationFailureException
c) IndexOutOfBoundsException
d) SyntaxErrorException

Answer: a) NullReferenceException

**18. What happens if an exception occurs in a middleware component in ASP.NET Core?**
a) The exception is silently ignored
b) The exception is logged and the application continues running
c) The exception propagates up the middleware pipeline
d) The application crashes immediately

Answer: c) The exception propagates up the middleware pipeline

**19. What status code is typically returned to the user if an unhandled exception occurs in ASP.NET Core?**
a) 200
b) 404
c) 500
d) 301

Answer: c) 500

**20. How can you ensure graceful handling of errors in an ASP.NET Core application?**
a) By ignoring all exceptions
b) By explicitly enabling error handling features
c) By crashing the application whenever an error occurs
d) By relying on the web server to handle errors automatically

Answer: b) By explicitly enabling error handling features

**21. What role does middleware play in ASP.NET Core?**
a) Managing database connections

b) Handling HTTP requests and responses
c) Rendering HTML templates
d) Managing user sessions

Answer: b) Handling HTTP requests and responses

**22. How is middleware composed in ASP.NET Core?**
a) In a stack
b) In a queue
c) In a pipeline
d) In a loop

Answer: c) In a pipeline

**23. What happens if a middleware short-circuits the pipeline?**
a) All subsequent middleware will still execute
b) None of the subsequent middleware will execute
c) Only the response will be affected
d) Only the request will be affected

Answer: b) None of the subsequent middleware will execute

**24. What does StaticFileMiddleware do when added to a middleware pipeline?**
a) It serves any requested files found in the application's database
b) It renders HTML templates for static files
c) It provides custom error handling messages
d) It serves any requested files found in the wwwroot folder of the application

Answer: d) It serves any requested files found in the wwwroot folder of the application

**25. When should DeveloperExceptionPageMiddleware be used?**
a) In production
b) In testing
c) In development
d) In staging

Answer: c) In development

# *CHAPTER 4*
# *Creating a website with Razor Pages*

**1. What is the purpose of Razor Pages in ASP.NET Core?**
a) To build single-page applications (SPAs)
b) To provide a streamlined experience for building server-side rendered websites
c) To handle database operations

d) To manage client-side interactions

Answer: b) To provide a streamlined experience for building server-side rendered websites

**2. How does a page-based website differ from other types of applications?**
a) It heavily relies on client-side interactions
b) It is not interactive
c) It consists of multiple pages and forms for user interaction
d) It does not involve browsing between pages

Answer: c) It consists of multiple pages and forms for user interaction

**2. Which version of ASP.NET Core introduced the Razor Pages programming model?**
a) ASP.NET Core 1.0
b) ASP.NET Core 2.0
c) ASP.NET Core 3.0
d) ASP.NET Core 4.0

Answer: b) ASP.NET Core 2.0

**3. How does Razor Pages utilize conventions to reduce boilerplate code?**
a) By eliminating the need for HTML
b) By providing built-in database management
c) By automatically generating JavaScript code
d) By simplifying routing and page structure

Answer: d) By simplifying routing and page structure

**4. What will you understand by the end of the section on Razor Pages?**
a) How to build single-page applications
b) The MVC design pattern
c) The overall design of Razor Pages and their relation to the MVC pattern
d) How to manage client-side interactions

Answer: c) The overall design of Razor Pages and their relation to the MVC pattern

**5. What is the primary goal of the MVC design pattern?**
a) To focus solely on graphical user interface (GUI) apps
b) To separate the management and manipulation of data from its visual representation
c) To eliminate the need for data manipulation
d) To simplify the visual representation of data

Answer: b) To separate the management and manipulation of data from its visual representation

**6. What does the MVC pattern aim to achieve for applications with UIs?**
a) Combine data management with visual representation

b) Provide a single model for data and visual elements
c) Separate concerns related to data, presentation, and user interaction
d) Minimize user interaction in favor of automated processes

Answer: c) Separate concerns related to data, presentation, and user interaction

**7. Which type of applications was the original MVC pattern primarily designed for?**
a) Web applications
b) Command-line applications
c) Graphical user interface (GUI) apps
d) Mobile applications

Answer: c) Graphical user interface (GUI) apps

**8. How does the MVC pattern handle requests in an application?**
a) By combining data management with visual representation
b) By separating request handling into distinct components
c) By directly manipulating the visual elements of the application
d) By focusing solely on data manipulation

Answer: b) By separating request handling into distinct components

**9. What aspect of the MVC pattern helps in generating the final response to a request?**
a) The data management component
b) The visual representation component
c) The separation of concerns
d) The integration with GUI environments

Answer: c) The separation of concerns

**10. Which design pattern is commonly associated with Razor Pages in ASP.NET Core?**
a) Model-View-Controller (MVC)
b) Model-View-View-Model (MVVM)
c) Model-View-Presenter (MVP)
d) Model-View-Adapter (MVA)

Answer: a) Model-View-Controller (MVC)

**11. How does the MVVM pattern differ from the MVC pattern?**
a) MVVM involves bidirectional interaction between the view and the model, while MVC focuses on unidirectional flow of data.
b) MVC relies on two-way data binding, whereas MVVM does not.
c) MVVM separates the view from the controller, while MVC does not.
d) MVC emphasizes encapsulating business logic in the view model, unlike MVVM.

Answer: a) MVVM involves bidirectional interaction between the view and the model, while MVC focuses on unidirectional flow of data.

**12. What is a characteristic feature of MVVM?**
a) Two-way data binding between view and model
b) Direct manipulation of the view by the controller
c) Encapsulation of data access logic in the view
d) Unidirectional flow of data from model to view

Answer: a) Two-way data binding between view and model

**13. Why does the author of the text disagree with describing Razor Pages as MVVM?**
a) Because Razor Pages are primarily used for desktop applications, not web applications
b) Because Razor Pages lack bidirectional interaction between the view and the PageModel
c) Because Razor Pages are based on the ASP.NET Core MVC framework and follow MVC principles
d) Because Razor Pages do not involve any interaction between the view and the model

Answer: c) Because Razor Pages are based on the ASP.NET Core MVC framework and follow MVC principles

**14. In which scenarios is MVVM commonly used?**
a) In server-side web development
b) In mobile and desktop applications
c) In low-level system programming
d) In network protocol design

Answer: b) In mobile and desktop applications

**15. In which scenario is it recommended to use MVC controllers instead of Razor Pages in ASP.NET Core?**
a) When building page-based server-side rendered applications
b) When developing a mobile application
c) When rendering views is not required, such as for Web APIs
d) When transitioning from an existing Razor Pages application

Answer: c) When rendering views is not required, such as for Web APIs

**16. What is a practical reason for not converting existing MVC controllers to Razor Pages?**
a) Razor Pages offer better performance compared to MVC controllers
b) Converting MVC controllers to Razor Pages requires extensive refactoring
c) Razor Pages lack support for partial page updates
d) MVC controllers provide better support for client-side JavaScript

Answer: b) Converting MVC controllers to Razor Pages requires extensive refactoring

**17. When might it be easier to achieve partial page updates using MVC controllers instead of Razor Pages?**
a) When developing a single-page application (SPA)
b) When using Web Forms for server-side rendering

c) When building page-based server-side rendered applications
d) When doing a lot of partial page updates with JavaScript

Answer: d) When doing a lot of partial page updates with JavaScript

**18. Which development approach is most suitable for building a Web API in ASP.NET Core?**
a) MVC controllers
b) Razor Pages
c) Blazor components
d) Web Forms

