**1. What is the difference between synchronous and asynchronous programming in .NET?**

**Answer: Synchronous code blocks execution until the task completes. Asynchronous code uses async and await keywords, allowing non-blocking operations which improve scalability, especially for I/O-bound tasks.**

**2. How do you implement logging in ASP.NET Core?**

**Answer: ASP.NET Core uses built-in ILogger<T> interface. We configure providers like Console, Debug, Application Insights, or Serilog in Program.cs or appsettings.json.**

**3. What is Kestrel in ASP.NET Core?**

**Answer: Kestrel is a cross-platform web server built into ASP.NET Core. It is lightweight and high-performance, used as the default web server in ASP.NET Core applications.**

**4. What is the purpose of app.UseRouting() and app.UseEndpoints()?**

**Answer:**

* **UseRouting() enables route matching.**
* **UseEndpoints() executes the matched endpoint. They must be used in the correct order in the middleware pipeline to route requests properly.**

**5. What are Value Types and Reference Types in C#?**

**Answer:**

* **Value Types store data directly (e.g., int, bool, structs)**
* **Reference Types store a reference to the data's memory location (e.g., class, string, array)**

**6. How does IQueryable differ from IEnumerable?**

**Answer:**

* **IEnumerable executes queries immediately and processes data in memory.**
* **IQueryable builds expression trees and executes queries on the data source, supporting deferred execution and server-side filtering.**

**7. What is Model Binding in ASP.NET Core?**

**Answer: Model binding maps data from HTTP requests (form, query string, route) to action method parameters or model properties. It simplifies reading request data in controllers.**

**8. What are some common HTTP status codes you use in Web APIs?**

**Answer:**

* **200 OK – Request successful**
* **201 Created – Resource created**
* **400 Bad Request – Validation errors**
* **401 Unauthorized – Authentication failed**
* **404 Not Found – Resource not found**
* **500 Internal Server Error – Unexpected error**

**9. How do you validate a model in ASP.NET Core?**

**Answer: Use Data Annotations (e.g., [Required], [StringLength]) and call ModelState.IsValid in controllers. Custom validators can be created by implementing IValidatableObject or using FluentValidation.**

**10. What is the difference between Put and Patch HTTP methods?**

**Answer:**

* **PUT replaces the entire resource.**
* **PATCH updates partial resource data. Useful for performance when only some fields need updating.**

**11. How do you manage secrets and sensitive settings in development and production?**

**Answer:**

* **Local Dev: Use User Secrets**
* **Production: Store in Azure Key Vault and access via Managed Identity**
* **Never hardcode sensitive data in source code**

**12. How do you scale an application in Azure App Service?**

**Answer:**

* **Vertically: Change service plan to a higher tier**
* **Horizontally: Enable autoscaling based on CPU, memory, or queue length**
* **Use deployment slots for zero-downtime deployments**

**13. What is the role of Startup.cs in ASP.NET Core?**

**Answer: It configures services (ConfigureServices) and the middleware pipeline (Configure). It plays a critical role in defining the application behavior and dependency injection setup.**

**14. How do you test Web APIs?**

**Answer:**

* **Unit Test: Use xUnit/NUnit and Moq**
* **Integration Test: Use TestServer and HttpClient**
* **Postman or Swagger UI for manual testing**

**15. Code Challenge: Check if a number is a palindrome**

**Answer:**

**public bool IsPalindrome(int number)**

**{**

**int original = number, reversed = 0;**

**while (number > 0)**

**{**

**int digit = number % 10;**

**reversed = reversed \* 10 + digit;**

**number /= 10;**

**}**

**return original == reversed;**

**}**