

# Top 50 .NET Full Stack Developer Interview Questions

## Top 50 .NET Full Stack Developer Interview Questions with Answers & Examples

1. What is the difference between .NET and .NET Core?

-> .NET Framework is Windows-only. .NET Core is cross-platform, faster, and used for modern apps.

Example: Use .NET Core to build REST APIs for web and mobile apps.

2. Why do we use the using keyword in C#?

-> For importing namespaces and disposing resources.

Example:

```
using (var file = new StreamReader("file.txt"))  
  
{  
  
    var data = file.ReadToEnd();  
  
}
```

3. What is Dependency Injection? Types?

-> Injects services into classes instead of hardcoding.

Types: Constructor, Property, Method.

Example:

```
public class MyService  
  
{  
  
    private readonly IRepo _repo;  
  
    public MyService(IRepo repo) { _repo = repo; }  
  
}
```

4. What is Middleware in .NET Core?

## Top 50 .NET Full Stack Developer Interview Questions

-> Code that processes requests/responses in the pipeline.

Example:

```
app.Use(async (context, next) => {  
    Console.WriteLine("Before request");  
    await next.Invoke();  
    Console.WriteLine("After response");  
});
```

5. What is a Delegate?

-> Delegate is a reference to a method. Supports callbacks/events.

Example:

```
public delegate void Greet(string name);  
  
Greet greet = name => Console.WriteLine("Hello " + name);  
  
greet("John");
```

6. What is a Function in C#?

-> A block of code that performs a task and may return a value.

Example:

```
int Add(int a, int b) => a + b;
```

7. What are Access Specifiers in C#?

-> Define visibility of members. Types: public, private, protected, internal.

Example:

```
public class Car { private string model; }
```

## Top 50 .NET Full Stack Developer Interview Questions

8. What is a Constructor and its Types?

-> Initializes object. Types: Default, Parameterized, Static, Copy.

Example:

```
public Car(string name) { this.Name = name; }
```

9. Difference: Abstract Class vs Interface

-> Abstract class can have logic. Interface = full abstraction.

Example:

```
interface IRun { void Start(); }
```

10. What is DbContext in EF Core?

-> Manages database connections and operations.

Example:

```
public class AppDbContext : DbContext  
  
{  
  
    public DbSet<Employee> Employees { get; set; }  
  
}
```

11. What is Entity Framework Core?

-> ORM to interact with DB using C# instead of SQL.

12. What is LINQ?

-> Language Integrated Query to query collections/DB.

Example:

```
var result = students.Where(s => s.Age > 18);
```

## Top 50 .NET Full Stack Developer Interview Questions

13. What is DTO and Why Use it?

-> Data Transfer Object: Sends only needed data between layers.

Improves security & performance.

14. What is appsettings.json in .NET Core?

-> Stores configuration settings.

Example:

```
"ConnectionStrings": { "Default": "Server=.;DB=AppDb;" }
```

15. Primary vs Composite vs Unique Key

-> Primary: Not null + unique. Composite: Multiple cols. Unique: Only unique.

16. Four Pillars of OOP?

-> Encapsulation, Abstraction, Inheritance, Polymorphism.

17. Value Type vs Reference Type?

-> Value: Holds data directly (int). Reference: Points to object (class).

18. Angular vs AngularJS?

-> Angular: TypeScript, Components. AngularJS: JavaScript, MVC.

19. What are Components and Modules?

-> Components = UI blocks. Modules = Group of components/services.

## Top 50 .NET Full Stack Developer Interview Questions

20. What is Lazy Loading in Angular?

-> Load modules only when needed.

Example: loadChildren in routes.

21. What is Data Binding in Angular?

-> Synchronizes UI and data.

Types: One-way, Two-way (`[(ngModel)]`)

22. What are Directives?

-> DOM instructions.

Structural (`*ngIf`), Attribute (`ngClass`)

23. What are Pipes?

-> Format data in UI.

Example: `{{ price | currency }}`

24. Input vs Output Decorators?

-> `@Input()` = Receive data. `@Output()` = Send event to parent.

25. ViewChild vs ViewChildren?

-> ViewChild = 1 child. ViewChildren = list of children.

Example:

`@ViewChild('input') inputEl: ElementRef;`

26. Angular Lifecycle Hooks?

## Top 50 .NET Full Stack Developer Interview Questions

-> ngOnInit, ngOnDestroy, etc.

27. What is JWT Token?

-> JSON Web Token for stateless auth.

Stored in headers, verifies user.

28. Authentication vs Authorization?

-> Auth: Who you are. Authorization: What you can access.

29. How to secure API in .NET Core?

-> JWT token, [Authorize], role-based access.

30. What is ASP.NET Core Identity?

-> Built-in user auth and management system.

31. DROP vs DELETE vs TRUNCATE?

-> DELETE: with condition. TRUNCATE: all, fast. DROP: remove table.

32. IEnumerable vs IQueryable?

-> IEnumerable: in-memory. IQueryable: DB-level query.

33. SQL JOIN Example:

```
SELECT s.Name, c.City FROM Student s JOIN City c ON s.CityId = c.Id;
```

34. Count Students Across Subjects:

## Top 50 .NET Full Stack Developer Interview Questions

```
SELECT COUNT(DISTINCT StudentID) FROM StudentSubjects;
```

35. EF Core Migration:

Add-Migration InitialCreate

Update-Database

36. Reverse a String:

```
string s = "hello";
```

```
string rev = new string(s.Reverse().ToArray());
```

37. Remove Duplicates:

```
int[] arr = {1,2,2,3};
```

```
var unique = arr.Distinct().ToArray();
```

38. Count Vowels:

```
int count = "hello".Count(c => "aeiou".Contains(c));
```

39. Longest Word:

```
string[] words = sentence.Split();
```

```
string longest = words.OrderByDescending(w => w.Length).First();
```

40. CRUD in Angular + .NET:

POST - Create

GET - Read

PUT - Update

## Top 50 .NET Full Stack Developer Interview Questions

DELETE - Delete

41. Country-State Dropdown:

Load countries via API.

On change, fetch states by selected countryId.

42. Exception Handling:

Use try-catch, logging, UseExceptionHandler middleware.

43. What is a Design Pattern?

-> Standard solutions for recurring problems.

Example: Singleton, Factory, Repository

44. SOLID Principles:

S - Single Responsibility

O - Open/Closed

L - Liskov Substitution

I - Interface Segregation

D - Dependency Inversion

45. ViewModel in MVC:

-> Custom model for Views, not DB mapped.

46. Single() vs First():

Single() - exactly one match



## Top 50 .NET Full Stack Developer Interview Questions

First() - first match

OrDefault() returns default if none

47. HTTP Methods:

GET - Read

POST - Create

PUT - Update

DELETE - Remove

48. Why appsettings.json?

-> Store connection strings, secrets, keys.

49. Repository Pattern:

-> Abstraction over DB logic. Promotes testability.

50. .csproj vs .cs files:

.csproj - Project settings.

.cs - Source code files.