Answer: a) MVC controllers

**19. In which scenario is it recommended to use Razor Pages instead of MVC controllers?**
a) When building a Web API
b) When transitioning from an existing MVC application to ASP.NET Core
c) When doing a lot of partial page updates with JavaScript
d) When rendering views for page-based applications

Answer: d) When rendering views for page-based applications

**20. Which IActionResult type would you use to generate an HTML view for a given Razor view when using MVC controllers?**
a) PageResult
b) ViewResult
c) RedirectToPageResult
d) RedirectResult

Answer: b) ViewResult

**21. Which IActionResult type would you use to send a 302 HTTP redirect response to automatically send a user to another page in Razor Pages?**
a) RedirectToPageResult
b) RedirectResult
c) FileResult
d) ContentResult

Answer: a) RedirectToPageResult

**22. When would you use RedirectToPageResult over RedirectResult?**
a) When you need to return a file as the response
b) When you need to send a 404 HTTP status code
c) When you want to automatically send a user to another page in Razor Pages
d) When you want to send a user to a specified URL

Answer: c) When you want to automatically send a user to another page in Razor Pages

**23. Which IActionResult type would you use to return a file as the response?**

a) RedirectToPageResult
b) RedirectResult
c) FileResult
d) ContentResult

Answer: c) FileResult

**24. What does StatusCodeResult IActionResult type do?**
a) Generates an HTML view for an associated page in Razor Pages
b) Returns a provided string as the response
c) Sends a raw HTTP status code as the response
d) Sends a raw 404 HTTP status code as the response

Answer: c) Sends a raw HTTP status code as the response

# *CHAPTER 5*

# *Mapping URLs to Razor Pages using routing*

**1. What is routing in ASP.NET Core?**
a) The process of generating HTML views for Razor Pages
b) Mapping an incoming request to a method that will handle it
c) Handling cross-cutting concerns such as logging and error handling
d) Extracting data from a request's URL

Answer: b) Mapping an incoming request to a method that will handle it

**2. How does routing contribute to controlling the URLs exposed in an application?**
a) By defining the behavior of the application's middleware pipeline
b) By automatically extracting data from a request's URL
c) By mapping incoming requests to specific handlers
d) By enabling powerful features like logging and error handling

Answer: c) By mapping incoming requests to specific handlers

**3. Which middleware in ASP.NET Core is typically used to handle requests by invoking page handlers on Razor Pages or action methods on MVC controllers?**
a) EndpointMiddleware
b) StaticFileMiddleware
c) DeveloperExceptionPageMiddleware
d) StatusCodePagesMiddleware

Answer: a) EndpointMiddleware

**4. What is the purpose of routing in ASP.NET Core?**
a) To define the structure of middleware pipelines
b) To generate HTML views for Razor Pages
c) To map incoming requests to appropriate handlers
d) To handle cross-cutting concerns like logging and error handling

Answer: c) To map incoming requests to appropriate handlers

**5. What is endpoint routing in ASP.NET Core?**
a) A routing system restricted to Razor Pages only
b) A new routing system introduced in ASP.NET Core 3.0 that decouples routing from the MVC infrastructure
c) A routing middleware specifically designed for handling MVC requests
d) A convention-based routing system introduced in ASP.NET Core 2.0

Answer: b) A new routing system introduced in ASP.NET Core 3.0 that decouples routing from the MVC infrastructure

**6. What were some limitations of routing in ASP.NET Core 2.0 and 2.1?**
a) Restricted routing to MVC infrastructure only
b) Limited cross-cutting concerns like authorization to MVC components
c) Required the use of convention-based routing exclusively
d) Excluded endpoint routing as a routing option

Answer: b) Limited cross-cutting concerns like authorization to MVC components

**7. How does endpoint routing differ from previous routing systems in ASP.NET Core?**
a) It restricts routing to MVC infrastructure
b) It eliminates the need for convention-based routing
c) It allows other middleware in the application to use routing
d) It removes attribute routing support

Answer: c) It allows other middleware in the application to use routing

**8. What are the two types of routing available in ASP.NET Core?**
a) MVC routing and Razor routing
b) Convention-based routing and attribute routing
c) Endpoint routing and middleware routing
d) Razor Pages routing and Web API routing

Answer: b) Convention-based routing and attribute routing

**9. What is convention-based routing in ASP.NET Core?**
a) A routing approach that ties specific URLs to endpoints using attributes
b) A routing approach that defines URL-endpoint mappings globally for the application
c) A routing approach that relies on explicit attribute declarations on controller actions
d) A routing approach that is exclusively used for building Web APIs

Answer: b) A routing approach that defines URL-endpoint mappings globally for the application

**10. How does attribute routing differ from convention-based routing?**
a) Attribute routing is more verbose and requires applying attributes to every action method
b) Attribute routing is a convention-based approach for mapping URLs to endpoints
c) Attribute routing allows for greater flexibility by explicitly defining URLs for each action method
d) Attribute routing is restricted to Web API development only

Answer: c) Attribute routing allows for greater flexibility by explicitly defining URLs for each action method

**11. Which routing approach is commonly used for building HTML-based websites with MVC controllers?**
a) Attribute routing
b) Convention-based routing
c) Endpoint routing
d) Attribute-based routing

Answer: b) Convention-based routing

**12. In convention-based routing, how are endpoints mapped to URLs?**
a) By explicitly defining URLs for each action method
b) By placing [Route] attributes on action methods
c) By globally defining routing conventions for the application
d) By relying on the built-in routing conventions of ASP.NET Core

Answer: c) By globally defining routing conventions for the application

**13. What is a downside of convention-based routing compared to attribute routing?**
a) It requires applying attributes to every action method
b) It provides less flexibility in customizing URLs for specific controllers and actions
c) It is more verbose and complex to implement
d) It is restricted to Web API development

Answer: b) It provides less flexibility in customizing URLs for specific controllers and actions

**14. What is the purpose of generating URLs dynamically at runtime in ASP.NET Core?**
a) To hardcode URLs for better performance
b) To ensure URLs remain static and unchanged even if pages are renamed
c) To dynamically adjust URLs based on changes in the application's routing infrastructure
d) To manually manage links within the application

Answer: c) To dynamically adjust URLs based on changes in the application's routing infrastructure

**15. Why is manually managing hardcoded URLs within an application not recommended?**

a) It ensures better performance and faster response times
b) It eliminates the need for the routing infrastructure
c) It can lead to broken links and 404 errors when pages are renamed or URLs change
d) It simplifies the development process by avoiding URL generation

Answer: c) It can lead to broken links and 404 errors when pages are renamed or URLs change

**16. What happens to URLs when a Razor Page is renamed in ASP.NET Core?**
a) They remain unchanged
b) They are automatically updated to reflect the new page name
c) They become broken and inaccessible
d) They need to be manually updated in the application code

Answer: b) They are automatically updated to reflect the new page name

**17. How does URL generation in ASP.NET Core relate to the routing process?**
a) URL generation reverses the routing process by matching route templates to URLs
b) URL generation is unrelated to the routing process
c) URL generation and routing process are identical
d) URL generation dynamically adjusts route templates based on URLs

Answer: a) URL generation reverses the routing process by matching route templates to URLs

**18. What is the benefit of using URL generation in ASP.NET Core?**
a) It simplifies the routing process
b) It ensures URLs remain hardcoded and unchanged
c) It frees developers from manually managing links and avoids broken URLs
d) It improves application performance by caching URLs

Answer: c) It frees developers from manually managing links and avoids broken URLs

# *CHAPTER 6*
# *The binding model: Retrieving and validating user input*

**1. What is the purpose of model binding in ASP.NET Core?**
a) To create HTTP requests from .NET objects
b) To extract values from a request and create .NET objects
c) To serialize .NET objects into JSON responses
d) To validate user input in Razor Pages

