Q1: What is an Object?

Q2: What is the difference between continue and break statements in C#?

Q3: What is C#?

Q4: What are Property Accessors?

**C#**

Q5: What are Nullable types in C#?

Q6: What are Reference Types in C#?

Q7: What are generics in C#?

Q8: What is an Abstract Class?

Q9: What is namespace in C#?

Q10: What is Serialization?

Q11: What are dynamic type variables in C#?

Q12: Can this be used within a Static method?

Q13: In how many ways you can pass parameters to a method?

Q14: What is the difference between a Struct and a Class in C#?

Q15: What are the different types of classes in C#?

Q16: What is Boxing and Unboxing? Related To: .NET Core

Q17: What is the difference between string and StringBuilder in C#?

Q18: How is Exception Handling implemented in C#?

Q19: What is LINQ in C#? Related To: LINQ

Q20: Can multiple catch blocks be executed?

Q21: What is enum in C#?

Q22: What are partial classes?

Q23: Why to use finally block in C#?

Q24: What you understand by Value types and Reference types in .NET? Provide some comparison.

Q25: What is Managed or Unmanaged Code?

Q26: What is the difference between a class and a structure? Related To: OOP

Q27: What is the difference between ref and out keywords?

Q28: Is there a difference between throw and throw ex?

Q29: Why can't you specify the accessibility modifier for methods inside the Interface?

Q30: What is the difference between Virtual method and Abstract method? Related To:

Q31: What is the use of Null Coalescing Operator (??) in C#?

Q32: What is Virtual Method in C#?

Q33: What is the difference between Interface and Abstract Class? Related To: OOP

Q34: What is lambda expressions in C#?

Q35: Explain Anonymous type in C#

Q36: What is the difference between dynamic type variables and object type variables?

Q37: What is the difference between Equality Operator (==) and Equals() Method in C#

Q38: What are the uses of using in C#

Q39: How encapsulation is implemented in C#?

Q40: What is the difference between overloading and overriding?

Q41: How can you prevent a class from overriding in C#? Related To: OOP

Q42: Explain Code Compilation in C#

Q43: What is an anonymous function in C#?

Q44: What is difference between constant and readonly?

Q45: What is Reflection in C#.Net?

Q46: What is a Destructor in C# and when shall I create one?

Q47: What is scope of a Internal member variable of a C# class?

Q48: What is Extension Method in C# and how to use them?

Q49: What is sealed Class in C#?

Q50: Is there a way to catch multiple exceptions at once and without code duplication?

Q51: Why to use of the IDisposable interface? Related To: .NET Core

Q52: Explain the difference between Task and Thread in .NET Related To: .NET Core

Q54: Explain what is Ternary Search? CSJava Related To: Searching

Q55: What interface should your data structure implement to make the Where method work?

Q56: What is difference between late binding and early binding in C#?

Q57: Is operator overloading supported in C#?

**C#**

Q58: Why to use lock statement in C#?

Q59: What is the Constructor Chaining in C#?

Q60: What is the difference between dispose and finalize methods in C#?

Q61: What is the difference between is and as operators in C#?

Q62: What is Marshalling and why do we need it?

Q63: Can you create a function in C# which can accept varying number of arguments?

Q64: What are the different ways a method can be overloaded?

Q65: When to use ArrayList over array[] in C#?

Q66: When would you use delegates in C#?

Q67: What is the best practice to have best performance using Lazy objects?

Q68: What is scope of a Protected Internal member variable of a C# class?

Q69: What is Indexer in C#?

Q70: Explain what is Short-Circuit Evaluation in C#

Q71: What are pointer types in C#?

Q72: What is the yield keyword used for in C#?

Q73: IEnumerable vs List - What to Use? How do they work?

Q74: What's the difference between StackOverflowError and OutOfMemoryError?

Q75: What is the difference between System.ApplicationException class and System.SystemException class?

Q76: Explain the difference between Select and Where Related To: LINQ

Q77: What is the difference between Func<string,string> and delegate?

Q78: Can Multiple Inheritance implemented in C# ?

Q79: What is the use of conditional preprocessor directive in C#?

Q80: What is a static constructor? Related To: OOP

Q81: Explain how does Asynchronous tasks Async/Await work in .NET? Related To: .NET

Q82: What happens when we Box or Unbox Nullable types?

Q83: Can you explain the difference between Interface, abstract, sealed, static and partial class in C#?

