**Essor it solutions Mohali**

**Difference between FirstOrDefault and LastOrDefault**

**---------------------------------------------------------------------------------------------------------------------**

**FirstOrDefault vs SingleOrDefault**

**---------------------------------------------------------------------------------------------------------------------**

**Single vs. SingleOrDefault**

**---------------------------------------------------------------------------------------------------------------------**

**Constructors**

Constructor(s) in a class is/are special methods which get called automatically when an object of a class is created. Constructors are specially used to initialize data members.

There are different types of constructors you can write in a class.

**Default Constructor:** When you do not declare any constructor, the class will call its default constructor which has a default public access modifier. The default constructor is a parameter less constructor which will be called by a class object.

Let's see an example of a Default Constructor -

|  |
| --- |
| public class Person  {  private int m\_PersonID;  private string m\_FirstName, m\_LastName, m\_City;  public Person()  {  m\_PersonID = 19929;  m\_FirstName = "No First Name";  m\_LastName = "No Last Name";  m\_City = "No City";  }  } |

This default constructor will be executed whenever the class is initialized – Person p = new Person();

### Note: If the class is abstract, then the accessibility of the default constructor is protected. Otherwise, the accessibility for the default constructor is public

### Parameterized constructors

You can also call it as constructor overloading. Default constructors always initialize the objects with the same values. In case you want to initialize the class with different values, you can use Parameterized constructors.

|  |
| --- |
| public class Person  {      private int m\_PersonID;      private string m\_FirstName, m\_LastName, m\_City;      public Person()      {          m\_PersonID = 19929;          m\_FirstName = "No First Name";          m\_LastName = "No Last Name";          m\_City = "No City";      }      public Person(string firstName, string lastName)      {          m\_FirstName = firstName;          m\_LastName = lastName;      }  } |

To invoke the parameterized constructor, use this Person p1 = new Person(“DotNet”, “Curry”);

**Copy Constructor**

Copy constructor is the parameterized constructor which takes a parameter of the same type. It allows you to initialize a new object with the existing object values.

|  |
| --- |
| public class Person  {      private int m\_PersonID;      private string m\_FirstName, m\_LastName, m\_City;      public Person()      {          m\_PersonID = 19929;          m\_FirstName = "No First Name";          m\_LastName = "No Last Name";          m\_City = "No City";      }      public Person(string firstName,string lastName)      {          m\_FirstName = firstName;          m\_LastName = lastName;      }        //copy constructor      public Person(Person person)      {          m\_PersonID = person.m\_PersonID;          m\_FirstName = person.m\_FirstName;          m\_LastName = person.m\_LastName;          m\_City = person.m\_City;      }  } |

Here’s an example:

|  |
| --- |
| // Instance constructor.  Person p1 = new Person(1, "DotNet", "Curry", "Pune");    // Copy Constructor  Person p2 = new Person(p1); |

**Static Constructors**

Static constructor is used to initialize the static data members of the class. Static constructor is only called once while creation of the first instance of the class. After that, no instance of a class will call the static constructor. You can also use static constructor to execute some code of the class which must be executed only once.

|  |
| --- |
| public class Person  {      static Person()      {          //Static Members      }  } |

In inheritance, the calling of the constructor starts from the parent class.

Let's see how to use these constructors -

|  |
| --- |
| static void Main(string[] args)  {      Person p1 = new Person();//This will call Default Constructor      Person p2 = new Person("Pravin", "D");//This will call two parameterized Constructor      Person p3 = new Person(p2);//This will call Copy Constructor  } |

It is worth mentioning that you can also create a **private constructor**, which is generally used in classes that contain static members only. If you create a private constructor, you cannot create an instance of a class.

|  |
| --- |
| class Person  {      // Private Constructor:      private Person() { }        ...    } |

# Private Constructors (C# Programming Guide)

A private constructor is a special instance constructor**. It is generally used in classes that contain static members only**. If a class has one or more private constructors and no public constructors, other classes (except nested classes) cannot create instances of this class. For example:

class NLog

{

// Private Constructor:

private NLog() { }

public static double e = Math.E; //2.71828...

}

The declaration of the empty constructor prevents the automatic generation of a default constructor. Note that **if you do not use an access modifier with the constructor it will still be private by default**. However, the [private](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/private) modifier is usually used explicitly to make it clear that the class cannot be instantiated.

Private constructors are used to prevent creating instances of a class when there are no instance fields or methods, such as the [Math](https://docs.microsoft.com/en-us/dotnet/api/system.math) class, or when a method is called to obtain an instance of a class. If all the methods in the class are static, consider making the complete class static. For more information see [Static Classes and Static Class Members](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/static-classes-and-static-class-members).

## Example

The following is an example of a class using a private constructor.

public class Counter

{

private Counter() { }

public static int currentCount;

public static int IncrementCount()

{

return ++currentCount;

}

}

class TestCounter

{

static void Main()

{

// If you uncomment the following statement, it will generate

// an error because the constructor is inaccessible:

// Counter aCounter = new Counter(); // Error

Counter.currentCount = 100;

Counter.IncrementCount();

Console.WriteLine("New count: {0}", Counter.currentCount);

// Keep the console window open in debug mode.

Console.WriteLine("Press any key to exit.");

Console.ReadKey();

}

}

// Output: New count: 101

Notice that if you uncomment the following statement from the example, it will generate an error because the constructor is inaccessible because of its protection level:

// Counter aCounter = new Counter(); // Error

**A class cannot be instantiated when it has Private constructor. Abstract class also serve the same purpose.**

**Private constructor class is used to create a specific creation pattern named "Singleton". This pattern enables to create only a single object of the class shares among various client applications. In real time, LoadBalancing applications using for sharing work load among different machines should be implemented with singleton. Because all servers should access the same instance to update It.**

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Interfaces**

An interface looks like a class, but has no implementation. The only thing it contains declarations of events, indexers, methods and/or properties. The reason interfaces only provide declarations is because they are inherited by structs and classes that must provide an implementation for each interface member declared.

**Abstract class vs interface**

| **Feature** | **Interface** | **Abstract class** |
| --- | --- | --- |
| Multiple inheritance | A class may inherit several interfaces. | A class may inherit only one abstract class. |
| Default implementation | An interface cannot provide any code, just the signature. | An abstract class can provide complete, default code and/or just the details that have to be overridden. |
| Access Modifiers | An interface cannot have access modifiers for the subs, functions, properties etc everything is assumed as **public** | An abstract class can contain access modifiers for the subs, functions, properties |
| Core VS Peripheral | Interfaces are used to define the peripheral abilities of a class. In other words both Human and Vehicle can inherit from an IMovable interface. | An abstract class defines the core identity of a class and there it is used for objects of the same type. |
| Speed | Requires more time to find the actual method in the corresponding classes. | Fast |
| Adding functionality (Versioning) | If we add a new method to an Interface then we have to track down all the implementations of the interface and define implementation for the new method. | If we add a new method to an abstract class then we have the option of providing default implementation and therefore all the existing code might work properly. |
| Fields and Constants | No fields can be defined in interfaces | An abstract class can have fields and constraints defined |

Abstract class must need not have abstract functions

Can abstract class inherit interface: Yes

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Datareader vs scalar vs dataset**

**DataReader**

DataReader is used to read the data from database and it is a read and forward only **connection oriented** architecture during fetch the data from database. DataReader will fetch the data very fast when compared with dataset. Generally we will use ExecuteReader object to bind data to datareader.