Answer: b) To extract values from a request and create .NET objects

**2. How are objects created for method parameters in page handlers or properties in PageModel in Razor Pages?**
a) By directly instantiating them in the code
b) Through two-way data binding
c) Using model binding
d) By manually parsing the request body

Answer: c) Using model binding

**3. What does the model binder do in ASP.NET Core?**
a) Validates user input before creating .NET objects
b) Converts .NET objects into strings for HTTP requests
c) Extracts values from a request and assigns them to .NET objects
d) Generates random data to populate .NET objects

Answer: c) Extracts values from a request and assigns them to .NET objects

**4. Which attribute is used to mark properties in PageModel for model binding?**
a) [BindData]
b) [ModelProperty]
c) [BindObject]
d) [BindProperty]

Answer: d) [BindProperty]

**5. Is model binding in ASP.NET Core Razor Pages a one-way or two-way process?**
a) One-way
b) Two-way
c) Bidirectional
d) Non-directional

Answer: a) One-way

**6. What is the purpose of using a nested class like InputModel for model binding in ASP.NET Core Razor Pages?**
a) To simplify the routing configuration
b) To handle multiple model bindings in a single class
c) To optimize the performance of Razor Pages
d) To enable two-way data binding in Razor Pages

Answer: b) To handle multiple model bindings in a single class

**7. Which attribute is typically used to mark properties for model binding in ASP.NET Core Razor Pages?**
a) [BindObject]
b) [ModelProperty]

c) [BindProperty]
d) [BindModel]

Answer: c) [BindProperty]

**8. What advantage does using a nested class like InputModel offer in ASP.NET Core Razor Pages?**
a) It allows for bidirectional data binding
b) It simplifies routing configuration
c) It centralizes all model bindings in one class
d) It improves the security of Razor Pages

Answer: c) It centralizes all model bindings in one class

**9. In the provided example, where does the product model reside in the Razor Page's structure?**
a) It is directly within the IndexModel class
b) It is within the InputModel class
c) It is within the OnGet() method
d) It is outside the IndexModel class

Answer: b) It is within the InputModel class

**10. What is the advantage of binding complex types in ASP.NET Core Razor Pages?**
a) It simplifies the routing configuration
b) It improves the security of the application
c) It reduces the clutter in method signatures
d) It enhances the performance of the Razor Pages

Answer: c) It reduces the clutter in method signatures

**11. Why might using multiple method parameters for data binding become cumbersome?**
a) It increases the risk of data loss during binding
b) It makes the code less readable and maintainable
c) It improves the flexibility of the application
d) It simplifies the debugging process

Answer: b) It makes the code less readable and maintainable

**12. What approach is suggested for handling many method parameters in ASP.NET Core Razor Pages?**
a) Using attribute-based binding
b) Using routing templates
c) Binding to complex objects
d) Using global convention-based routing

Answer: c) Binding to complex objects

**13. How does binding to complex objects help in managing changing requirements?**
a) It allows for direct conversion of string parameters
b) It simplifies the process of adding new parameters
c) It restricts the types of parameters that can be used
d) It increases the complexity of the method signature

Answer: b) It simplifies the process of adding new parameters

**14. What is the purpose of DataAnnotations attributes in ASP.NET Core Razor Pages?**
a) They provide data for model binding
b) They control the routing configuration
c) They specify the rules for model validation
d) They handle exception handling in the application

Answer: c) They specify the rules for model validation

**15. How do DataAnnotations attributes contribute to model validation?**
a) By directly validating the user input
b) By providing metadata about the binding model
c) By mapping URLs to page handlers
d) By handling routing logic

Answer: a) By directly validating the user input

**16. What can be achieved by applying DataAnnotations attributes to binding models?**
a) Data encryption
b) User authentication
c) Data validation rules
d) Database queries

Answer: c) Data validation rules

**17. In which scenario might DataAnnotations attributes be useful?**
a) Managing middleware pipelines
b) Configuring routing templates
c) Validating user input on a form
d) Handling HTTP response status codes

Answer: c) Validating user input on a form

# ***CHAPTER 7***
# *Rendering HTML using Razor views*

**1. What is the recommended approach for generating HTML responses in ASP.NET Core?**
a) Generating HTML directly in page handlers
b) Using middleware to generate HTML responses

c) Utilizing Razor views to generate HTML
d) Sending raw HTML strings as responses

Answer: c) Utilizing Razor views to generate HTML

**2. Why is it discouraged to directly generate HTML strings in page handlers?**
a) It leads to better separation of concerns
b) It simplifies the development process
c) It improves performance
d) It provides more control over HTML rendering

Answer: a) It leads to better separation of concerns

**3. What advantage do Razor views offer over directly generating HTML in page handlers?**
a) They provide access to a wider variety of features
b) They offer faster rendering performance
c) They allow for easier debugging
d) They reduce the amount of code needed

Answer: a) They provide access to a wider variety of features

**4. When might it be acceptable to generate HTML responses without using a view?**
a) When building complex web applications
b) When generating static content
c) When working with dynamic data
d) When handling form submissions

Answer: b) When generating static content

**5. What is the recommended approach for passing data from a page handler to its associated Razor view in ASP.NET Core?**
a) HttpContext
b) ViewData
c) @inject services
d) PageModel properties

Answer: d) PageModel properties

**6. When might you consider using ViewData to pass data to a Razor view?**
a) When you need to access services in the view
b) When you need to pass data to _layout files
c) When you want to expose data as properties on the PageModel
d) When you want to utilize dependency injection

Answer: b) When you need to pass data to _layout files

**7. Why is using HttpContext to pass data between a page handler and a Razor view discouraged?**

a) It leads to better separation of concerns
b) It provides limited flexibility in data passing
c) It may cause performance issues
d) It is not supported in Razor views

Answer: b) It provides limited flexibility in data passing

**8. What is the purpose of using @inject services in a Razor view?**
a) To access HttpContext properties
b) To pass data between the page handler and the view
c) To expose data as properties on the PageModel
d) To make services available in the view through dependency injection

Answer: d) To make services available in the view through dependency injection

**9. What is the purpose of the _ViewStart.cshtml file in ASP.NET Core?**
a) It contains directives that are inserted at the top of every view.
b) It defines child actions for rendering discrete sections of a layout.
c) It replaces the need for adding layout code redundantly in every view.
d) It is used to invoke controller actions from within a view.

Answer: c) It replaces the need for adding layout code redundantly in every view.

**11. In ASP.NET Core, what has replaced child actions for rendering complex layout sections?**
a) Partial views
b) View models
c) View components
d) Layout components

Answer: c) View components

**12. What does the _ViewImports.cshtml file contain in ASP.NET Core?**
a) Layout definitions for every view
b) Code for rendering complex layout sections
c) Directives inserted at the top of every view
d) Child actions invoked from within views

Answer: c) Directives inserted at the top of every view

**13. How does using _ViewImports.cshtml help in managing common tasks in Razor views?**
a) It defines child actions for rendering complex layout sections.
b) It provides a mechanism to insert layout code in every view.
c) It includes directives that need to be added to every view, reducing redundancy.
d) It replaces the need for view components in rendering HTML.

Answer: c) It includes directives that need to be added to every view, reducing redundancy.

# *CHAPTER 8*
# *Building forms with Tag Helpers*

**1. What is the primary purpose of Tag Helpers in ASP.NET Core?**
a) To modify the behavior of HTML elements on the client-side.
b) To replace standard HTML elements with new ones entirely.
c) To easily integrate server-side values with generated HTML.
d) To enforce strict validation rules on form inputs.

Answer: c) To easily integrate server-side values with generated HTML.

