

.NET Core and API Interview Questions & Answers

1. What is .NET Core?

.NET Core is an open-source, cross-platform framework developed by Microsoft for building modern, cloud-based, and high-performance applications. It supports Windows, Linux, and macOS.

2. Difference between .NET Framework and .NET Core?

.NET Framework: Windows-only, older technology, supports WebForms, WPF, etc. .NET Core: Cross-platform, lightweight, modular, better performance, supports microservices.

3. What are the advantages of .NET Core?

Cross-platform support, High performance, Open-source, Modular architecture, Built-in dependency injection.

4. Explain the concept of Middleware in .NET Core.

Middleware is software that processes requests and responses in the HTTP pipeline. Examples: Authentication, Logging, Exception Handling.

5. What is the Startup.cs file used for?

It configures services and the app's request pipeline using ConfigureServices() and Configure() methods.

6. What is Dependency Injection in .NET Core?

A design pattern where dependencies are injected into classes rather than hard-coded. .NET Core has built-in DI using IServiceCollection.

7. What is Kestrel in .NET Core?

Kestrel is a cross-platform web server for ASP.NET Core, optimized for performance.

8. What are the different types of Hosting in .NET Core?

In-process hosting: Runs inside IIS worker process. Out-of-process hosting: Uses Kestrel and communicates with IIS via reverse proxy.

9. What is the difference between AddSingleton, AddScoped, and AddTransient?

Singleton: One instance for the entire application. Scoped: One instance per request. Transient: New instance every time it's requested.

10. What is the role of Program.cs in .NET Core?

It contains the Main() method and sets up the host builder for the application.

11. What is REST API?

REST (Representational State Transfer) is an architectural style for designing networked applications using HTTP methods like GET, POST, PUT, DELETE.

12. Difference between REST and SOAP?

REST: Lightweight, uses JSON/XML, stateless, faster. SOAP: Heavy, uses XML, supports advanced security and transactions.

13. What are HTTP Status Codes commonly used in APIs?

200 OK, 201 Created, 400 Bad Request, 401 Unauthorized, 404 Not Found, 500 Internal Server Error.

14. What is the difference between PUT and PATCH?

PUT: Updates the entire resource. PATCH: Updates part of the resource.

15. How do you secure an API in .NET Core?

Use JWT tokens, Implement Authentication & Authorization, Use HTTPS, Validate inputs.

16. What is Swagger in .NET Core?

Swagger (OpenAPI) is used for API documentation and testing. Implemented via Swashbuckle package.

17. What is Model Binding in ASP.NET Core API?

It maps HTTP request data (query string, route, body) to action method parameters.

18. Explain Action Filters in ASP.NET Core.

Action Filters allow you to run code before or after an action executes (e.g., logging, validation).

19. What is the difference between synchronous and asynchronous API calls?

Synchronous: Blocks the thread until the operation completes. Asynchronous: Uses async/await, non-blocking, better performance under load.

20. How do you handle exceptions in .NET Core API?

Use Exception Middleware, Use try-catch blocks, Implement global error handling with `ExceptionHandler`.