Q84: How to solve Circular Reference? Related To: OOP

Q85: What's the difference between the System.Array.CopyTo() and System.Array.Clone()

Q86: What is jagged array in C# and when to prefer jagged arrays over multi-dimensional arrays?

Q87: What is the volatile keyword used for?

Q88: What is the method MemberwiseClone() doing?

Q89: Can you add extension methods to an existing static class?

Q90: Explain the difference between IQueryable, ICollection, IList & IDictionary interfaces?

Q91: What is the use of static constructors?

Q92: Implement the Where method in C#. Explain.

Q93: What is the difference between Lambdas and Delegates?

Q94: What is a preprocessor directives in C#?

Q95: Could you explain the difference between Func vs. Action vs. Predicate?

Q96: What is Multicast Delegate in C#?

Q97: Why Abstract class cannot be sealed or static?

Q98: What are the differences between IEnumerable and IQueryable?

Q99: in C#, when should we use abstract classes instead of interfaces with extension methods?

Q100: Could you explain the difference between destructor, dispose and finalize method and when to use?

Q101: What are Circular References in C#?

Q102: What is deep or shallow copy concept in C#?

Q103: Explain what is Weak Reference in C#?

Q104: What are the benefits of a Deferred Execution in LINQ? Related To: LINQ

Q105: List some different ways for equality check in .NET

Q106: You have defined a destructor in a class that you have developed by using the C#

Q107: Why doesn't C# allow static methods to implement an interface? Related To: OOP

* Model View Controller, Folders => Model / Views / Controllers / App\_Start
* advantages of MVC
  + Multiple view support, the user interface can display multiple views of the same data at the same time.
  + Change Accommodation, adding new types of views to the system generally does not affect the model.
  + SoC – Separation of Concerns, separation of the UI, Business Logic, Model or Data.
  + More Control, more control over HTML, JavaScript, and CSS than the traditional Web Forms.
  + Testability, good support for test driven development.
  + Lightweight, doesn’t use View State and thus reduces the bandwidth of the requests to an extent.
* MVC application life cycle
  + Browser
  + Request
  + Routing

**ASP.NET MVC**

* + MVC Handler
  + Controller
  + Action Execution
    - Authentication Filters
      * Authorization Filters
        + Model Binding
        + Action Filters

Action Method

|  |  |  |
| --- | --- | --- |
| Action Filters  View | View Engine  Response | View Result |

* return types of a controller action method

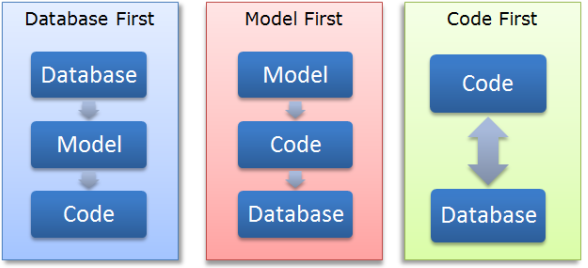
|  |  |  |
| --- | --- | --- |
| ViewResult (View)  PartialviewResult (Partialview)  RedirectResult (Redirect) | jsonResult (json)  javascriptResult (javascript)  ContentResult (Content) | FileResult (File)  EmptyResult |

* Filters in MVC
  + Action Filters, implement logic that gets executed before and after a controller action executes.
    - Output Cache, caches the output for a specified amount of time.
  + Authorization Filters, implement authentication and authorization for controller actions.
  + Result Filters, contain logic that is executed before and after a view result is executed, you might want to modify a view result right before the view is rendered to the browser.
  + Exception Filters, last type of filter to run, handle errors raised by either your controller actions or controller action results. can also use exception filters to log errors.
    - can be applied over the action method as well as Controller or at the global level. Global.asax.cs (Application\_Start)  
      public static void RegisterGlobalFilters(GlobalFilterCollection flt)  
      {  
       flt.Add(new HandleErrorAttribute());//[HandleError(View = "Error")]  
      }
* Routing is a mechanism to process the incoming URL that is more descriptive, default Controller/Action/Param
* TempData, TempDataDictionary, pass data from one page to another
* ViewData, ViewDataDictionary , pass data from controller to view
* ViewBag, dynamic property, pass data from the controller to the respective view
* A partial view is a chunk of HTML that can be safely inserted into an existing DOM / Page. Like User Control
* View, page layout, not light weight, it is like a page, render after viewstart page is rendered.
* Partial View, is not page layout, only be render inside view just like a user control, RenderPartial method
* Attribute based routing, MVC 5 route can define the URL structure with action. [Route("Users/about")]
* Razor, previously aspx view engine was used, MVC 3 has introduced a new view engine called Razor.
* HTML helpers, just like web form controls in ASP.NET, does not have an event, can create your own helpers