**2. How do Tag Helpers simplify the process of creating HTML forms in ASP.NET Core?**
a) By automatically generating JavaScript code for client-side validation.
b) By allowing the use of new HTML elements not supported by standard HTML.
c) By dynamically populating values from the PageModel and setting correct attributes.
d) By enabling direct communication between the client-side and server-side components.

Answer: c) By dynamically populating values from the PageModel and setting correct attributes.

**3. What role do Tag Helpers play in handling form submission in ASP.NET Core?**
a) They enforce strict validation rules on form inputs before submission.
b) They automatically populate input values based on the PageModel properties.
c) They generate unique identifiers for each form element to ensure proper binding.
d) They transform standard HTML elements into custom elements for better performance.

Answer: b) They automatically populate input values based on the PageModel properties.

**4. Which aspect of form validation do Tag Helpers assist with?**
a) Client-side validation logic
b) Generation of CAPTCHA challenges
c) Rendering error messages for invalid inputs
d) Encrypting form data before transmission

Answer: c) Rendering error messages for invalid inputs

**5. What is the primary purpose of Tag Helpers when creating forms in ASP.NET Core?**
a) To validate user inputs on the client-side.
b) To generate HTML markup based on PageModel properties.
c) To execute server-side logic for form submissions.
d) To define routing rules for form actions.

Answer: b) To generate HTML markup based on PageModel properties.

**6. How do Tag Helpers assist in reducing manual markup in form creation?**
a) By providing server-side validation for form inputs.

b) By automatically generating JavaScript code for form handling.
c) By dynamically setting attributes like id, name, and value based on PageModel properties.
d) By enforcing strict security measures for form submissions.

Answer: c) By dynamically setting attributes like id, name, and value based on PageModel properties.

**7. In what scenario would Tag Helpers be particularly beneficial for form creation?**
a) When implementing complex client-side animations for form elements.
b) When integrating forms with external APIs for data retrieval.
c) When designing forms with a large number of input fields.
d) When implementing server-side validation logic for form submissions.

Answer: c) When designing forms with a large number of input fields.

**8. What is the primary purpose of the Label Tag Helper in ASP.NET Core?**
a) To validate user inputs on form fields.
b) To automatically generate <label> elements for form fields.
c) To enforce authorization policies for form submissions.
d) To handle server-side logic for form label generation.

Answer: b) To automatically generate <label> elements for form fields.

**9. How does the Label Tag Helper determine the caption to display for a property?**
a) By extracting the caption from the input field's placeholder attribute.
b) By using the property name as the default caption if no [Display] attribute is present.
c) By retrieving the caption from the [Label] attribute applied to the property.
d) By dynamically generating a caption based on the PageModel's structure.

Answer: b) By using the property name as the default caption if no [Display] attribute is present.

**10. Which attribute is utilized by the Label Tag Helper to determine the appropriate value to display for a property?**
a) [Caption]
b) [Name]
c) [Display]
d) [Value]

Answer: c) [Display]

**11. What is the primary purpose of Tag Helpers in ASP.NET Core?**
a) To handle client-side validation of HTML forms.
b) To directly integrate server-side values with HTML elements.
c) To generate dynamic HTML using frontend frameworks like Angular or React.
d) To manage user authentication and authorization.

Answer: b) To directly integrate server-side values with HTML elements.

**12. Which of the following statements about Tag Helpers is true?**
a) Tag Helpers can only be used in frontend frameworks like Angular or React.
b) Tag Helpers are primarily used for server-side validation of HTML forms.
c) Tag Helpers attach to existing HTML elements using attributes to customize them.
d) Tag Helpers are standalone HTML elements that cannot modify existing elements.

Answer: c) Tag Helpers attach to existing HTML elements using attributes to customize them.

**13. How can you set the action URL of a <form> element using the Form Tag Helper?**
a) Using the asp-action attribute.
b) Using the asp-route attribute.
c) Using the asp-page and asp-page-handler attributes.
d) Using the asp-validation-for attribute.

Answer: c) Using the asp-page and asp-page-handler attributes.

**14. What is the purpose of the Label Tag Helper?**
a) To generate dropdown <select> elements.
b) To display validation error messages for form fields.
c) To generate captions and for attributes for <label> elements.
d) To conditionally render HTML based on the app's execution environment.

Answer: c) To generate captions and for attributes for <label> elements.

**15. How does the Input Tag Helper reduce the amount of HTML code needed for form fields?**
a) By automatically generating JavaScript code for client-side validation.
b) By dynamically generating input types and validation attributes based on model properties.
c) By providing built-in CSS styles for form elements.
d) By integrating with frontend frameworks like Angular or React.

Answer: b) By dynamically generating input types and validation attributes based on model properties.

**16. What is required to enable client-side validation with Tag Helpers?**
a) Adding the [Validate] attribute to model properties.
b) Including necessary JavaScript files for jQuery validation and unobtrusive validation.
c) Applying the [ClientValidation] attribute to Razor views.
d) Enabling validation middleware in the Startup class.

Answer: b) Including necessary JavaScript files for jQuery validation and unobtrusive validation.

**17. How can you generate a <select> element for an IEnumerable property in the PageModel?**
a) Use the @Html.DropDownListFor helper method.

b) Use the SelectListItemFor Tag Helper.
c) Use the @Html.GetEnumSelectList<TEnum>() helper method.
d) Use the Select Tag Helper with the asp-items attribute.

Answer: d) Use the Select Tag Helper with the asp-items attribute.

**18. What does the Validation Summary Tag Helper display?**
a) Individual validation error messages for each form field.
b) Model-level validation errors for the entire form.
c) Client-side JavaScript validation errors.
d) Server-side validation errors for API requests.

Answer: b) Model-level validation errors for the entire form.

**19. How can you generate <a> URLs using the Anchor Tag Helper?**
a) By using the asp-href attribute.
b) By using the asp-route attribute.
c) By using the asp-action attribute.
d) By using the asp-append-version attribute.

Answer: b) By using the asp-route attribute.

**20. What is the purpose of the Environment Tag Helper?**
a) To handle caching of static assets like images and scripts.
b) To conditionally render HTML based on the app's current execution environment.
c) To display validation error messages for form fields.
d) To generate URLs for API endpoints.

Answer: b) To conditionally render HTML based on the app's current execution environment.

# ***CHAPTER 9***

# *Creating a Web API for mobile and client applications using MVC*

**1. What is a Web API?**
a) A framework for building server-side web applications.
b) A programming language for client-side scripting.
c) A set of rules and protocols for exchanging data between web applications.
d) An interface for controlling web browsers.

Answer: c) A set of rules and protocols for exchanging data between web applications.

**2. When should you use a Web API?**
a) When you need to generate HTML with Razor templates.
b) When you want to build a single-page application (SPA) using client-side frameworks.

c) When you need to handle requests by returning HTML to the user.
d) When you need to exchange data between different web applications or services.

Answer: d) When you need to exchange data between different web applications or services.

**3. What distinguishes a Web API from traditional web applications?**
a) Web APIs use client-side frameworks like Angular, React, and Vue.
b) Web APIs return HTML directly to the user's web browser.
c) Web APIs rely on server-side rendering of HTML with Razor templates.
d) Web APIs exchange data using standardized protocols and formats, rather than returning HTML.

Answer: d) Web APIs exchange data using standardized protocols and formats, rather than returning HTML.

**4. What was the purpose of the ASP.NET Web API framework?**
a) To generate HTML for web applications.
b) To create HTTP endpoints for returning formatted JSON or XML.
c) To build single-page applications using client-side frameworks.
d) To handle routing and request processing for MVC applications.

Answer: b) To create HTTP endpoints for returning formatted JSON or XML.