* Holds the connection open until you are finished (don't forget to close it!).
* Can typically only be iterated over once
* Is not as useful for updating back to the database

**DataSet**

DataSet is a **disconnected connection orient** architecture that means there is no need of active connections during work with datasets and it is a collection of DataTables and relations between tables. It is used to hold multiple tables with data. You can select data form tables, create views based on table and ask child rows over relations. Also DataSet provides you with rich features like saving data as XML and loading XML data.

**DataAdapter**

**DataAdapter acts as a Bridge between DataSet and database.** This dataadapter object is used to read the data from database and bind that data to dataset. **DataAdapter is a disconnected oriented architecture.**

* Lets you close the connection as soon it's done loading data, and may even close it for you automatically
* All of the results are available in memory
* You can iterate over it as many times as you need, or even look up a specific record by index
* Has some built-in faculties for updating back to the database.

**DataTable**  
DataTable represents a single table in the database. It has rows and columns. There is no much difference between dataset and datatable, dataset is simply the collection of datatables.

--------------------------------------------------------------------------------------------------------------------------------------------------------

**Connected vs disconnected approach**

The ADO.NET Framework supports two models of Data Access Architecture, Connection Oriented Data Access Architecture and Disconnected Data Access Architecture.

In Connection Oriented Data Access Architecture the application makes a connection to the Data Source and then interact with it through SQL requests using the same connection. In these cases the application stays connected to the database system even when it is not using any Database Operations.

ADO.Net solves this problem by introduces a new component called Dataset. The DataSet is the central component in the ADO.NET Disconnected Data Access Architecture. A DataSet is an in-memory data store that can hold multiple tables at the same time. DataSets only hold data and do not interact with a Data Source. One of the key characteristics of the DataSet is that it has no knowledge of the underlying Data Source that might have been used to populate it.

------------------------------------------------------------------------------------------------------

**Jquery vs JavaScript**

-----------------------------------------------------------------------------------------------------------------------------------------------

Get checked value of check boxes without using id or class

$('input[type=checkbox]:checked') will give you all the checked checkboxes.

-------------------------------------------------------------------------------------------------------------------------------------

**Custom paging syntax**

**A Simple usage syntax is as:**

CREATE PROCEDURE dbo.uspGetPageRecords  
(  
    @OffSetRowNo INT,       
    @FetchRowNo INT  
)   
AS

SELECT colName1, colName2, colName3, colName4 FROM tblMyTableName  
ORDER BY colNameForSorting  
OFFSET ( @OffSetRowNo-1 ) \* @FetchRowNo ROWS  
FETCH NEXT @FetchRowNo ROWS ONLY

GO

**A  simple example is:**

CREATE PROCEDURE dbo.uspGetPageRecords  
(  
    @PageNo INT,      
    @RecordsPerPage INT  
)

AS

SET NOCOUNT ON;

--The offset specified in a OFFSET clause may not be negative.

--So check & set the initial for avoiding negative OFFSET

IF @PageNo < 1

       SET @PageNo = 1

SELECT AutoID, Name, City, RegistrationDate FROM tblUserMaster

ORDER BY RegistrationDate

OFFSET ( @PageNo - 1 ) \* @RecordsPerPage ROWS

FETCH NEXT @RecordsPerPage ROWS ONLY  
  
GO

-------------------------------------------------------------------------------------------------------------------------------------

**Curser in sql server**

A Cursor is a SQL Object, or we can say like a Virtual table that retrieves data from the table one row at a time. We use cursors when we need to update records in a database table in singleton fashion in other words row by row.  
  
**Life Cycle of cursor:**

1. Declare
2. Open
3. Fetch
4. Close
5. Deallocate

Before using a cursor, you first must declare the cursor. Once a cursor has been declared, you can open it and fetch from it. You can fetch row by row and make multiple operations on the currently active row in the cursor. When you have finished working with a cursor, you should close the cursor and deallocate it to release SQL Server resources.  
Declare @Username varchar(200)

Declare @ID int

Declare cur\_Test cursor

Static for

Select top 10 username, id from [dbo].[tblUser]

open cur\_Test

If @@CURSOR\_ROWS>0

Begin

Fetch Next from cur\_Test into @UserName,@Id

while @@Fetch\_Status=0

begin

Print 'User Name : '+ @UserName+ ' / User ID : '+Convert(nvarchar(200),@Id)

End

End

Close cur\_Test

Deallocate cur\_Test

**Type of cursor:**

1. Forward Only Cursor
2. Scroll Cursor
3. Static Cursor
4. Dynamic Cursor
5. Keyset Driven Cursor

**--------------------------------------------------------------------------------------------------------**

Left join vs right join

Pivot table in sql server

Code first vs database first

**Code first**

* Very popular because hardcore programmers don't like any kind of designers and defining mapping in EDMX xml is too complex.
* Full control over the code (no auto generated code which is hard to modify).
* General expectation is that you do not bother with DB. DB is just a storage with no logic. EF will handle creation and you don't want to know how it does the job.
* Manual changes to database will be most probably lost because your code defines the database.

**Database first**

* Very popular if you have DB designed by DBAs, developed separately or if you have existing DB.
* You will let EF create entities for you and after modification of mapping you will generate POCO entities.
* If you want additional features in POCO entities you must either T4 modify template or use partial classes.
* Manual changes to the database are possible because the database defines your domain model. You can always update model from database (this feature works quite good).

**Model first**

* IMHO popular if you are designer fan (= you don't like writing code or SQL).
* You will "draw" your model and let workflow generate your database script and T4 template generate your POCO entities. You will lose part of the control on both your entities and database but for small easy projects you will be very productive.
* If you want additional features in POCO entities you must either T4 modify template or use partial classes.
* Manual changes to database will be most probably lost because your model defines the database. This works better if you have Database generation power pack installed. It will allow you updating database schema (instead of recreating) or updating database projects in VS.

----------------------------------------------------------------------------------

Project management tools used

Code management tools

Why switching the company

**Artesian software technologies Mohali**

Select query with record id is too slow for a table which has lakhs of records, how you fasten it.

Select only needed column, user index,

------------------------------------------------------------------------------------

Indexes in sql server

Clustered and Non Clustered

Index provides a fast way to look up data based on the values provide in where clause those columns. In SQL Server there is a system table with the name sysindexes that content information about indexes are available on tables in the database. A table even has no index, there will be one row in the sysindexes table related to that table indicating there is no index on the table.

For example, if you create an index on the primary key and then search for a row of data based on one of the primary key values, SQL Server first finds that value in the Sysindexs and if it have in, and then uses the index to quickly locate the entire row of data. Without the index, a table scan (process checks the given condition with each and every row of the table called a table scan) would have to be performed in order to locate the row, which can have a significant effect on performance.

Clustered Indexes

A clustered index stores the actual data rows at the leaf level of the index. An important characteristic of the clustered index is that the indexed values are sorted in either ascending or descending order. As a result, there can be only one clustered index on a table or view. In addition, data in a table is sorted only if a clustered index has been defined on a table.

Note: A table that has a clustered index is referred to as a clustered table. A table that has no clustered index is referred to as a heap.

Nonclustered Indexes

Unlike a clustered indexed, the leaf nodes of a nonclustered index contain only the values from the indexed columns and row locators that point to the actual data rows, rather than contain the data rows themselves. This means that the query engine must take an additional step in order to locate the actual data.