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BeginForm() | TextArea()  TextBox() | CheckBox()  RadioButton() | ListBox() | Hidden() | Password() | DropDownList() | EndForm() |  |

* Forms authentication, it occurs after IIS authentication is completed,   
  <system.web>  
   <authenticationmode="Forms">  
   <formsloginUrl="Login.aspx" protection="All" timeout="30"  
   name=".ASPXAUTH" path="/" equireSSL="false"  
   slidingExpiration="true" defaultUrl="default.aspx"  
   cookieless="UseDeviceProfile" enableCrossAppRedirects="false" />  
   </authentication>  
  </system.web>
* Areas in MVC 2.0, divide or “isolate” the modules of large applications in multiple or separated MVC.
* ASP.NET MVC 5 provide a way of separating page content from the way it is rendered on various devices, like web, mobile, iPhone, iPod and Windows Phones.
* ASP.NET Scaffolding is a code generation framework for ASP.NET Web applications (MVC / Web APIs). Reduce the amount of time to develop standard data operations.
* The Route Constraint in ASP.NET MVC Routing allows us to apply a regular expression to a URL segment to restrict whether the route will match the request. In simple words, we can say that the Route constraint is a way to put some validation around the defined route. Suppose you have defined the following route in your application.
* Caching has huge advantages, such as it reduces server round trips, reduces database server round trips, reduces network traffic etc. Cached Content Locations

**ASP.NET MVC**

* + Any (Default), stored on the server
  + Client, cache is located on the browser client
  + Downstream, tored in any HTTP 1.1 cache-capable devices other than the origin server.
  + Server, located on the Web server
  + None, cache is disabled
  + ServerAndClient, cache can be stored only at the origin server or at the requesting client.
* Bundling and Minification, two new techniques introduced to improve request load time
  + Bundling, multiple JavaScript (.js) files or multiple cascading style sheet (.css)
  + Minification, removes whitespaces, performs other types of compression of files as small as possible.
* Validation Summary, generates an unordered list (ul element) of validation messages that are in the ModelStateDictionary object. can be used to display all the error messages for all the fields.
* Approach in MVC using Entity Framework
  + **Model First** allows you to create a new model using the Entity Framework Designer and then generate a database schema from the model.
  + **Database First** allows you to reverse engineer a model from an existing database.
  + **Code First** allows you to use your domain classes to build the model which in-turn will be used by EF in different activity.
* ViewStart, Razor View Engine layout named \_ViewStart which is applied on all view (Like master pages).
* Data Annotations, nothing just validations required / RegularExpression / Range / Length
* <customErrors mode="On"> to use a custom page for this error. First you need to update your web.config file.
* Server Side Validation in MVC, validates any data passed to the controller action that is executing
  + Approach 1 - Manually Add Error to ModelState object, manually check each property of model object.
  + Approach 2 - uses the Data Annotation API to validate the model data. (**ModelState.IsValid**)
* Remote validation (ajax), validate specific posted data to a server without posting the entire form to the server.  
  [Remote("CheckExistingEmail","Home",ErrorMessage="Email already exists!")]
* RenderSection in MVC, specifies the name of the section we want to render at that location in the layout.
* System.Web.Dll from MVC 6, because it's consume 30k of memory, MVC 6 only requires 2k of memory.

Q1 What is Layout in MVC?

Q2 What is ASP.NET MVC?

Q3 What is Razor Pages?

Q4 What are Actions in MVC?

Q5 Can you explain Model, Controller and View in MVC?

Q6 Explain Bundle.Config in MVC?

Q7 What you mean by Routing in MVC?

**ASP.NET MVC**

Q8 Explain Sections is MVC?

Q9 What is the use of ViewModel in MVC?

Q10 What are NonAction methods in MVC?

Q11 What are Scaffold templates in MVC?

Q12 What is Razor View Engine?

Q13 What are the advantages of MVC over ASP.NET?

Q14 What is Attribute Routing in MVC?

Q15 What is the HelperPage.IsAjax Property?

Q16 What are HTML Helpers in MVC?

Q17 Explain ASP.NET WebAPI vs MVC

Q18 What is the difference between ViewResult() and ActionResult() in ASP.NET MVC?

Q19 What are some of the advantages of using ASP.NET MVC vs Web Forms?

