Coding Problems

Answers

Q1. What is the ASP.NET Core?

Answer:
ASP.NET Core is a cross-platform, high-performance, open-source framework for building modern, cloud-enabled, Internet-connected apps. It is not an upgraded version of ASP.NET. ASP.NET Core is completely rewriting that work with .net Core framework.

With ASP.NET Core, you can:

- Build web apps and services, Internet of Things (IoT) apps, and mobile backends.
- Use your favorite development tools on Windows, macOS, and Linux.
- Deploy to the cloud or on-premises.
- Run on .NET Core.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q2. What are the features provided by ASP.NET Core?

Answer:

- Cross-platform - the previous version of asp.net will run on only windows platform but this core framework will run on Windows, Linux, mac.
- Hosting - Previous version of asp host only on IIS but this framework will host on Apache, Docker, Self-hosting also.
- It also provides the same single unified programming model for creating web MVC application and web API also.
- Asp.net core has built-in support for dependency injection.
- Open source- It is developed by Microsoft but it is collaborating through the vast community of open-source developers. Because of the community, it is evolving very fast.
- Modular - As asp.net providing middleware component so modularity is achieved to a great extent.so modularity is achieved to a great extent.
- In this, both request and response pipeline are composed using a middleware component.
- We can also create our custom middleware component.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

f 🐦 in

Q3. What is Metapackages?

Answer:

The framework .NET Core 2.0 introduced Metapackage that includes all the supported package by ASP.NET code with their dependencies into one package. It helps us to do fast development as we don't require to include the individual ASP.NET Core packages. The assembly Microsoft.AspNetCore.All is a meta package provide by ASP.NET core.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

f 🐦 in

Q4. Can ASP.NET Core application work with full .NET 4.x Framework?

Answer:

Yes. ASP.NET core application works with full .NET framework via the .NET standard library.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

f 🐦 in

Q5. What is the startup class in ASP.NET core?

Answer:

❖ Startup class is the entry point of the ASP.NET Core application.

❖ Every .NET Core application must have this class.

❖ This class contains the application configuration related items.

❖ It is not necessary that class name must "Startup", it can be anything, we can configure startup class in Program class.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

f 🐦 in

Q6. What is the use of ConfigureServices method of startup class?

Answer:

❖ This is an optional method of startup class.

❖ This method is used to configure services that are used by the application.

❖ When the application is requested for the first time, it calls ConfigureServices method.

❖ This method must be declared with a public access modifier, so that environment will be able to read the content from metadata.

❖ ASP.net core has built-in support for Dependency Injection. We can add services to DI container using this method.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q7. What is middleware?

Answer:

❖ Middleware is a piece of code in an application pipeline used to handle requests and responses.

❖ Middleware can be built-in as part of the .NET Core framework, added via NuGet packages, or can be custom middleware.

❖ These middleware components are configured as part of the application startup class in the configure method.

❖ Configure methods set up a request processing pipeline for an ASP.NET Core application.

❖ It consists of a sequence of request delegates called one after the other.

❖ For example, we may have a middleware component to authenticate a user, another piece of middleware to handle errors, and another middleware to serve static files such as JavaScript files, CSS files, images, etc.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q8. What is the difference between IApplicationBuilder.Use() and IApplicationBuilder.Run()?

Answer:
We can use both the methods in Configure methods of startup class. Both are used to add middleware delegate to the application request pipeline. The middleware adds using IApplicationBuilder.Use may call the next middleware in the pipeline whereas the middleware adds using IApplicationBuilder.Run method never calls the subsequent ore next middleware. After IApplicationBuilder.Run method, system stop adding middleware in request pipeline.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q9. What is the use of "Map" extension while adding middleware to ASP.NET Core pipeline?

Answer:

❖ Map extensions are used as convention for branching the pipeline.

❖ The Map extension method is used to match request delegates based on a request's path.

❖ Map simply accepts a path and a function that configures a separate middleware pipeline.

❖ If the request path starts with the given path, the branch is executed.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q10. What is routing in ASP.NET Core?

Answer:

❖ Routing in ASP.NET Core MVC is the mechanism through which incoming requests are mapped to controllers and their actions.

- ❖ This is achieved by adding Routing middleware to the pipeline and using IRouteBuilder to map URL pattern (template) to a controller and action.
- ❖ All the routes are registered when the application is started.
- ❖ There are two types of routing supported by ASP.NET Core
- ❖ The conventional routing
- ❖ Attribute routing

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q11. How to enable Session in ASP.NET Core?