A row locator’s structure depends on whether it points to a clustered table or to a heap. If referencing a clustered table, the row locator points to the clustered index, using the value from the clustered index to navigate to the correct data row. If referencing a heap, the row locator points to the actual data row.

Nonclustered indexes cannot be sorted like clustered indexes; however, you can create more than one nonclustered index per table or view. SQL Server 2005 supports up to 249 nonclustered indexes, and SQL Server 2008 support up to 999. This certainly doesn’t mean you should create that many indexes. Indexes can both help and hinder performance.

SYNTEX:

CREATE INDEX index\_nameON table\_name (column1, column2, ...);

CREATE INDEX idx\_lastnameON Persons (LastName);// or column Names if want to add multiple columns

------------------------------------------------------------------------------------------------------------------------------------------

IQueryable vs IEnumrable

IEnumerable

IEnumerable exists in the System.Collections namespace.

IEnumerable is suitable for querying data from in-memory collections like List, Array and so on.

While querying data from the database, IEnumerable executes "select query" on the server-side, loads data in-memory on the client-side and then filters the data.

IEnumerable is beneficial for LINQ to Object and LINQ to XML queries.

IQueryable

IQueryable exists in the System.Linq Namespace.

IQueryable is suitable for querying data from out-memory (like remote database, service) collections.

While querying data from a database, IQueryable executes a "select query" on server-side with all filters.

IQueryable is beneficial for LINQ to SQL queries.

-------------------------------------------------------------------------------------------------------

How to show 100000 record in ui

-------------------------------------------------------------------------------------------------------

What is used for paging in linq

How to do paging from databases

-------------------------------------------------------------------------------------------------------

Cte in sql server

For any operation over a temporary result set, SQL Server has to offer Temp table, a table variable, etc. But sql server also provides a very powerful feature: Common Table Expression (CTE).

CTE is a temporary result set derived from the underling definition.

We have a simple table Products in our database.

Select \* from PRODUCTS

ProductID ProductDesc ManufacturingDate ExpiryDate IsSalable Price

---------------------------------------------------------------------------------------

1 Biscuits 2011-09-01 00:00:00.000 2012-09-01 00:00:00.000 1 20.00

2 Butter 2010-09-01 00:00:00.000 2011-09-01 00:00:00.000 1 30.00

3 Milk 2011-10-01 00:00:00.000 2011-11-01 00:00:00.000 1 46.00

We have created a simple ProductsCTE for displaying all the Products with Price greater than 20.00. Here CTE performs the job of acting as a simple derived table.

;WITH ProductsCTE(ProdName,Price) AS

( SELECT ProductDesc,Price

FROM PRODUCTS

WHERE Price>20.00

)

SELECT \* FROM ProductsCTE

ProdName Price

-------------------------------------------------- ---------------------

Butter 30.00

Milk 46.00

Important point that needs a mention is the SELECT followed by the CTE definition, any operation SELECT, INSERT, UPDATE, DELETE or Merge can be performed immediately after the CTE and the CTE lasts only for a single such operation.

It means that the below code is erroneous:

DECLARE @T INT,@I INT

SET @T = 10

SET @I = 20

;WITH ProductsCTE(ProdName,Price) AS

( SELECT ProductDesc,Price

FROM PRODUCTS

WHERE Price>20.00

)

SELECT @T+@I

SELECT \* FROM ProductsCTE

On executing the code, it results in the below error. Which means I need to execute the ProductsCTE SELECT immediately after the CTE definition completes.

Msg 422, Level 16, State 4, Line 10

Common table expression def

---------------------------------------------------------------------------------------------------------------------------------------

Why we create apis

----------------------------------------------------------------------------------------------------------------------------------

Exception handling

An exception is a problem that arises during the execution of a program. A C# exception is a response to an exceptional circumstance that arises while a program is running, such as an attempt to divide by zero.

Exceptions provide a way to transfer control from one part of a program to another. C# exception handling is built upon four keywords: try, catch, finally, and throw.

try − A try block identifies a block of code for which particular exceptions is activated. It is followed by one or more catch blocks.

catch − A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.

finally − The finally block is used to execute a given set of statements, whether an exception is thrown or not thrown. For example, if you open a file, it must be closed whether an exception is raised or not.

throw − A program throws an exception when a problem shows up. This is done using a throw keyword.

------------------------------------------------------------------------------------------------------

Tools used for creating logs. Eg nlog. C# Logger, Logger.NET

--------------------------------------------------------------------------------------------------------------

Dependency injection

SignalR. How it works

Message queues. Msmq

Azure

Left outer join vs inner join

--------------------------------------------------------------------------------------------------------

A table has messages of different users. I want latest message of all user

SELECT t1.\* FROM messages t1

JOIN (SELECT from\_id, MAX(timestamp) timestamp FROM messages GROUP BY from\_id) t2

ON t1.from\_id

-------------------------------------------------------------------------------------------------------------

Foreign id can be null: Yes

Primary id can be null: No

Bundling

-------------------------------------------------------------------------------------------------------------------

lazy loading

Lazy Loading: It is the default behavior of an Entity Framework, where a child entity is loaded only when it is accessed for the first time. It simply delays the loading of the related data, until you ask for it.

For example, when we run the query given below, UserDetails table will not be loaded along with the User table.

User usr = dbContext.Users.FirstOrDefault(a => a.UserId == userId);

It will only be loaded when you explicitly call for it, as shown below.

UserDeatils ud = usr.UserDetails; // UserDetails are loaded here

Eager Loading: Eager Loading helps you to load all your needed entities at once; i.e., all your child entities will be loaded at single database call. This can be achieved, using the Include method, which returs the related entities as a part of the query and a large amount of data is loaded at once.

For example, you have a User table and a UserDetails table (related entity to User table), then you will write the code given below. Here, we are loading the user with the Id equal to userId along with the user details.

User usr = dbContext.Users.Include(a => a.UserDetails).FirstOrDefault(a => a.UserId == userId);

If you have multiple level of child entities, then you can load, using the query given below.

User usr = dbContext.Users.Include(a => a.UserDetails.Select(ud => ud.Address)).FirstOrDefault(

Explicit Loading: There are options to disable Lazy Loading in an Entity Framework. After turning Lazy Loading off, you can still load the entities by explicitly calling the Load method for the related entities. There are two ways to use Load method Reference (to load single navigation property) and Collection (to load collections), as shown below.

User usr = dbContext.Users.FirstOrDefault(a => a.UserId == userId);

dbContext.Entry(usr).Reference(usr => usr.UserDetails).Load();

When to use what

Use Eager Loading when the relations are not too much. Thus, Eager Loading is a good practice to reduce further queries on the Server.

Use Eager Loading when you are sure that you will be using related entities with the main entity everywhere.

Use Lazy Loading when you are using one-to-many collections.

Use Lazy Loading when you are sure that you are not using related entities instantly.

When you have turned off Lazy Loading, use Explicit loading when you are not sure whether or not you will be using an entity beforehand.

--------------------------------------------------------------------------------------------------------------------------------------

test case tdd

test case tdd

service oriented architect- soa

micro services

dependency injection

extension methods in mvc

oath

benefits of knockoutjs

Second round

difference between mvc and .net core

web api

how to right api in mvc

[www.abc.com/api/hello/adeep](http://www.abc.com/api/hello/adeep). Output should be hello adeep. How

Routing. Write rout for above

How to return xml from mvc controller/action result. In future if I create my own language and want to return data in that language. How. Eg. In Ajax content type is used=contentResult

How many ways there to fetch data from database. Hint connected and disconnected

How many ways are there to fetch data from database in entity framework?