**5. How did the ASP.NET Web API framework relate to the MVC framework?**
a) They were completely separate and couldn't interoperate.
b) They shared the same underlying web stack and could seamlessly integrate.
c) Web API was a subset of the MVC framework.
d) MVC framework was deprecated in favor of the Web API framework.

Answer: a) They were completely separate and couldn't interoperate.

**6. What did the ASP.NET Web API framework primarily return in response to requests?**
a) HTML pages.
b) JavaScript files.
c) Formatted JSON or XML.
d) CSS stylesheets.

Answer: c) Formatted JSON or XML.

**7. What was the purpose of the ASP.NET Web API framework?**
a) To generate HTML for web applications.
b) To create HTTP endpoints for returning formatted JSON or XML.
c) To build single-page applications using client-side frameworks.
d) To handle routing and request processing for MVC applications.

Answer: b) To create HTTP endpoints for returning formatted JSON or XML.

**8. How did the ASP.NET Web API framework relate to the MVC framework?**
a) They were completely separate and couldn't interoperate.
b) They shared the same underlying web stack and could seamlessly integrate.
c) Web API was a subset of the MVC framework.
d) MVC framework was deprecated in favor of the Web API framework.

Answer: a) They were completely separate and couldn't interoperate.

**9. What did the ASP.NET Web API framework primarily return in response to requests?**
a) HTML pages.
b) JavaScript files.
c) Formatted JSON or XML.
d) CSS stylesheets.

Answer: c) Formatted JSON or XML.

**10. What is attribute routing used for in ASP.NET Core?**
a) Generating HTML views for web applications.
b) Associating API controller actions with specific route templates.
c) Defining routing middleware in the middleware pipeline.
d) Mapping incoming requests to Razor Pages.

Answer: b) Associating API controller actions with specific route templates.

**11. How do you specify route templates for API controller actions with attribute routing?**
a) Using the @page directive.
b) Using conventional routing.
c) By decorating each action method with an attribute.
d) By specifying routing middleware in the pipeline.

Answer: c) By decorating each action method with an attribute.

**12. What is the alternative to attribute routing for API controllers?**
a) Conventional routing.
b) Razor Pages routing.
c) Middleware routing.
d) Dependency injection routing.

Answer: a) Conventional routing.
**13. What is the purpose of the [ApiController] attribute in ASP.NET Core?**
a) It defines route templates for API controllers.
b) It specifies middleware for routing requests.
c) It reduces the amount of code needed to create consistent Web API controllers.
d) It configures dependency injection for API controller actions.

Answer: c) It reduces the amount of code needed to create consistent Web API controllers.

**14. When was the [ApiController] attribute introduced in .NET Core?**

a) .NET Core 1.0
b) .NET Core 2.0
c) .NET Core 2.1
d) .NET Core 3.0

Answer: c) .NET Core 2.1

**15. What does the [ApiController] attribute do for Web API controllers?**
a) It adds authentication to API endpoints.
b) It enforces HTTPS for API requests.
c) It automatically applies common conventions, such as model validation and response formatting.
d) It generates HTML views for API responses.

Answer: c) It automatically applies common conventions, such as model validation and response formatting.

**16. What is content negotiation in the context of ASP.NET Core?**
a) It refers to negotiating the terms of data transmission between the server and the client.
b) It involves negotiating the pricing of content served by the server.
c) It refers to negotiating the layout of web pages with CSS.
d) It involves negotiating access rights to content based on user authentication.

Answer: a) It refers to negotiating the terms of data transmission between the server and the client.

**17. What HTTP status code does ASP.NET Core return by default when a null API model is returned from an action method?**
a) 200 OK
b) 204 No Content
c) 400 Bad Request
d) 404 Not Found

Answer: b) 204 No Content

**18. How does ASP.NET Core format the response when a string is returned as the API model and no Accept header is set?**
a) JSON
b) XML
c) Plain text
d) HTML

Answer: c) Plain text

# *CHAPTER 10*
# *Service configuration with dependency injection*

**1. What is dependency injection (DI) commonly referred to as in ASP.NET Core?**
a) Dependency Inversion
b) Dependency Management
c) Dependency Resolution
d) Dependency Isolation

Answer: a) Dependency Inversion

**2. Which software engineering principles does ASP.NET Core adhere to, making DI a core concept?**
a) SOLID
b) Agile
c) Waterfall
d) RAD

Answer: a) SOLID

**3. What is another term often used interchangeably with dependency injection?**
a) Dependency Association
b) Dependency Provisioning
c) Inversion of Control (IoC)
d) Dependency Aggregation

Answer: c) Inversion of Control (IoC)

**4. Which book is recommended for a more in-depth understanding of dependency injection in C#?**
a) "The Pragmatic Programmer" by Andrew Hunt and David Thomas
b) "Clean Code" by Robert C. Martin
c) "Dependency Injection Principles, Practices, and Patterns" by Steven van Deursen and Mark Seemann
d) "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

Answer: c) "Dependency Injection Principles, Practices, and Patterns" by Steven van Deursen and Mark Seemann

**5. What benefit does dependency injection provide in terms of managing class dependencies?**
a) It eliminates the need for class dependencies altogether.
b) It centralizes the management of class dependencies.
c) It automatically resolves class dependencies at runtime.
d) It increases the number of class dependencies in the application.

Answer: b) It centralizes the management of class dependencies.

**6. Which software engineering principle emphasizes the importance of decoupling dependencies between classes?**
a) DRY (Don't Repeat Yourself)
b) YAGNI (You Ain't Gonna Need It)
c) KISS (Keep It Simple, Stupid)
d) DIP (Dependency Inversion Principle)

Answer: d) DIP (Dependency Inversion Principle)

**7. What benefits does dependency injection provide beyond managing class dependencies?**
a) Increased code complexity
b) Improved testability and maintainability
c) Decreased modularity
d) Reduced scalability

Answer: b) Improved testability and maintainability

**8. In which part of the ASP.NET Core framework is dependency injection integrated?**
a) At the presentation layer
b) Within the data access layer
c) At the heart of the framework
d) Within the security layer

Answer: c) At the heart of the framework

**9. What is the primary purpose of dependency injection in ASP.NET Core?**
a) To increase the complexity of the application
b) To minimize modularity and flexibility
c) To improve the performance of the application
d) To promote loose coupling and enhance maintainability

Answer: d) To promote loose coupling and enhance maintainability

**10. Which article is recommended for a deeper understanding of dependency injection by Martin Fowler?**

a) "Refactoring: Improving the Design of Existing Code"
b) "Continuous Integration"
c) "Inversion of Control Containers and the Dependency Injection pattern"
d) "Patterns of Enterprise Application Architecture"

Answer: c) "Inversion of Control Containers and the Dependency Injection pattern"

**11. What does coupling refer to in object-oriented programming?**
a) The relationship between classes and methods
b) The degree of interaction between different classes
c) The size of the codebase
d) The inheritance hierarchy of classes

Answer: b) The degree of interaction between different classes

**12. What is the impact of tight coupling between classes?**
a) Improved testability
b) Reduced flexibility and maintainability
c) Enhanced scalability
d) Simplified codebase

Answer: b) Reduced flexibility and maintainability

**13. In the context of dependency injection, what does it mean for code to be loosely coupled?**
a) The codebase is small and concise
b) Classes depend on specific implementations rather than abstractions
c) Classes do not need to know many details about other components to use them
d) The codebase relies heavily on inheritance

Answer: c) Classes do not need to know many details about other components to use them

**14. How does injecting dependencies via the constructor help reduce coupling?**
a) It increases the complexity of the system
b) It reduces the need for unit testing
c) It eliminates the need for dependencies altogether
d) It removes the responsibility of creating dependencies from the class, making it more flexible and easier to test

