|  |  |
| --- | --- |
| 1 | What is the role of the Startup.cs class? Explain with example. |
| 2 | Explain the complete request processing pipeline in ASP.NET Core MVC |
| 3 | What problems does Dependency Injection solve? |
| 4 | Describe the service lifetimes |
| 5 | What is request delegate? |
| 6 | Explain the various configuration providers in ASP.NET Core |
| 7 | What is In-memory cache? |
| 8 | How to prevent Cross-Site Scripting (XSS) in ASP.NET Core? |
| 9 | Explain View-Model with an example. |
| 10 | Describe Attribute based routing |
| 11 | How does ASP.NET Core support dependency injection into views? |
| 12 | Differentiate between launchSettings.json and appSettings.json |
| 13 | What are Tag Helpers? |
| 14 | What is response caching? |
| 15 | Differentiate between ViewBag and ViewData with examples |
| 16 | difference between app.Use vs app.Run while adding middleware |
| 17 | What does WebHost.CreateDefaultBuilder() do? |
| 18 | What is the role of IHostingEnvironment interface in ASP.NET Core? |
| 19 | What is CTS in .Net core? |
| 20 | Difference between ConfigureServices and Configure Method in Startup class |

1.

The Startup class contains the Configure Services and Configure methods. While the former is used to configure the required services, the latter is used to configure the request processing pipeline. The Configure method is executed immediately after the Configure Services method

2.

The ASP.NET Core request processing pipeline consists of a sequence of middleware components that are going to be called one after the other. Each middleware component can perform some operations before and after invoking the next component using the next method

3.

Dependency injection helps to develop testable code, allowing developers to write unit tests easily

4.

The service lifetime controls how long a result object will live for after it has been created by the container.

5.

Request delegates are used to build the request pipeline. The request delegates handle each HTTP request. Request delegates are configured using Run, Map, and Use extension methods.

6.

* File configuration provider (default)
* Secret Manager (default)
* Environment Variables configuration provider (default)
* Command-line configuration provider (default)
* Memory configuration provider.
* Azure Key Vault configuration provider.

7.

ASP.NET Core supports several different caches. The simplest cache is based on the IMemoryCache. IMemoryCache represents a cache stored in the memory of the web server. Apps running on a server farm (multiple servers) should ensure sessions are sticky when using the in-memory cache.

8.

* Never put untrusted data into your HTML input, unless you follow the rest of the steps below. Untrusted data is any data that may be controlled by an attacker, HTML form inputs, query strings, HTTP headers, even data sourced from a database as an attacker may be able to breach your database even if they cannot breach your application.
* Before putting untrusted data inside an HTML element ensure it's HTML encoded. HTML encoding takes characters such as < and changes them into a safe form like &lt;
* Before putting untrusted data into an HTML attribute ensure it's HTML encoded. HTML attribute encoding is a superset of HTML encoding and encodes additional characters such as " and '.
* Before putting untrusted data into JavaScript place the data in an HTML element whose contents you retrieve at runtime. If this isn't possible, then ensure the data is JavaScript encoded. JavaScript encoding takes dangerous characters for JavaScript and replaces them with their hex, for example < would be encoded as \u003C.
* Before putting untrusted data into a URL query string ensure it's URL encoded.

9.

A view model represents the data that you want to display on your view/page, whether it be used for static text or for input values (like textboxes and dropdown lists) that can be added to the database (or edited).

public class Employee : IEntity

{

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public DateTime DateCreated { get; set; }

}

10.

It used to define routes.

attribute routing uses attributes to define routes.

11.

ASP.NET Core supports dependency injection into views. This can be useful for view-specific services, such as localization or data required only for populating view elements. Most of the data views display should be passed in from the controller.

12.

* launchSettings. json, which is placed in the Properties folder of a project, describes how the application can be launched — the command to execute, whether the browser should be opened, which environment variables should be set, and so on.
* appsettings. json file is generally used to store the application configuration settings such as database connection strings, any application scope global variables, and much other information

13.

A Tag Helper Component is a Tag Helper that allows you to conditionally modify or add HTML elements from server-side code.

14.

* Response Caching is the mechanism of caching a server’s response by a browser or other clients. This will help in serving future requests for the same resources very quickly. Additionally, this will free up the server from processing and generating the same response multiple times.
* ASP.NET Core uses the ResponseCache attribute to set the response caching headers. Furthermore, we can use the Response Caching Middleware to control the caching behavior from the server-side. Once we set the response caching headers, clients and other proxies can read those to determine how to cache the response from the server.

15.

1. ViewData in MVC
   * ViewData is property of ControllerBase class.
   * ViewData is a type of dictionary object.
   * ViewData is key-value dictionary collection.
   * ViewData was introduced in MVC 1.0 version.
   * ViewData works with .Net framework 3.5 and above.
   * Need to do type conversion of code while enumerating.
   * ViewData object keeps data only for current request.
2. ViewBag in MVC
   * ViewBag is property of ControllerBase class.
   * ViewBag is a type of dynamic object.
   * ViewBag is a type of object.
   * ViewBag was introduced in MVC 3.0 version.
   * ViewBag works with .Net framework 4.0 and above.
   * ViewBag uses property and handles it, so no need to do type conversion while enumerating.
   * ViewBag object keeps data only for current request.

16.

For app.Run, it adds a terminal middleware delegate to the application's request pipeline.

For app.Use, it adds a middleware delegate to the application's request pipeline.

17.

* Uses Kestrel server
* Sets current directory as the Content Root
* Adds the appsettings.json and appsettings.{EnvironmentName}.json files as a config source
* Adds user secrets as a config source (only in development environment)
* Adds environment variables and commandline arguments as config sources
* Configures logging with settings from the Logging configuration section
* Adds Console and Debug loggers
* Uses IIS integration
* Adds a scope validator that checks if scoped services are not resolved from the root provider (in development only)

18.

IHostingEnvironment Interface (Microsoft. AspNetCore. Hosting) Provides information about the web hosting environment an application is running in.

The IHostingEnvironment is an interface for .Net Core 2.0.

The IHostingEnvironment interface need to be injected as dependency in the Controller and then later used throughout the Controller.

The IHostingEnvironment interface have two properties.

1. Webroot Path – Path of the www folder.

2. ContentRootPath – Path of the root folder which contains all the Application files.

19.

The Common Type System (CTS) is a standard for defining and using data types in the. NET framework. CTS defines a collection of data types, which are used and managed by the run time to facilitate cross-language integration.

20.

* Configure Services is used to configure Dependency Injection
* Items in Configure Services are part of Dependency Injection like logger, Database etc. These kinds of things aren't directly associated with a http request.
* Configure is used to set up middleware’s, routing rules, etc.
* Items in configure are part of a http request like routing, middleware’s, static files all these triggers directly when user makes a request.