We have teacher table with id and name. Student table with id and name. Many to many relationship. Create relation in database and create everything to map in edmx file.

@@identity

-----------------------------------------------------------------------------------------------------------------------------------

Antier software solutions Mohali

Mvc

Html helpers

Routing

Minification

Return type of action result

-----------------------------------------------------------------------------------------------------------------------

How to create a function not accessible in mvc

By default, the MVC framework treats all public methods of a controller class as action methods. If your controller class contains a public method and you do not want it to be an action method, you must mark that method with the NonActionAttribute attribute.

-----------------------------------------------------------------------

How to use multiple models in view. tupple

Filters

Web api

How to update html of particular section of page after some interval or on hitting a button

There is Ajax function which will work for particular button in mvc.

Get third max salary

Second round

How you can increase the performance of database at initial stage of application and also in live application.

Initial state: Columns Size, Datatype, Normalization, Indexes, schema

Live Application: Column size, Index,

Triggers in sql

When trigger fires

Can we create single trigger for insert and update(Y) or update and delete(N).

Benefits of mvc

Action results

Type of action results

Is mvc uses view state

Difference between server side and client side controls

Mvc uses which type of controls

Life cycle of mvc application

How controller knows that view is updated. Hint ismodelupdate

Authorization

Identity and oath

How we can made our site secure

Token, expiry limit, can we increate the expiry limit

Refresh tokens

Test first approach

Tdd

**TMotions Mohali**

1. Data Structure: **data structure** is a particular way of organizing and storing [data](https://en.wikipedia.org/wiki/Data_(computing)) in a computer so that it can be accessed and modified [efficiently](https://en.wikipedia.org/wiki/Algorithmic_efficiency). Different types of data structures are:-  
   Stack- Works in first in last out order. The element inserted first in stack is removed last.  
   Queue- First in First out order. The element inserted first is removed first.  
   **Lists:** A group of similar items with connectivity to the previous or/and next data items.  
   **Arrays:** A set of homogeneous values  
   **Records:** A set of fields, where each field consists of data belongs to one data type.  
   **Trees:** A data structure where the data is organized in a hierarchical structure. This type of data structure follows the sorted order of insertion, deletion and modification of data items.  
   **Tables:** Data is persisted in the form of rows and columns. These are similar to records, where the result or manipulation of data is reflected for the whole table.

-------------------------------------------------------------------------------------------------------------------------------------------------

1. Stack & Heap and LIFO & FIFO: Stack is used for static memory allocation and Heap for dynamic memory allocation, both stored in the computer's RAM.

Variables allocated on the stack(Value type i.e. int) are stored directly to the memory and access to this memory is very fast, and it's allocation is dealt with when the program is compiled. The stack is always reserved in a LIFO order,

Variables allocated on the heap(Reference type, i.e. objects) have their memory allocated at run time and accessing this memory is a bit slower, but the heap size is only limited by the size of virtual memory . Element of the heap have no dependencies with each other and can always be accessed randomly at any time. You can allocate a block at any time and free it at any time. This makes it much more complex to keep track of which parts of the heap are allocated or free at any given time

----------------------------------------------------------------------------------

1. MVC structure

--------------------------------------------------------------------------------------------------------------------------------------------

1. Return types in MVC

-----------------------------------------------------------------------------------------------------------------------------------------------

# Action Result Return Type in MVC 4

ActionResult is a return type of a controller method in ASP.NET MVC. It help us to return models to views, other return value, and also redirect to another controller’s action method. There are many derived ActionResult types in MVC that we use to return the result of a controller method to the view.

## ****What is ActionResult ?****

* An ActionResult is a return type of a controller method in MVC.
* ActionResult are more specific for a particular view
* It is abstract class that has many subtypes

## ****Types of ActionResult :****

* ViewResult
* PartialViewResult
* ContentResult
* RedirectResult
* RedirectToRouteResult
* JsonResult
* EmptyResult
* FileResult
* JavaScriptResult

## ****ViewResult :****

* It renders a specified view to the response stream.

**Example :**

public ViewResult Index() {   return View(“sampleResult”); }

## ****PartialViewResult:****

* Renders a specifed partial view to the response stream.

**Example :**

Public PartialViewResult SamplePartialView()  
{    return PartialView(“\_LoadSamplePartial”); }

**Note :**

* It will load the partial view to the main view

## ****ContentResult :****

* It display the response stream without requiring a view .(like a plain text).

**Example**

public ActionResult About()  
{  
ViewBag.Message = “Your app description page.”;  
return Content(“Sample Content Action”);  
}

It simply write the string value in the page

## ****RedirectResult**** :

* It is used to perform an HTTP redirect to a given URL.

**Example :**

public ActionResult SampleRedirectResult()  
{  return Redirect(“http://www.google.com/”); }

## ****RedirectToRouteResult :****

* RedirectToResult is used to redirect by using the specified route values dictionary.

**Example :**

public ActionResult SampleRedirectToRouteResult()  
{  
return new RedirectToRouteResult(new System.Web.Routing.RouteValueDictionary(new  
{  
controller = “Sample”,  
action = “SampleRedirectToRouteResultTest”,  
Id = new int?()  
}));  
}

## ****JsonResult :****

* Action methods return JsonResult (JavaScript Object Notation result) that can be used in ajax based application.

**Example :**

public ActionResult SampleJsonResult  
{  
… Logic here  
return Json(sampleinfo);  
}

## ****EmptyResult :****

* It return  NULL that is EmptyResult
* return instance of EmptyResult class

**Example :**

return new EmptyResult();

## ****FileResult :****

* Return a file from an action. (Pdf,Excel,image file,Html…)

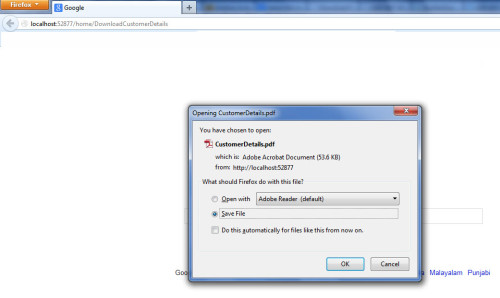
**Example :**

public FileResult DownloadCustomerDetails()  
{  
//Parameters to file are  
//1. The File Path on the File Server  
//2. The content type MIME type  
//3. Name of file To be save by Browser  
string contentType = “application/pdf”;  
string filepath = AppDomain.CurrentDomain.BaseDirectory + “/FileFolder/CustomerDetails.pdf”;  
return File(filepath, contentType, “CustomerDetails.pdf”);

## ****Output:****

After calling the action method in browser , the file will ask for download as shown in below

http://localhost:52877/home/DownloadCustomerDetails

}[](http://dotnethelpers.files.wordpress.com/2014/01/actionreturntype3.jpg)

## ****JavaScriptResult:****

* To improve clean separation of concerns by introducing the JavaScriptResult ActionResult.
* Release notes says, “the JavaScriptResult class is used to execute JavaScript code that is created on the server and sent to the client.”

