Q) Explain the usage of Dependency Injection in ASP.NET Core

A) Dependency Injection is the design pattern that helps us to create an application which loosely coupled. This means that objects should only have those dependencies that are required to complete tasks. The main advantage of DI (Dependency Injection) is our application is loosely coupled and has provided greater maintainability, testability, and also re-usability. It is loosely coupled because dependency required by the class is injected from the outer world rather than created themselves directly win-in code.

Q) What is routing in ASP.NET Core?

"Routing is responsible for matching incoming HTTP requests and dispatching those requests to the app's executable endpoints. Endpoints are the app's units of executable request-handling code. Endpoints are defined in the app and configured when the app starts. The endpoint matching process can extract values from the request's URL and provide those values for request processing. Using endpoint information from the app, routing is also able to generate URLs that map to endpoints.

Apps can configure routing using:

Controllers

Razor Pages

SignalR

gRPC Services

Endpoint-enabled middleware such as Health Checks.

Delegates and lambdas registered with routing."

Q) How many kinds of lifetimes does build-in IOC container supports in .net core ?

A) The built-in IoC container supports three kinds of liftimes

Singleton: IoC container will create and share a single instance of a service throughout the application’s lifetime.

Transient: The IoC container will create a new instance of the specified service type every time you ask for it.

Q) Describe the request flow in the ASP.NET MVC framework

A) Request flow handles the request from the clients and passes it to the server. The Request flow is as follows:

-Request is being taken from User to controller.

-Controller processes the request from the user and creates a data Model of that particular request.

-Data model that is being created is then passed to View that handles the frontend or the design.

-View then transforms the Data Model by using its own functions in an appropriate output format.

-The output format that is being given by the View is then gets rendered to the Browser and the View will be seen by the user.

Q) Explain difference between Task and Thread in .net.

A) A thread is one of the many possible workers which performs that task.

In .NET 4.0 terms, a Task represents an asynchronous operation. Thread(s) are used to complete that operation by breaking the work up into chunks and assigning to separate threads

Q) Explain difference between sync and async method (Provide example)?

A) Async is multi-thread, which means operations or programs can run in parallel. Sync is single-thread, so only one operation or program will run at a time.

Async is non-blocking, which means it will send multiple requests to a server. Sync is blocking — it will only send the server one request at a time and will wait for that request to be answered by the server.

Async increases throughput because multiple operations can run at the same time. Sync is slower and more methodical.

Q) What is the difference between Web API and WCF?

A) WCF offers request-reply, one-way, or duplex while Web API is by default request-reply only.

WCF is used for developing SOAP-based services whereas Web API is used for both SOAP-based and RESTful services.

WCF does not offer any support for MVC features whereas Web API supports MVC features.

WCF supports HTTP, UDP, and custom transport protocol whereas Web API supports only HTTP protocol.

WCF offers Text, MTOM, and Binary Encoding support whereas Web API supports the UTF-8 encoding format.

WCF supports message queues, message security, duplex communication, transaction whereas Web API doesn’t support.

WCF stands for Windows Communication Foundation whereas API stands for Application Program Interface

Q) Difference between HTTP GET vs HTTP Post?

A)

|  |  |
| --- | --- |
| **HTTP GET** | **HTTP POST** |
| In GET method we can snot send large amount of data rather limited data is sent because the request parameter is appended into the URL. | In POST method large amount of data can be sent because the request parameter is appended into the body. |
| GET request is comparatively better than Post so it is used more than the  Post request. | POST request is comparatively less better than Get so it is used less than the Get request. |
| GET request is comparatively less secure because the data is exposed in the URL bar. | POST request is comparatively more secure because the data is not exposed in the URL bar. |
| Request made through GET method are stored in Browser history. | Request made through POST method is not stored in Browser history. |
| GET method request can be saved as bookmark in browser. | POST method request can not be saved as bookmark in browser. |
| Request made through GET method are stored in cache memory of Browser. | Request made through POST method are not stored in cache memory of Browser. |
| Data passed through GET method can be easily stolen by attackers. | Data passed through POST method can not be easily stolen by attackers. |
| In GET method only ASCII characters are allowed. | In POST method all types of data is allowed. |

Q) What is Postback?

A) PostBack is the name given to the process of submitting an ASP.NET page to the server for processing. PostBack is done if certain credentials of the page are to be checked against some sources (such as verification of username and password using database). This is something that a client machine is not able to accomplish and thus these details have to be 'posted back' to the server

Q) Explain Singleton design pattern.

A) The singleton pattern is one of the simplest design patterns. Sometimes we need to have only one instance of our class for example a single DB connection shared by multiple objects as creating a separate DB connection for every object may be costly. Similarly, there can be a single configuration manager or error manager in an application that handles all problems instead of creating multiple managers.

Definition:

The singleton pattern is a design pattern that restricts the instantiation of a class to one object.