Interview Questions for 3 Years’ Experience

Tell me about your experience with .NET Core and Web API.

Ans.

I’ve been working with .NET Core and Web API for 3 years. I’ve built RESTful APIs using ASP.NET Core, following SOLID principles and Clean Architecture. I’m comfortable with routing, model binding, middleware, exception handling, and securing APIs using JWT. I’ve also integrated APIs with frontend apps and third-party services.

1. Explain difference between .Net and C#?

Ans.

* .Net is a software framework developed by Microsoft used for building and running applications.
* C# is an object-oriented programming language also developed by Microsoft that is used to write code that runs on .Net framework.

1. Which is language would you prefer to use C#, VB.net, F# and why?

Ans.

* **Popularity & Demand:** C# because it is most widely used programming language for application development.
* **Modern Features:** C# offers modern features like LINQ, async/await for asynchronous programming, and lambda expressions, making it a powerful, efficient, and clean language for development.
* **Cross-Platform:** With .NET Core, C# is now cross-platform, allowing developers to build applications for Windows, Linux, and macOS.

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

Ans.

* .Net framework is used for building windows-based application only.
* Where .Net core is a cross-platform, open-source i.e. it can be used for building application that run on multiple platforms, supporting windows, macOs, Linux
* .Net core has better performance than .Net.

1. Explain Dependency Injection (DI) in .NET Core.

Ans.

Dependency Injection is a design pattern used to achieve loose coupling between classes and their dependencies. This allows objects to provide with their dependencies rather than creating them internally. In .Net Core DI is built in and uses a container to manage dependencies. It is used to inject services into controllers, classes, and components. The common DI container in .Net core is Startup.cs or Program.cs files, using methods like AddSingleton, AddScoped, and AddTransient to register services.

1. What are Transient, Scoped and Singleton?

Ans.

Transient, Scoped and Singleton are the lifetime options for services that you register in DI container.

* Transient: The instance of the service is created every time when requested.
* Scoped: The instance of the service is created for once per request or scope.
* Singleton: Only single instance of the service is created once when the application runs is used throughout the entire application.

1. Explain how Entity Framework Core works in .NET Core applications.

Ans.

Entity Framework (EF) Core is an Object-Relational Mapping (ORM) framework that allows developers to interact with databases using object-oriented programming concepts. It eliminates the need for writing raw SQL queries, simplifying data access.

EF Core uses DbContext to represent the session with the database. A DbSet<T> represents a collection of entities in the context. You can use LINQ to query the database, and EF Core will translate those queries into SQL.

1. What is difference between IEnumerable and IQueryable in LINQ?

Ans.

IEnumerable is the interface in LINQ which is used to query In-Memory collections like Array, List etc. For performance it is sufficient for in-memory data but it does not work well with remote data sources like database because it doesn't translate LINQ to SQL.

IQueryable is the interface in LINQ which is used to query remote data sources like databases, where the query is translated into SQL and executed on the database.

It is more efficient when dealing with large datasets because it can push the query execution to the server.

9. What are the advantages of using stored procedures over LINQ queries in .NET Core?

Ans. Stored procedures provide several advantages:

* Performance: Stored procedures are precompiled and optimized by the database engine, making them faster than dynamic queries generated by LINQ.
* Security: Stored procedures can encapsulate complex logic, reducing the risk of SQL injection attacks.
* Code Reusability: Stored procedures can be reused across multiple applications or services.
* Maintenance: Stored procedures allow database logic to be managed and optimized independently of application code.

10. How do you secure a Web API?

Ans.

* Use JWT (JSON Web Tokens) for authentication.
* Configure authentication in Startup.cs using AddAuthentication ().
* Add [Authorize] attributes to controllers or actions.
* Use HTTPS, API rate limiting, input validation, etc.

11. Difference between FirstOrDefault and SingleOrDefault in LINQ?

Ans.

* FirstOrDefault returns the first element or returns null if none.
* SingleOrDefault returns the only element if exactly one match is found or throws element If more than one element is found or return null if nothing is found.

12. What are the HTTP methods used in REST APIs?

Ans.

* GET – To Retrieve data
* POST – To Create data
* PUT – To Update data
* Patch – To partially update the data
* Delete – To remove data.