**For Example,**

**Controller :**

public JavaScriptResult SampleJavaScriptResult()  
{  
var s = “alert(‘Hi’)”;  
return JavaScript(s);  
}

**View :**

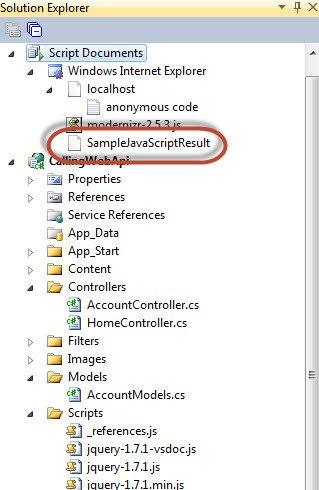
<script type=”text/javascript” src=”@Url.Content(“~/Home/SampleJavaScriptResult”)”></script>

Note :

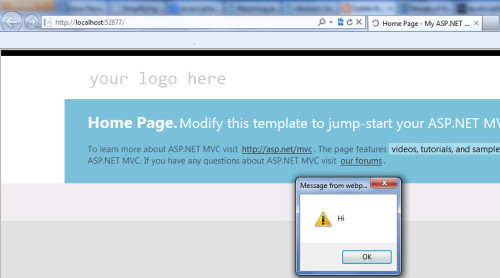
The dynamic script file will be generated based on the action name (SampleJavaScriptResult)

## ****Output:****

After press F5, it will create the dynamic script file “SampleJavaScriptResult” which will append the content from action

[](http://dotnethelpers.files.wordpress.com/2014/01/actionreturntype2.jpg?resize=319,490)

Then it will display the alert message which was return from the action “SampleJavaScriptResult” as show in below

[](http://dotnethelpers.files.wordpress.com/2014/01/actionreturntype1.jpg)

------------------------------------------------------------------------------------------------------------------------------------------------

1. View vs Partial view: A partial view is a view that's rendered within another view. The HTML output generated by executing the partial view is rendered into the calling (or parent) view. Like views, partial views use the .cshtml file extension.

Partial views are an effective way of breaking up large views into smaller components. They can reduce duplication of view content and allow view elements to be reused. Common layout elements should be specified in [\_Layout.cshtml](https://docs.microsoft.com/en-us/aspnet/core/mvc/views/layout?view=aspnetcore-2.0). Non-layout reusable content can be encapsulated into partial views.

----------------------------------------------------------------------------------------------------------------------------

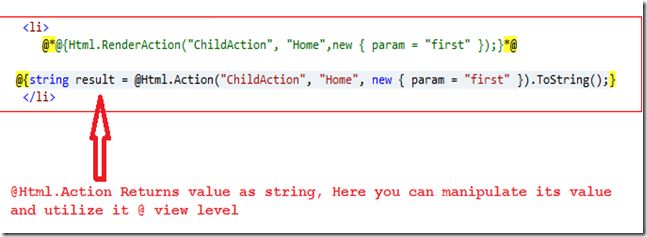
1. Html.Partial and Render Partial:

**Html.RenderPartial**  
1. This method result will be directly written to the HTTP response stream means it used the same TextWriter object as used in the current webpage/template.  
2. This method returns void.  
3. Simple to use and no need to create any action.  
4. RenderPartial method is useful when the displaying data in the partial view is already in the corresponding view model. For example: In a blog to show comments of an article, we would like to use RenderPartial method since an article information with comments are already populated in the view model.  
@{Html.RenderPartial("\_Comments");}  
5. This method is faster than Partial method since its result is directly written to the response stream which makes it fast.  
  
**Html.Partial**  
1. Renders the partial view as an HTML-encoded string.  
2. This method result can be stored in a variable, since it returns string type value.  
3. Simple to use and no need to create any action.  
4. Like RenderPartial method, Partial method is also useful when the displaying data in the partial view is already in the corresponding view model. For example: In a blog to show comments of an article, you can use Partial method since an article information with comments are already populated in the view model.  
@Html.Partial("\_Comments")

--------------------------------------------------------------------------------------------------------

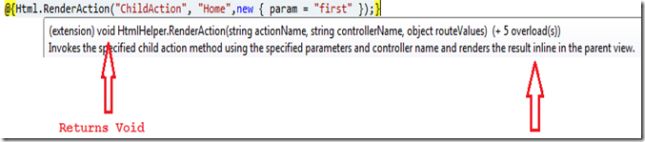
1. Html.Action and Html.RenderAction: **@Html.Action():** Invokes the specified child action method and returns the result as an HTML string.   
     
   The way to call an action via RenderAction is shown below:
2. @Html.Action("ChildAction", "Home", new { param = "first" })

This method result can be stored in a variable, since it returns a string type value. Kindly look at the image shown below:

  
  
**@{ Html.RenderAction():** Invokes the specified child action method and renders the result inline in the parent view.   
  
This method is more efficient if the action returns a large amount of HTML.   
  
The way to call an action via RenderAction is shown below:

1. @{Html.RenderAction("ChildAction", "Home",new { param = "first" });}

It returns Voids and renders/gives a result directly to the response.

  
  
Both methods are also used for rendering the partial view using the Child Action.  
  
The difference between the two is that Html.RenderAction will render the result directly to the Response (which is more efficient if the action returns a large amount of HTML) whereas Html.Action returns a string with the result.   
  
This method is faster than the Action method since its result is directly written to the HTTP response stream.

---------------------------------------------------------------------------------------------------------------------------------------  
Filters in MVC: **Filters in MVC:**

In ASP.NET MVC, user request is routed to the appropriate controller and action method. However, there may be circumstances where you want to execute some logic before or after an action method executes. ASP.NET MVC provides filters for this purpose.

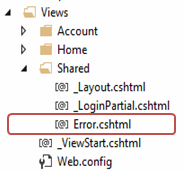
ASP.NET MVC Filter is a custom class where you can write custom logic to execute before or after an action method executes. Filters can be applied to an action method or controller in a declarative or programmatic way. Declarative means by applying a filter attribute to an action method or controller class and programmatic means by implementing a corresponding interface.

MVC provides different types of filters.

| **Filter Type** | **Description** | **Built-in Filter** | **Interface** |
| --- | --- | --- | --- |
| Authorization filters | Performs authentication and authorizes before executing action method. | [Authorize], [RequireHttps] | IAuthorizationFilter |
| Action filters | Performs some operation before and after an action method executes. |  | IActionFilter |
| Result filters | Performs some operation before or after the execution of view result. | [OutputCache] | IResultFilter |
| Exception filters | Performs some operation if there is an unhandled exception thrown during the execution of the ASP.NET MVC pipeline. | [HandleError] | IExceptionFilter |

To understand the filter in detail, let's take an example of built-in Exception filter.

An exception filter executes when there is an unhandled exception occurs in your application. HandleErrorAttribute ([HandlerError]) class is a built-in exception filter class in MVC framework. This built-in HandleErrorAttribute class renders Error.cshtml included in the Shared folder by default, when an unhandled exception occurs.

[](http://www.tutorialsteacher.com/Content/images/mvc/errorpage.png)

The following example demonstrates built-in exception filter HandErrorAttribute.

Authorization filter example:

[HandleError]

public class HomeController : Controller

{

public ActionResult Index()

{

//throw exception for demo

throw new Exception("This is unhandled exception");

return View();

}

public ActionResult About()

{

return View();

}

public ActionResult Contact()

{

return View();

}

}

In the above example, we have applied [HandleError] attribute to HomeController. So now it will display Error page if any action method of HomeController would throw unhandled exception. Please note that unhandled exception is an exception which is not handled by the try-catch block.