Answer:
The middleware for the session is provided by the package Microsoft.AspNetCore.Session. To use the session in ASP.NET Core application, we need to add this package to csproj file and add the Session middleware to ASP.NET Core request pipeline.

```
 public class Startup

  {

   public void ConfigureServices(IServiceCollection services)

  {

   services.AddSession();

   services.AddMvc();

  }

  public void Configure(IApplicationBuilder app, IHostingEnvironment env)

   {

….

app.UseSession();

….

  }

}
```

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q12. What are the various JSON files available in ASP.NET Core?

Answer:
There are following JSON files in ASP.NET Core :

❖
global.json

❖
launchsettings.json

❖
appsettings.json

❖
bundleconfig.json

❖
bower.json

❖
package.json

Source: ItLabz Technology

Author : Ishant Charagi

(0) (0)

Q13. What is tag helper in ASP.NET Core?

Answer:
❖ It is a feature provided by Razor view engine that enables us to write server-side code to create and render the HTML element in view (Razor).

❖ Tag Helpers are classes written in C# but are attached to HTML elements in order to run server-side code from Razor view.

❖ In other words, view created in HTML has its presentation logic defined in C#, which is ultimately executed on the web server.

Source: ItLabz Technology

Author : Ishant Charagi

(0) (0)

Q14. How to disable Tag Helper at element level?

Answer:
We can disable Tag Helper at element level using the opt-out character ("!"). This character must apply opening and closing the Html tag.

Example

<!span asp-validation-for="phone" class="divPhone"></!span>

Source: ItLabz Technology

Author : Ishant Charagi

(0) (0)

Q15. What is Kestrel?

Answer:
Kestrel is an open-source, cross-platform web server designed for ASP.NET Core. Kestrel is included and enabled by default in ASP.NET Core. It is very light-weight when compared with IIS.Kestrel can be used as a web server processing requests directly from a network, including the Internet.

Though Kestrel can serve an ASP.NET Core application on its own, Microsoft recommends using it along with a reverse proxy such as IIS, Nginx, or Apache, for better performance, security, and reliability.

Source: ItLabz Technology

Author : Ishant Charagi

(0) (0)

Q16. How to specify service lifetime for register service that added as a dependency?

Answer:
The built-in IoC container supports three kinds of lifetimes -

Singleton - IoC container will create and share a single instance of a service throughout the application's lifetime. The service can be added as a singleton using AddSingleton method of IServiceCollection.

Transient - The IoC container will create a new instance of the specified service type every time you ask for it. The service can be added as Transient using AddTransient method of IServiceCollection.

Scoped - IoC container will create an instance of the specified service type once per request and will be shared in a single request. The service can be added as scoped using an AddScoped method of IServiceCollection.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q17. When do you choose classic ASP.NET MVC over ASP.NET Core?

Answer:
Though it’s a better choice in almost all the aspects, you don’t have to switch to ASP.NET Core if you are maintaining a legacy ASP.NET application that you are happy with and that is no longer actively developed.

ASP.NET MVC is a better choice if you:

❖ Don’t need cross-platform support for your Web app.

❖ Need a stable environment to work in.

❖ Have nearer release schedules.

❖ Are already working on an existing app and extending its functionality.

❖ Already have an existing team with ASP.NET expertise

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q18. What is CoreCLR?

Answer:
CoreCLR is the .NET execution engine in .NET Core, performing functions such as garbage collection and compilation to machine code.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q19. How ASP.NET Core serve static files?

Answer:

In ASP.NET Core, Static files such as CSS, images, JavaScript files, HTML are served directly to the clients. The wwwroot folder contains all these static files. UseStaticFiles() method inside Startup.Configure enables the static files to be served to client.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Q20. What is the purpose of IHostingEnvironment interface in ASP.NET Core?

Answer:
The "HostingEnvironment Interface" provides information about the web hosting environment an application is running in. IHostingEnvironment, allows you to programmatically retrieve the current environment so you can have an environment-specific behaviour.

Source: ItLabz Technology

Author : Ishant Charagi

👍 (0) 👎 (0)

Comment Box

Enter Your Comment

Name

Email ID

Phone

Post Comment

No more Comments

About Us

Authors

Expert Videos

FAQ