Answer: d) It removes the responsibility of creating dependencies from the class, making it more flexible and easier to test

**15. What is the advantage of coding to interfaces in reducing coupling?**
a) It limits the functionality of classes
b) It simplifies the codebase
c) It ties classes to a single implementation
d) It allows for interchangeable implementations, making classes more flexible and testable

Answer: d) It allows for interchangeable implementations, making classes more flexible and testable

**16. What does the "transient" lifetime mean in dependency injection?**
a) The service instance remains the same throughout the application's lifetime
b) A new instance of the service is created for each request or usage
c) The service instance is shared across different scopes
d) The service instance is cached indefinitely

Answer: b) A new instance of the service is created for each request or usage

**17. In ASP.NET Core, which lifetime corresponds to a service being scoped to a web request?**
a) Transient
b) Scoped
c) Singleton
d) Request

Answer: b) Scoped

**18. What happens to a service with a "singleton" lifetime?**
a) A new instance is created for each request
b) The service instance is shared across different scopes
c) A new instance is created for each usage
d) The same instance is returned for every request throughout the application's lifetime

Answer: d) The same instance is returned for every request throughout the application's lifetime

**19. What is configuration in the context of ASP.NET Core applications?**
a) It refers to the internal parameters used by the application to control its behavior.
b) It consists only of settings and does not include secrets.
c) Configuration is the set of external parameters provided to an application that controls its behavior.
d) Configuration is used only during development and not during deployment.

Answer: c) Configuration is the set of external parameters provided to an application that controls its behavior.

**20. What is the purpose of storing settings and secrets outside of compiled code in ASP.NET Core applications?**
a) It simplifies the development process.
b) It improves code organization.
c) It enables easy tweaking of values without recompiling the code.
d) It prevents unauthorized access to sensitive data.

Answer: c) It enables easy tweaking of values without recompiling the code.

**21. Why is it considered a security best practice to externalize secret values like API keys or passwords in ASP.NET Core applications?**
a) Hardcoding secrets into code makes it easier to access them.
b) Externalizing secrets helps in source control management.
c) Embedding secrets in compiled code can make them publicly available.
d) Secrets stored in code are more secure than external storage.

Answer: c) Embedding secrets in compiled code can make them publicly available.

**22. Which of the following are built-in configuration providers in ASP.NET Core?**
a) YAML files
b) JSON files
c) CSV files
d) Markdown files

Answer: b) JSON files

**23. How can you load configuration data from environment variables in ASP.NET Core?**
a) By directly accessing the environment variable from code
b) By specifying the environment variable name in the configuration file
c) By using the built-in configuration provider for environment variables
d) By creating a custom provider specifically for environment variables

Answer: c) By using the built-in configuration provider for environment variables

**24. What is a potential solution if the built-in configuration providers in ASP.NET Core do not meet your requirements?**
a) Writing custom configuration classes
b) Using third-party libraries from GitHub and NuGet
c) Using YAML files instead of JSON or XML
d) Modifying the ASP.NET Core framework

Answer: b) Using third-party libraries from GitHub and NuGet

**25. What is a key advantage of using configuration and settings in ASP.NET Core?**
a) Improved security measures
b) Automatic compilation of the application
c) Ability to edit settings without recompiling or restarting the application
d) Enhanced logging capabilities

Answer: c) Ability to edit settings without recompiling or restarting the application

**26. In ASP.NET Core, what is the benefit of being able to change configuration at runtime?**
a) It reduces the verbosity of log entries
b) It allows for automatic compilation of the application
c) It enables easier debugging by adjusting logging levels
d) It improves security measures

Answer: c) It enables easier debugging by adjusting logging levels

**27. When does ASP.NET Core support reloading of configuration files?**
a) Only for environment variable-based configuration
b) Only for User Secrets provider-based configuration
c) Only for file-based configuration providers
d) For all types of configuration providers

Answer: c) Only for file-based configuration providers

**28. What is the preferred way of accessing configuration in ASP.NET Core?**
a) Using IConfiguration abstraction directly
b) Accessing configuration using string keys
c) Utilizing strongly typed configuration and the Options pattern
d) Employing IConfigurationBuilder for configuration access

Answer: c) Utilizing strongly typed configuration and the Options pattern

**29. What are the advantages of using strongly typed settings in ASP.NET Core?**
a) Reduced verbosity and increased readability
b) Enhanced security measures
c) Simplified testing and avoidance of typos
d) Better performance optimization

Answer: c) Simplified testing and avoidance of typos

**30. How are strongly typed settings typically represented in ASP.NET Core?**
a) As complex JSON objects
b) As IConfiguration instances
c) As simple POCO objects
d) As IConfigurationSection instances

Answer: c) As simple POCO objects

# *CHAPTER 12*

# *Saving data with Entity Framework Core*

**1. What problem does Entity Framework Core (EF Core) aim to solve?**
a) Handling network connections
b) Writing SQL statements
c) Mapping database responses to .NET classes
d) Managing variable result data

Answer: c) Mapping database responses to .NET classes

**2. Which term best describes EF Core's role in database access?**

a) Query builder
b) Object-relational mapper (ORM)
c) Database connector
d) SQL interpreter

Answer: b) Object-relational mapper (ORM)

**3. What distinguishes EF Core from the existing Entity Framework libraries?**
a) It provides cross-platform support only
b) It is based on ADO.NET
c) It offers highly performant database access
d) It focuses solely on NoSQL databases

Answer: c) It offers highly performant database access

**4. Which type of databases can EF Core interact with?**
a) Only relational databases
b) Only NoSQL databases
c) Both relational and NoSQL databases
d) Only in-memory databases

Answer: a) Only relational databases

**5. What feature of EF Core allows for creating temporary databases for testing purposes?**
a) Database providers
b) Entity Framework libraries
c) Object-relational mapping
d) In-memory feature

Answer: d) In-memory feature

**6. What is the primary goal of EF Core?**
a) To handle network connections efficiently
b) To provide cross-platform database access
c) To implement a wide range of NoSQL databases
d) To map database responses to Java classes

Answer: b) To provide cross-platform database access

**7. What type of model does EF Core use for database access?**
a) Structural model
b) Provider model
c) Relational model
d) Object-oriented model

Answer: b) Provider model

**8. What is the benefit of EF Core's provider model?**

a) It limits database access to Microsoft SQL Server only
b) It enables support for various relational databases
c) It restricts the use of SQL queries
d) It eliminates the need for database mapping

Answer: b) It enables support for various relational databases

**9. Which version of .NET Core introduced EF Core?**
a) 1.x
b) 2.x
c) 3.x
d) 5.x

Answer: b) 2.x

**10. What is the primary motivation behind EF Core's development?**
a) To provide low-level database access
b) To promote SQL query building
c) To improve performance and cross-platform support
d) To replace ADO.NET libraries

Answer: c) To improve performance and cross-platform support

# *CHAPTER 13*
# *The MVC and Razor Pages filter pipeline*

**1. What is the primary purpose of the filter pipeline in ASP.NET Core?**
a) To handle network requests
b) To manage database transactions
c) To provide hooks into the request processing lifecycle
d) To parse JSON payloads

Answer: c) To provide hooks into the request processing lifecycle

**2. How do filters differ from middleware in ASP.NET Core?**
a) Filters are executed before middleware
b) Middleware is executed before filters
c) Filters are specific to MVC, while middleware can be used across different ASP.NET Core frameworks
d) Middleware and filters are interchangeable terms

Answer: c) Filters are specific to MVC, while middleware can be used across different ASP.NET Core frameworks