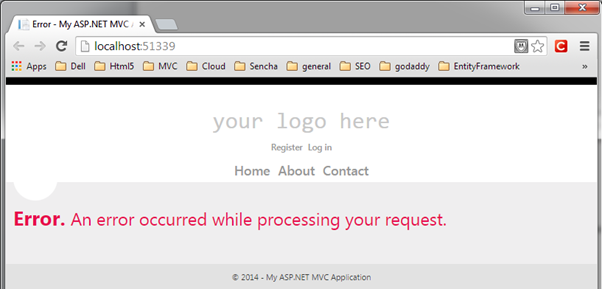
Filters applied to the controller will automatically be applicable to all the action methods of a controller.

Please make sure that CustomError mode is on in System.web section of web.config, in order for HandleErrorAttribute work properly.

CustomError in web.config:

<customErrors mode="On" />

Now, if you run the application. You would get following error page because we throw exception in Index action method for the demo purpose.

[](http://www.tutorialsteacher.com/Content/images/mvc/handleerror-demo.png)HandleError Demo

Thus, HandleError attribute will display common error page for any unhandled exception occurred in HomeController.

## Register Filters:

Filters can be applied at three levels.

### 1. Global Level:

You can apply filters at global level in the Application\_Start event of Global.asax.cs file by using default FilterConfig.RegisterGlobalFilters() mehtod. Global filters will be applied to all the controller and action methods of an application.

The [HandleError] filter is applied globaly in MVC Application by default in every MVC application created using Visual Studio as shown below.

Example: Register Global level Filters

// MvcApplication class contains in Global.asax.cs file

public class MvcApplication : System.Web.HttpApplication

{

protected void Application\_Start()

{

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

}

}

// FilterConfig.cs located in App\_Start folder

public class FilterConfig

{

public static void RegisterGlobalFilters(GlobalFilterCollection filters)

{

filters.Add(new HandleErrorAttribute());

}

}

### 2. Controller level:

Filters can also be applied to the controller class. So, filters will be applicable to all the action method of Controller class if it is applied to a controller class.

Example: Action Filters to Controller

[HandleError]

public class HomeController : Controller

{

public ActionResult Index()

{

return View();

}

}

### 3. Action method level:

You can apply filters to an individual action method also. So, filter will be applicable to that particular action method only.

Apply Filters to Action method:

public class HomeController : Controller

{

[HandleError]

public ActionResult Index()

{

return View();

}

}

The same way, you can apply multiple built-in or custom filters globally or at controller or action method level for different purpose such as [Authorize],[RequireHttps], [ChildActionOnly],[OutputCache],[HandleError].

## Filter Order:

As mentioned above, MVC includes different types of filters and multiple filters can be applied to a single controller class or action method. So, filters run in the following order.

1. Authorization filters
2. Action filters
3. Response filters
4. Exception filters

## Create Custom Filter:

You can create custom filter attributes by implementing an appropriate filter interface for which you want to create a custom filter and also derive a FilterAttribute class so that you can use that class as an attribute.

For example, implement IExceptionFilter and FilterAttribute class to create custom exception filter. In the same way implement an IAuthorizatinFilter interface and FilterAttribute class to create a custom authorization filter.

Example: Custom Exception Filter

class MyErrorHandler : FilterAttribute, IExceptionFilter

{

public override void IExceptionFilter.OnException(ExceptionContext filterContext)

{

Log(filterContext.Exception);

base.OnException(filterContext);

}

private void Log(Exception exception)

{

//log exception here..

}

}

Alternatively, you can also derive a built-in filter class and override an appropriate method to extend the functionality of built-in filters.

Let's create custom exception filter to log every unhandled exception by deriving built-in HandleErrorAttribute class and overriding OnException method as shown below.

Example: Custom Exception Filter

class MyErrorHandler : HandleErrorAttribute

{

public override void OnException(ExceptionContext filterContext)

{

Log(filterContext.Exception);

base.OnException(filterContext);

}

private void Log(Exception exception)

{

//log exception here..

}

}

Now, you can apply MyErrorHandler attribute at global level or controller or action method level, the same way we applied the HandleError attribute.

Example: Custom Action Filters to Controller

[MyErrorHandler]

public class HomeController : Controller

{

public ActionResult Index()

{

return View();

}

}

### Points to Remember :

1. MVC Filters are used to execute custom logic before or after executing action method.
2. MVC Filter types:
   1. Authorization filters
   2. Action filters
   3. Result filters
   4. Exception filters
3. Filters can be applied globally in FilterConfig class, at controller level or action method level.
4. Custom filter class can be created by implementing FilterAttribute class and corresponding interface.

----------------------------------------------------------------------------------------------

1. Output Cache:

The main purpose of using Output Caching is to dramatically improve the performance of an ASP.NET MVC Application. It enables us to cache the content returned by any controller method so that the same content does not need to be generated each time the same controller method is invoked. Output Caching has huge advantages, such as it reduces server round trips, reduces database server round trips, reduces network traffic etc.

[OutputCache(Duration = 10, VaryByParam = "none", Location = System.Web.UI.OutputCacheLocation.Client)]

public ActionResult Index()

------------------------------------------------------------------------------------------------------------------

1. Routing in MVC
2. HTML Helpers in MVC
3. Razor in MVC
4. Scaffolding
5. Bundling and Magnification
6. Web API (Difference in Web API1 and Web API2)
7. Advantages of Web Api2
8. OData
9. CORS:

(CORS) is a W3C standard that allows a server to relax the same-origin policy. Using CORS, a server can explicitly allow some cross-origin requests while rejecting others. CORS is safer and more flexible than earlier techniques such as [JSONP](http://en.wikipedia.org/wiki/JSONP).

To enable CORS in .net we have to include CORS NuGet package and then enable it in WebApiConfig file and use attribute over controller in which you want it to enable

[EnableCors(origins: "http://mywebclient.azurewebsites.net", headers: "\*", methods: "\*")]

1. OWIN
2. External Authentication in Web Api
3. Difference between MVC5 and MVC6
4. Clouding Computing
5. Different type of service on Cloud
6. Services of which manages resource on Azure
7. Which type of web application we can deploy on Azure. Hint: Php application or asp.net application
8. .Net Core
9. Polymorephism
10. Overloading Vs Overriding
11. Virtual function
12. Modifiers. What is new modifier? Any example where we can use new modifier
13. Sealed Modifier in C#
14. Interface vs abstract class. Can abstract class have sealed Method. Abstract class without abstract method. If yes give Live Exaple
15. Default Modifier of the class. Internal vs private
16. Lazy Loading. Can we turn loazy loading on/Off
17. ADO.net vs Entity Framework
18. T4 template in MVC
19. POCO classer Hint POCO and LOCO classes are there in EF
20. Truncate vs Delete
21. CTE. Syntax
22. Triggers
23. Can I call trigger from my SP
24. Can I call SP from Trigger
25. Front Languages. Hint Angular, Knockout
26. I which scenario we can use JQuery
27. Selectors in JQuery
28. $.each
29. JQuery Load
30. What is KnockoutJs
31. Knockout VM
32. Can you have multiple VM on single page. Yes
33. File sharing tools : TFC, Git
34. How can we deploy new application
35. Automated deployment
36. Blob Storage