Q20 What is the difference between ViewBag and ViewData in MVC?

Q21 Explain the methods used to render the views in MVC?

Q22 Can you explain RenderBody and RenderPage in MVC?

Q23 Can you explain the page life cycle of MVC?

Q24 Can a view be shared across multiple controllers? If Yes, How we can do that?

Q25 What is PartialView in MVC?

Q26 Explain the difference between MVC vs ASP.NET Web API Related To

Q27 What about MVC in .NET Core? Related To

Q28 In OOP, what is the difference between the Repository Pattern and a Service Layer? Related To

Q29 Describe Flux vs MVC? Related To

Q30 What is the difference between ViewData and TempData?

Q31 Explain Dependency Resolution in MVC?

Q32 Why to use Html.Partial in MVC?

Q33 What are Validation Annotations?

Q34 How route table has been created in ASP.NET MVC?

Q35 What is Separation of Concerns in ASP.NET MVC?

Q36 What are AJAX Helpers in MVC?

Q37 Is it possible to create web application with both WebForms and MVC? Related To

Q38 What is the difference between Html.Partial vs Html.RenderPartial & Html.Action vs Html.RenderAction?

Q39 What are the advantages of MVC?

Q40 MVC application life cycle

Q41 What are the Filters in MVC (Action Filters / Authorization Filters / Result Filters / Exception Filters)

Q42 Explain what is routing in MVC? What are the three segments for routing important?  
(Control Name / Action Method / Parameter)

Q43 What is Route in MVC

Q44 Mention what is the difference between Temp data, ViewData, and View Bag?

Q45 What are HTML helpers in MVC

Q46 Explain attribute based routing in MVC

Q47 What is Razor in MVC (Answer ViewEngine, MVC 3 has introduced Razor / ASPX)

Q48 How do you implement Forms authentication in MVC (Web.Config)

Q49 Explain Areas in MVC (way to divide modules of large application in multiple mvc)

Q50 What's new in MVC 6 (Removed System.Web.Dll because 30K File Size, MVC6 only requires 2k memory)

Q1 What is ASP.NET Web API? (Next Page)

Q2 Which status code used for all uncaught exceptions by default?  
By default, most exceptions are translated into an HTTP response with status code 500, Internal Server Error.

Q3 Explain the usage of HttpResponseMessage? A way of returning a message/data from your action.

Q4 What New Features are Introduced in ASP.NET Web API 2.0?  
Attribute Routing   
Cross Origin Resource Sharing (CORS) (Supports CallBack) - Normally not allowed  
Open Web Interface (self-hosting package like development virtual server)  
IHttpActionResult, creating a response from the controller action  
Web API OData (Open Data Protocol), web protocol for querying and updating data, added support for $expand, $select, and $value options

Q5 What are the Advantages of Using ASP.NET Web API? better choice for simpler, light weight services.

Q6 What are main return types supported in Web API? Void/HttpResponseMessage/IHttpActionResult/AnyType

Q7 What exactly is OAuth (Open Authorization)?   
OAuth is an open-standard authorization protocol or framework that describes how unrelated servers and services can safely allow authenticated access to their assets without actually sharing the initial, related, single logon credential.

Q8 What is the difference between ApiController and Controller?   
Web API controllers do not return views, they return data.

**Web Services**

Q9 Compare WCF vs ASP.NET Web API? (Next Page)

Q10 What are the differences between Web API and Web API 2?   
Enhanced feature IHttpActionResult return type / A new Routing Attribute

Q11 What is Attribute Routing in ASP.NET Web API 2.0? Routing is how Web API matches a URI to an action.

Q12 Name types of Action Results in Web API 2 (Repeated Question No.6)

Q13 Explain the difference between WCF RESTful Service vs ASP.NET Web API? (Next Page)

Q14 Explain the difference between MVC vs ASP.NET Web API, ASP.Net MVC Web App and 2nd one is for Services

Q15 Is it True that ASP.NET Web API has Replaced WCF? No, it's not true, WCF Soap Based and Web API Non-Soap

Q16 In OOP, what is the difference between the Repository Pattern and a Service Layer?  
Repository Pattern to perform database operations, Service Layer to perform business logic.

Q17 What is Delegating Handler? A series of message handlers are chained together.

Q18 Why are the FromBody and FromUri attributes needed in ASP.NET Web API? force read simple from request

Q19 What's the difference between OpenID and OAuth? OpenID for authentication and OAuth for authorization.