**3. What is an example scenario where filters are commonly used?**
a) Handling database queries
b) Authenticating users
c) Generating HTML templates
d) Parsing request payloads

Answer: b) Authenticating users

**4. How many types of filters are there in ASP.NET Core?**
a) Three
b) Four
c) Five
d) Six

Answer: d) Six

**5. Which of the following is NOT a type of filter in ASP.NET Core?**
a) Action filters
b) Result filters
c) Model filters
d) Exception filters

Answer: c) Model filters

**6. When might you use an action filter?**
a) To modify the response body before sending it to the client
b) To handle exceptions thrown during action execution
c) To perform validation on incoming request data
d) To execute custom logic before or after an action method

Answer: d) To execute custom logic before or after an action method

**7. How can you control the order in which filters are executed?**
a) By setting their priority in the ConfigureServices method
b) By setting their priority in the Configure method
c) By decorating them with the [Order] attribute
d) By configuring them in the Startup class

Answer: c) By decorating them with the [Order] attribute

**8. Which type of filter is primarily used for protecting APIs and action methods by determining if a request is authorized?**
a) Resource filters
b) Action filters
c) Authorization filters
d) Exception filters

Answer: c) Authorization filters

**9. When do resource filters typically execute in the filter pipeline?**
a) After action filters
b) Before authorization filters
c) Before and after action methods
d) At the beginning and end of the pipeline

Answer: d) At the beginning and end of the pipeline

**10. What can action filters manipulate before an action method executes?**
a) The incoming request body
b) The action method's return type
c) The arguments to the method
d) The URL routing parameters

Answer: c) The arguments to the method

**11. When do exception filters come into play in the filter pipeline?**
a) Before authorization filters
b) After action filters
c) After resource filters
d) When an exception occurs in the filter pipeline

Answer: d) When an exception occurs in the filter pipeline

**12. How do result filters differ from action filters?**
a) Result filters only run after an action method's execution
b) Result filters can only manipulate the action result's return type
c) Result filters run both before and after an action method's execution
d) Result filters are primarily used for handling authorization logic

Answer: c) Result filters run both before and after an action method's execution

**13. When do page filters execute in the Razor Pages filter pipeline?**
a) Before model binding
b) After page handler execution
c) Only after model binding and validation
d) Three times during different stages of the pipeline

Answer: d) Three times during different stages of the pipeline

**14. What happens after the first execution of a page filter in the Razor Pages filter pipeline?**
a) Page handler execution
b) Model binding and validation
c) Customization of the page handler's result
d) Short-circuiting the pipeline

Answer: b) Model binding and validation

**15. In the Razor Pages filter pipeline, when does the third execution of a page filter occur?**
a) After page handler selection
b) Before model binding
c) After model binding and validation
d) After page handler execution

Answer: d) After page handler execution

**16. How do page filters differ from action filters in the MVC filter pipeline?**
a) Page filters execute only once in the pipeline
b) Page filters are executed after action filters
c) Page filters have triple execution, whereas action filters execute only once
d) Page filters cannot manipulate model-bound data

Answer: c) Page filters have triple execution, whereas action filters execute only once

**17. Which interface should you implement to create an authorization filter in ASP.NET Core?**
a) IAuthorizationFilter
b) IAsyncAuthorizationFilter
c) IFilter
d) IAuthorizationAsyncFilter

Answer: a) IAuthorizationFilter

**18. What interface should you implement to develop a resource filter asynchronously?**
a) IAsyncFilter
b) IResourceFilter
c) IAsyncResultFilter
d) IAsyncResourceFilter

Answer: d) IAsyncResourceFilter

**19. Which interface is suitable for creating action filters that execute synchronously?**
a) IFilter
b) IActionFilter
c) IAsyncFilter
d) IAsyncActionFilter

Answer: b) IActionFilter

**20. If you need to develop an exception filter that works asynchronously, which interface should you implement?**
a) IAsyncExceptionFilter
b) IExceptionFilter
c) IFilter
d) IAsyncResultFilter

Answer: a) IAsyncExceptionFilter

**21. What is the primary purpose of creating custom filters in ASP.NET Core?**
a) To replace built-in filters with custom implementations
b) To enhance the security features of the application
c) To handle specific requirements unique to the application
d) To reduce the overall complexity of the application

Answer: c) To handle specific requirements unique to the application

**22. When should you consider creating custom filters?**
a) When built-in filters cover all the required functionalities
b) When there is a need to streamline the application's codebase
c) When certain functionalities cannot be achieved using built-in filters
d) When performance optimization is the primary concern

Answer: c) When certain functionalities cannot be achieved using built-in filters

**23. Which section of the application's architecture often benefits the most from the implementation of custom filters?**
a) Data access layer
b) Presentation layer
c) Business logic layer
d) Service layer

Answer: b) Presentation layer

**24. In what scenario might you create a custom filter for an ASP.NET Core Web API?**
a) When handling user authentication and authorization
b) When implementing cross-cutting concerns such as logging
c) When defining specific data validation rules for incoming requests
d) When optimizing database queries for improved performance

Answer: c) When defining specific data validation rules for incoming requests

# *CHAPTER 14*
# *Authentication: Adding users to your application with Identity*

**1. What are the two primary aspects to consider when adding users to a web application?**
a) Authentication and authorization

b) Encryption and decryption
c) Frontend and backend development
d) Database management and optimization

Answer: a) Authentication and authorization

**2. What is authentication in the context of web applications?**
a) The process of securing data transmission between client and server
b) Customizing the user experience based on their interactions with the application
c) The process of creating users and allowing them to log in to the application
d) Managing user permissions and access rights within the application

Answer: c) The process of creating users and allowing them to log in to the application

**3. What is the primary purpose of ASP.NET Core Identity?**
a) Handling user input validation
b) Providing services for creating and managing users, and authentication functionality
c) Optimizing database queries for improved performance
d) Implementing frontend user interfaces for registration and login

Answer: b) Providing services for creating and managing users, and authentication functionality

**4. What does ASP.NET Core Identity integrate with to provide user management functionalities?**
a) Entity Framework Core
b) Angular framework
c) React library
d) MongoDB database

Answer: a) Entity Framework Core

**5. What is the process of customizing the logic associated with a Razor Page in ASP.NET Core Identity called?**
a) Templating
b) Scaffolding
c) Authorization
d) Customization

Answer: b) Scaffolding

**6. What is the benefit of customizing Razor templates in ASP.NET Core Identity?**
a) Improved database performance
b) Enhanced user authentication
c) Customizing the appearance and functionality of authentication pages
d) Optimizing server response times

Answer: c) Customizing the appearance and functionality of authentication pages

**7. What additional information about a user can be stored using ASP.NET Core Identity?**
a) Email address only
b) Name and date of birth
c) Password and username
d) Social security number

Answer: b) Name and date of birth

**8. What should developers consider when designing apps to provide authentication functionality?**
a) User interface design only
b) Cross-platform compatibility
c) Security, user experience, and scalability
d) Backend database architecture

Answer: c) Security, user experience, and scalability

**9. What is authentication in the context of security?**
a) Determining user permissions
b) Verifying user identities
c) Controlling access to certain functions
d) Encrypting data transmission

Answer: b) Verifying user identities

**10. What does authorization involve?**
a) Determining user identities
b) Verifying user permissions
c) Securing data transmission
d) Encrypting user passwords

Answer: b) Verifying user permissions

**11. Which aspect of security comes first, authentication, or authorization?**
a) Authorization
b) Authentication
c) They occur simultaneously
d) Depends on the application's configuration

Answer: b) Authentication

**12. In traditional web apps, what typically happens after authentication?**
a) The user is redirected to the homepage
b) The user's session is terminated
c) The user's permissions are verified
d) The user's identity is remembered for subsequent requests