**OneBCG**

1. What challenging done in project
2. why shifting company
3. Why leave previous companies
4. Custom paging in Sql, what is in Linq for it
5. Difference between where & Having. Where can not be used with aggregate function like having, count. So having is used in place of where to apply conditions.
6. Interface
7. Default Access specfier of interface. : Public
8. Default Access specfier of Class: internal
9. Default access specifier of constructor : Private
10. Interface and Class name can be same: It's not possible to have two types with the same name under the same namespace (this is one reason why namespaces are useful). A good convention is to have the letter I before interface names
11. Disposable method. How we do it
12. Design Pattern
13. What is repository
14. How you initialize interface
15. Why interface initialize with class
16. Dependency Injection
17. Joins
18. Employee table, Department table, require DepartmentName, Employee count, Max salary
19. Partial Render and Html.Partial and Html.RenderPartail.
20. Partial view in Modal
21. You have sent user email many times. What you do if you want to do same functionality in different projects
22. Cloud
23. How to solve a problem
24. If you are doing same thing in different projects then what is the good way to use it
25. What different roles you performed in your carrier
26. How you estimate the task and assign to the team
27. Most complex functionality of your project according to you which you achieved
28. achievements in carriers
29. apart from technologies what you learn by your own
30. How you decide how much time the project takes
31. I already have candidate who knows much technologies then you like angular, then why I choose you
32. what you know about Our company
33. what are your long term goals

BD

EClerx

1. can we pass parametersied constructor in mvc.
2. What we have to do use parametersied constrcutr
3. what is class
4. What is Interface
5. Abstract Class
6. SOLID Principles

SQL

1. Steps to optimize database for new and running appoicaton
2. Index, why we need it
3. What are functions .
4. How we can know which things are slowing down our database
5. How to optimize as SP
6. Delete duplicte records from table using CTE
7. CharIndex:

C

1. Code first Approach in MVC
2. What happen I made change in database in database first and in code first
3. When to use code first and when to use database first approach
4. Polymorephism
5. Virtual functions
6. Webservice
7. webservice vs. wcf, web api

## Web Service

* It is based on SOAP and return data in XML form.
* It support only HTTP protocol.
* It is not open source but can be consumed by any client that understands xml.
* It can be hosted only on IIS.

## WCF

* 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 applicaion or on IIS or using window service.

## WCF Rest

* To use WCF as WCF Rest service you have to enable webHttpBindings.
* It support 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 support XML, JSON and ATOM data format.

## Web API

* 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.

**To 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.

1. soap vs. rest

SOAP (Simple Object Access Protocol) and REST (Representational State Transfer) are both web service communication protocols. SOAP was long the standard approach to web service interfaces, although it’s been dominated by REST in recent years, with REST now representing more than 70% of public APIs.

## SOAP vs. REST: Primary Differences

REST operates through a interface to access named resources. It’s most commonly used when you’re exposing a public API over the Internet. SOAP, on the other hand, exposes components of application logic as services rather than data. Additionally, it operates through different interfaces. To put it simply, REST accesses data while SOAP performs operations through a more [standardized set](http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/) of messaging patterns. Still, in most cases, either REST or SOAP could be used to achieve the same outcome (and both are infinitely scalable), with some differences in how you’d configure it.

SOAP was originally created by Microsoft, and it’s been around a lot longer than REST. This gives it the advantage of being an established, legacy protocol. But REST has been around for a good time now, as well. Plus, it entered the scene as a way to access web services in a much simpler way than possible with SOAP by using HTTP.

## Benefits of REST Over SOAP

In addition to using HTTP for simplicity, REST offers a number of other benefits over SOAP:

* **REST allows a greater variety of data formats**, whereas SOAP only allows XML.
* Coupled with JSON (which typically works better with data and offers faster parsing), **REST is generally considered easier to work with**.
* Thanks to JSON, **REST offers better support for browser clients**.
* **REST provides superior performance**, particularly through caching for information that’s not altered and not dynamic.
* **REST is the protocol used most often for major services** such as Yahoo, Ebay, Amazon, and even Google.
* **REST is generally faster and uses less bandwidth**. It’s also easier to integrate with existing websites with no need to refactor site infrastructure. This enables developers to work faster rather than spend time rewriting a site from scratch. Instead, they can simply add additional functionality.

Still, SOAP remains the preferred protocol for certain use cases. The general consensus among experts these days is that REST is the typically preferred protocol unless there’s a compelling reason to use SOAP (and there are some cases in which SOAP is preferred).

## Benefits of SOAP Over REST

Because you can achieve most outcomes using either protocol, it’s sometimes a matter of personal preference. However, there are some use cases for which SOAP tends to be better-suited. For instance, if you need more robust(strong) security, SOAP’s support for WS-Security can come in handy. It offers some additional assurances for data privacy and integrity. It also provides support for identity verification through intermediaries rather than just point-to-point, as provided by SSL (which is supported by both SOAP and REST).

**Another advantage of SOAP is that it offers built-in retry logic to compensate for failed communications. REST, on the other hand, doesn’t have a built-in messaging system.** If a communication fails, the client has to deal with it by retrying. There’s also no standard set of rules for REST. This means that both parties (the service and the consumer) need to understand both content and context.

Other benefits of SOAP include:

* **SOAP’s standard HTTP protocol makes it easier for it to operate across firewalls and proxies** [without modifications](http://searchmicroservices.techtarget.com/tip/REST-vs-SOAP-Choosing-the-best-web-service) to the SOAP protocol itself. But because it uses the complex XML format, it tends to be slower compared to middleware such as ICE and COBRA.
* Additionally, while it’s rarely needed, some use cases require greater transactional reliability than what can be achieved with HTTP (which limits REST in this capacity). **If you need ACID-compliant transactions, SOAP is the way to go**.
* In some cases, **designing SOAP services can actually be less complex** compared to REST. For web services that support complex operations, requiring content and context to be maintained, designing a SOAP service requires less coding in the application layer for transactions, security, trust, and other elements.
* **SOAP is highly extensible** through other protocols and technologies. In addition to WS-Security, SOAP supports WS-Addressing, WS-Coordination, WS-ReliableMessaging, and a host of other web services standards, a full list of which you can find on [W3C](https://www.w3.org/Submission/).

At the end of the day, the best protocol is the one that makes the most sense for the organization, the types of clients that you need to support, and what you need in terms of flexibility. Most new APIs are built using REST and JSON simply because it typically consumes less bandwidth and is easier to understand both for developers implementing initial APIs as well as other developers who may write other services against it. Because it’s more easily consumed by most of today’s web browsers, REST+JSON has become the defacto technology for the majority of public APIs. However, SOAP remains a valuable protocol in some circumstances. Plus, you don’t have to look far to find die-hard fans advocating for SOAP for certain use cases.

IInd Round

1. Current Project
2. How you rate yourself in diffident technologies. Like SQl, MVC, C#, Jquery
3. Parsing of JQuery selector(chained selector i.e. xpath). Its left to right or right to left
4. Want to show a chat box div in right corner, what should be the positing: fixed, bottom, and right=0px
5. different between absolute, relative, fixed

**Relative**: its mean “Relative to itself” eg position: relative; top: 10px will shift the div 10px down from its actual position

**Absolute**: This is a very powerful type of positioning that allows you to literally place any page element exactly where you want it. You use the positioning attributes top, left, bottom. and right to set the location. Remember that these values will be relative to the next parent element with relative (or absolute) positioning.