Q20 Explain briefly CORS (Cross-Origin Resource Sharing)?  
Browser security prevents a web page from making AJAX requests to another domain. This restriction is called the same-origin policy, and prevents a malicious site from reading sensitive data from another site. However, sometimes you might want to let other sites call your web API.

Q21 Can we use Web API with ASP.NET Web Form? Yes, ASP.NET Web API is packaged with ASP.NET MVC, it is easy to add Web API to a traditional ASP.NET Web Forms application.

Q22 How to Return View from ASP.NET Web API Method? You can only return data from web api method.

Q23 How to register exception filter globally? add an instance of the filter to the GlobalConfiguration. Configuration. Filters collection

Q24 What is ASP.NET Web API OData? Data access protocol way to query and manipulate data sets through CRUD

Q25 Explain advantages/disadvantages of using HttpModule vs DelegatingHandler?  
HTTP modules allow security code to execute early as part of the IIS pipeline.  
A series of message handlers are chained together.

Q26 Why should I use IHttpActionResult instead of HttpResponseMessage?  
IHttpActionResult we are only concentrating on the data not status codes,  
HttpResponseMessage, response content as well as status code.

Q27 Explain briefly OWIN (Open Web Interface for .NET) Self Hosting? (Next Page)

Q28 What is difference between WCF and Web API and WCF REST and Web Service? (Next Page)

Q29 Could you clarify what is the best practice with Web API error management?

Q30 Explain the difference between WCF, Web API, WCF REST and Web Service? (Next Page)

* A web service is a kind of software that is accessible on the Internet.

**Web Services**

* Types of Web Services
  + ASMX Web Service (SOAP Based)
    - These use the [WebMethod] attribute to assign to methods we want to expose
    - These are much simpler than WCF web references.
    - These must be added using the Web Reference
    - This is a wrapper over the wsdl.exe and can be used to create proxies for .NET 1.1 and .NET 2.0
    - Receives and sends XML format data.
    - It supports only HTTP protocol.
    - It can be hosted only on IIS.
  + Windows Communication Foundation (WCF) Services
    - It is also based on SOAP and return data in XML form.
    - It is the evolution of the web service(ASMX) and support various protocols like TCP, HTTP, HTTPS, Named Pipes, MSMQ.
    - The main issue with WCF is, its tedious and extensive configuration.
    - It is not open source but can be consumed by any client that understands xml.
    - It can be hosted with in the application or on IIS or using window service.
  + WCF Rest
    - To use WCF as WCF Rest service you have to enable webHttpBindings.
    - It supports HTTP GET and POST verbs by [WebGet] and [WebInvoke] attributes respectively.
    - To enable other HTTP verbs, you have to do some configuration in IIS to accept request of that particular verb on .svc files
    - Passing data through parameters using a WebGet needs configuration. The UriTemplate must be specified.
    - It supports XML, JSON and ATOM data format.
  + Web API (MVC Pattern)
    - This is the new framework for building HTTP services with easy and simple way.
    - Web API is open source an ideal platform for building REST-ful services over the .NET Framework.
    - Unlike WCF Rest service, it use the full feature of HTTP (like URIs, request/response headers, caching, versioning, various content formats)
    - It also supports the MVC features such as routing, controllers, action results, filter, model binders, IOC container or dependency injection, unit testing that makes it more simple and robust.
    - It can be hosted with in the application or on IIS.
    - It is light weight architecture and good for devices which have limited bandwidth like smart phones.
    - Responses are formatted by Web API’s MediaTypeFormatter into JSON, XML or whatever format you want to add as a MediaTypeFormatter.
* Whom choose between WCF or WEB API
  + Choose WCF when you want to create a service that should support special scenarios such as one-way messaging, message queues, duplex communication etc.
  + Choose WCF when you want to create a service that can use fast transport channels when available, such as TCP, Named Pipes, or maybe even UDP (in WCF 4.5), and you also want to support HTTP when all other transport channels are unavailable.
  + Choose Web API when you want to create a resource-oriented services over HTTP that can use the full features of HTTP (like URIs, request/response headers, caching, versioning, various content formats).
  + Choose Web API when you want to expose your service to a broad range of clients including browsers, mobiles, iphone and tablets.
* Web API Error Management, problem of error handling from two angles: the what and the how.
  + Use status codes appropriately  
    Providing helpful and descriptive errors to their consumers, 4xx client errors, 5xx server errors
  + Apply Filters => Validation / Exception / Logging