Answer: d) The user's identity is remembered for subsequent requests

**13. What is discussed in chapter 15 of the book?**
a) Authentication
b) Authorization
c) Secure data transmission
d) Web application architecture

Answer: b) Authorization

**14. What is the first step in the authentication process for web apps in ASP.NET Core?**
a) Setting the user principal
b) Sending the user's identifier and secret
c) Verifying the user's permissions
d) Encrypting user data

Answer: b) Sending the user's identifier and secret

**15. How does ASP.NET Core verify the user's credentials?**
a) By encrypting the user's identifier
b) By comparing the secret to a stored hash
c) By requesting additional user information
d) By generating a random token

Answer: b) By comparing the secret to a stored hash

**16. What does the app typically use to store user details for subsequent requests in traditional web apps?**
a) Session variables
b) Local storage
c) Encrypted cookies
d) Server-side caching

Answer: c) Encrypted cookies

**17. In ASP.NET Core, where is the user's principal set after successful authentication?**
a) In the session state
b) In a database
c) In an encrypted token
d) In the HTTP context

Answer: d) In the HTTP context

**18. What is ASP.NET Core Identity primarily used for?**
a) Managing user interface components
b) Handling database migrations
c) Storing and managing user details

d) Optimizing server performance

Answer: c) Storing and managing user details

**19. Why is it recommended not to build your own authentication and membership system?**
a) It is more cost-effective to use third-party solutions
b) Third-party solutions offer better security measures
c) Building a secure system requires expertise and effort
d) Third-party solutions are more flexible and customizable

Answer: c) Building a secure system requires expertise and effort

**20. What is a benefit of using ASP.NET Core Identity with EF Core?**
a) It automatically configures cloud authentication
b) It simplifies integration with third-party identity providers
c) It seamlessly integrates with existing EF Core projects
d) It provides built-in support for multi-factor authentication

Answer: c) It seamlessly integrates with existing EF Core projects

**21. What does ASP.NET Core Identity use by default to store user details in the database?**
a) MongoDB
b) SQLite
c) Entity Framework Core
d) PostgreSQL

Answer: c) Entity Framework Core

**22. What are some features provided by ASP.NET Core Identity?**
a) Role-based authentication
b) Two-factor authentication
c) User session management
d) Real-time data synchronization

Answer: a) Role-based authentication, b) Two-factor authentication

**23. What is an alternative to ASP.NET Core Identity for managing user authentication?**
a) Azure Active Directory
b) Amazon Cognito
c) IdentityServer
d) Auth0

Answer: c) IdentityServer

**24. What does ASP.NET Core Identity NOT provide?**
a) Role-based authorization
b) User registration
c) Password hashing

d) Session management

Answer: d) Session management

**25. What is the relationship between ASP.NET Identity and ASP.NET Core Identity?**
a) They are interchangeable and can be used together
b) ASP.NET Core Identity is an improved version of ASP.NET Identity
c) ASP.NET Core Identity is deprecated in favor of ASP.NET Identity
d) They serve different purposes and cannot be compared

Answer: b) ASP.NET Core Identity is an improved version of ASP.NET Identity

**26. What is the benefit of using ASP.NET Core Identity with EF Core for existing projects?**
a) It provides automatic data migration
b) It eliminates the need for database setup
c) It simplifies user interface development
d) It seamlessly integrates with existing database structures

Answer: d) It seamlessly integrates with existing database structures

**27. How does ASP.NET Core Identity contribute to security?**
a) By automatically generating encryption keys
b) By implementing industry-standard password hashing algorithms
c) By restricting access to certain database tables
d) By encrypting all user data stored in the database

Answer: b) By implementing industry-standard password hashing algorithms

# *CHAPTER 15*
# *Authorization: Securing your application*

**1. What is the primary difference between authentication and authorization?**
a) Authentication verifies the identity of a user, while authorization determines what actions they can perform.
b) Authentication determines what actions a user can perform, while authorization verifies their identity.
c) Authentication and authorization are two terms for the same process.
d) Authentication and authorization are unrelated concepts in web security.

Answer: a) Authentication verifies the identity of a user, while authorization determines what actions they can perform.

**2. Which step in the airport scenario best represents authentication?**
a) Showing the boarding pass to enter security

b) Showing the passport at the check-in desk
c) Showing the frequent flyer card to enter the airline lounge
d) Showing the boarding pass to board the flight

Answer: b) Showing the passport at the check-in desk

**3. What does a boarding pass represent in the airport scenario?**
a) Authentication
b) Authorization
c) Claim
d) Role

Answer: c) Claim

**4. In the airport scenario, what happens if a passenger doesn't have a valid BoardingPassNumber?**
a) They are directed to the airline lounge.
b) They are allowed to proceed to security.
c) They are directed back to the check-in desk for authentication and ticket purchase.
d) They are allowed to board the flight.

Answer: c) They are directed back to the check-in desk for authentication and ticket purchase.

**5. Which term is used to describe a piece of information about a user in security contexts?**
a) Password
b) Identifier
c) Claim
d) Token

Answer: c) Claim

**6. What is the main focus of this chapter in terms of security?**
a) Authentication
b) Authorization
c) Encryption
d) Firewall configuration

Answer: b) Authorization

**7. Which step in the airport scenario best represents authorization?**
a) Showing the passport at the check-in desk
b) Showing the boarding pass to enter security
c) Showing the frequent flyer card to enter the airline lounge
d) Showing the boarding pass to board the flight

Answer: b) Showing the boarding pass to enter security

**8. What is the purpose of the BoardingPassNumber in the airport scenario?**
a) It represents the user's identity.
b) It verifies the user's authentication.
c) It serves as an additional claim associated with the user's identity.
d) It determines the user's authorization level.

Answer: c) It serves as an additional claim associated with the user's identity.

**9. Which aspect of web security deals with determining what actions a user can perform?**
a) Authentication
b) Authorization
c) Encryption
d) Session management

Answer: b) Authorization

**10. What analogy does the author use to explain authentication and authorization?**
a) Grocery shopping
b) Traffic rules
c) Airport check-in process
d) Classroom attendance

Answer: c) Airport check-in process

**11. What is the main purpose of authorization policies in ASP.NET Core?**
a) To define the authentication requirements for accessing endpoints
b) To customize the user experience based on their claims
c) To encapsulate the rules for determining if a user is authorized to access a resource
d) To manage user sessions and track their activities

Answer: c) To encapsulate the rules for determining if a user is authorized to access a resource

**12. How are authorization policies applied to actions in ASP.NET Core?**
a) Using the [Authorize] attribute with a specific policy name
b) Using the [Authorize] attribute with inline policy definitions
c) By configuring policies in the ConfigureServices method
d) By directly manipulating the ClaimsPrincipal object

Answer: a) Using the [Authorize] attribute with a specific policy name

**13. What does a ClaimsPrincipal object represent in ASP.NET Core authentication?**
a) A user's session data
b) A collection of user claims
c) The user's password
d) The user's role in the application

Answer: b) A collection of user claims

**14. Which of the following is an example of claims-based authorization?**
a) Requiring users to enter a username and password
b) Allowing only users with admin roles to access certain pages
c) Allowing only users with a specific claim to access a resource
d) Allowing anonymous access to all endpoints

Answer: c) Allowing only users with a specific claim to access a resource

**15. What is the purpose of the [Authorize] attribute in ASP.NET Core?**
a) To define authentication requirements
b) To specify the HTTP method for an action
c) To declare authorization policies
d) To indicate that a user is authorized to access a resource

Answer: d) To indicate that a user is authorized to access a resource