**Fixed**: This type of positioning is fairly rare but certainly has its uses. A fixed position element is positioned relative to the *viewport*, or the browser window itself. The viewport doesn't change when the window is scrolled, so a fixed positioned element will stay right where it is eg Chat box in corner of window

1. syntax to find id which starts with some specific pattern like div\_1, div\_2 div\_3: $(‘[id^=div\_]’), for ends with use $
2. JQuery No conflict
3. When I go to one page to other page, I want some delay, Like I want to show thankyou message. How this can be achieved
4. Window timeout vs. Set Interval

setTimeout(): It is a function that execute a JavaScript statement AFTER x interval.

setTimeout(function () {

something();

}, 1000); // Execute something() 1 second later.

setInterval(): It is a function that execute a JavaScript statement EVERY x interval.

setInterval(function () {

somethingElse();

}, 2000); // Execute somethingElse() every 2 seconds.

1. document ready and window on load. which one fire first:
2. SSRS
3. Serialization: When you create an object in a .Net framework application, you don't need to think about how the data is stored in memory. Because the .Net Framework takes care of that for you. However, if you want to store the contents of an object to a file, send an object to another process or transmit it across the network, you do have to think about how the object is represented because you will need to convert to a different format. This conversion is called SERIALIZATION.

Serialization is the process of converting an object into a stream of bytes to store the object or transmit it to memory, a database, or a file. Its main purpose is to save the state of an object in order to be able to recreate it when needed. The reverse process is called deserialization.

1. WebGarden and Web farm
2. If my hosted on web garden, which session I have to use and how I get it
3. Where session actually stores. Describe session properly
4. What happen If I restart the web server and my session are stored on sql server
5. Inproc and OutProc
6. Caching
7. When to put things in session and when to put things in cache
8. You have a project with 100 pages, what to do add exception handling for all pages
9. Request life cycle of MVC
10. Exception handling handles which type of exception, Compile time or run time
11. Out Parameter and Reference paramters
12. What indexing
13. when non cluster indexing work
14. TableScan, Index scan
15. I have a table in which salary of each employee for each month for many years. Show Anuall salary of each employee with year. Year EmplyeeName Salary
16. How much Year wise salary I get from the company
17. Right join in Entity Framework
18. I have a table Fist Name LastName Gender, show Mr or Misses + Fukll Name like Mr Adip Singhh
19. TempTable and table Variable

# Dependency, Generalization, Association, Aggregation, Composition in Object Oriented ProgrammingM

Mislenious

1. Finalize vs Dispose

dispose() and finalize() are the methods of C# which are invoked to free the unmanaged resources held by an object. The dispose() method is defined inside the interface IDisposable whereas, the method finalize() is defined inside the class object. The main difference between dispose() and finalize() is that the method **dispose**() has to be explicitly invoked by the user whereas, the method **finalize()** is invoked by the garbage collector, just before the object is destroyed.

The dispose() method releases the unmanaged resources that are held by an object of the class. The unmanaged resources are files, data connections, etc. The method dispose() is declared in the interface **IDisposeable** and it is implemented by the class by implementing the interface IDisposable. This method is not called automatically. The programmer has to implement it manually when you are creating a custom class that will be used by others

The finalize() method is defined in the **object** class. It is used for cleanup activities. This method is called by the garbage collector when the reference of an object is not used for a long time. Garbage collector frees the managed resources automatically but if you want to free the unmanaged resources like file handle, data connection, etc., the finalize method has to be implemented manually. The garbage collector invokes the method finalize() just before it destroys the object completely

1. Extention Methods

Extension methods enable you to “add” methods to existing types without creating a new derived type, recompiling the derived type or otherwise modifying the original type. It means we can extend methods from any type, even types we don't have access to the source code for. Some of the extension methods commonly used are Where, Select and Count.

So how do we create our own extension methods? Simple! We need a couple of things. First we need to create a **static** class.  
  
I will provide an example extending strings. Additionally, for simplifying the process to truncate a string, I will create a method.

1. **namespace** Example.Extensions
2. {
3. **public** **static** **class** StringExtender
4. {
5. **public** **static** **string** TruncateAt(**this** **string** source
6. ,   **int** maxLength)
7. {
8. **if** (source.Length <= maxLength)
9. {
10. **return** source;
11. }
13. **return** source.Substring(0, maxLength);
14. }
15. }
16. }

Asynchronous and awit in c#

When we are dealing with UI and on button click, we use a long running method like reading a large file or something else which will take a long time, in that case, the entire application must wait to complete the whole task.

In other words, if any process is blocked in a synchronous application, the entire application gets blocked and our application stops responding until the whole task completes.

Asynchronous programming is very helpful in this condition. By using Asynchronous programming, the Application can continue with the other work that does not depend on the completion of the whole task.

Suppose, we are using two methods as Method1 and Method2 respectively and both the methods are not dependent on each other and Method1 is taking a long time to complete its task. In Synchronous programming, it will execute the first Method1 and it will wait for completion of this method and then it will execute Method2. Thus, it will be a time intensive process even though both the methods are not depending on each other.

We can run all the methods parallelly by using the simple thread programming but it will block UI and wait to complete all the tasks. To come out of this problem, we have to write too many codes in traditional programming but if we will simply use the async and await keywords, then we will get the solutions in much less code

**FIS**

1. Authentication in Angular
2. How you authenticate the request after login
3. What you do with the token after authentication: Save in auth.service in token variable and access in the application when needed. We also have a getTocken function which hits the server to get new token. But this get token function is a Observable and it will may take time to get the token so we are passing token in supscribe menthod and returning the token. This will return old token for now but updates the token when observable get the response.
4. When you hit the api then how you get this request is authenticated . : Token
5. Where you save the token: answer in question 3
6. Local storage vs session storage in browser
7. If I closed my browser and opened after some time, will I get local and serssion storage? And what about the session , token
8. Directives
9. Pipes
10. Can I override or modify in build pipes.no
11. I have a component in component, how can they communicate: Input(), output(), shared services etc
12. Observables
13. Is it mandatory to subscribe the observable? Yes
14. If I don’t want observable the what can I use in place of it? promise
15. What are verbs( get put post patch delete) and to use which.
16. Which verb if have to use when deleting a user, but on back end I am update record to isDeleted =true i.e.(soft delete)
17. Validations in forms, if someone changed value using fiddler after submitino?
18. Private constructor in a class. Then how to create the object of that class(singleton class). We create a propery or method to expose its instance
19. In which order the constructor of base and drived class are called : First Base then Child
20. I have a class A and child class B and have Same method GetData() in base and child class, will it run or what happen. Ther will be a warning that you are hiding the definition of base class. We have to use new keyword to hiding the definition.
21. If I do A objA= new B(); and objA.GetData(). Which class function will call.
22. What will happen if we don’t have same function i.e. not overridden function
23. I have Interface IA and IB and both have same method GetData(). I implemented the both interface in same class.
24. Can Unique constrain have null value and how much and why
25. What will be the result of null== null in sql: false
26. What will the the result of null==null in c#. and if a=null and b=null then what will a==b
27. I have inserted a record in a table and want to get inserted primary key.” @@Identity
28. What is @@Identity: Global variables
29. Paging in sql. What if I don’t have primery key to fetch record.
30. RowOver
31. Pivot Functions
32. What are the point to be in mind when righting queries.: avoid or or where cluse, use cast instead of convert, avoid distince and orderby, avoid cursors, select only needed colums, use joins instead of sub queries, created table instead of select into, use vaarialbles instead of temporary tables, use proper indexes
33. Can I get table from sp: User defined types..