**Generics** allow you to delay the specification of the data type of programming elements in a class or a method, until it is actually used in the program.

Features of Generics

Generics is a technique that enriches your programs in the following ways:

* It helps you to maximize code reuse, type safety, and performance.
* static void Swap<T>(ref T lhs, ref T rhs)
* {
* T temp;
* temp = lhs;
* lhs = rhs;
* rhs = temp;
* }
* static void Main(string[] args)
* {
* int a, b;
* char c, d;
* a = 10;
* b = 20;
* c = 'I';
* d = 'V';
* //display values before swap:
* Console.WriteLine("Int values before calling swap:");
* Console.WriteLine("a = {0}, b = {1}", a, b);
* Console.WriteLine("Char values before calling swap:");
* Console.WriteLine("c = {0}, d = {1}", c, d);
* //call swap
* Swap<int>(ref a, ref b);
* Swap<char>(ref c, ref d);

# [Overloading generic methods](http://stackoverflow.com/questions/15525190/overloading-generic-methods)

|  |  |
| --- | --- |
| up vote7down voteaccepted | You could do :  public bool Save<T>(T entity) where T : class  { ... some storage logic ... }  public bool Save(SpecificClass entity)  { ... special logic ... } |

# [IEnumerable](http://stackoverflow.com/questions/3628425/ienumerable-vs-list-what-to-use-how-do-they-work)

|  |  |
| --- | --- |
| 366down voteaccepted | IEnumerable describes behavior, while List is an implementation of that behavior. When you use IEnumerable, you give the compiler a chance to defer work until later, possibly optimizing along the way. If you use ToList() you force the compiler to reify the results right away.  Whenever I'm "stacking" LINQ expressions, I use IEnumerable, because by only specifying the behavior I give LINQ a chance to defer evaluation and possibly optimize the program. Remember how LINQ doesn't generate the SQL to query the database until you enumerate it? Consider this: Enumerable Class Provides a set of **static** (**Shared** in Visual Basic) methods for querying objects that implement [IEnumerable<T>](https://msdn.microsoft.com/en-us/library/9eekhta0(v=vs.100).aspx).  Areas..   1. **protected void Application\_Start()** 2. **{** 3. ***//Register all application Areas*** 4. **AreaRegistration.RegisterAllAreas();** 6. **WebApiConfig.Register(GlobalConfiguration.Configuration);** 7. **FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);** 8. **RouteConfig.RegisterRoutes(RouteTable.Routes);** 9. **BundleConfig.RegisterBundles(BundleTable.Bundles);** 11. **}** 12. Its required for Modularization of project. 13. Suppose we don’t want to give any module to client we need to go to areas and delete the VIEWS in area also we need to delete the dll in bin folder 14. It should be defind in post build description.  Different types of routing in ASP.NET MVC  1. **Convention based routing** - to define this type of routing, we call MapRoute method and set its unique name, url pattern and specify some default values. 2. **Attribute based routing** - to define this type of routing, we specify the Route attribute in the action method of the controller.   routes.MapRoute(      name: "Default",      url: "{controller}/{action}/{id}",      defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }      ); Partial View in ASP.NET MVC 4 |

Partial view is like a regular view with a file extension .cshtml. We can use partial views in a situation where we need a header, footer reused for an MVC web application. We can say that it’s like a user control concept in ASP.NET.

We can call the partial view in a normal view like:

Hide   Copy Code

Html.RenderPartial("~/Views/Shared/\_Product.cshtml", product);

Or

Hide   Copy Code

@Html.Partial("~/Views/Shared/\_Product.cshtml", product);

Html.Partial returns a string,

 Html.RenderPartial calls Write internally, and returns void.

You can store the output of Html.Partial in a variable, or return it from a function. You cannot do this withHtml.RenderPartial because the result will be written to the Response stream during execution.

So@html.RenderPartial() has faster execution than @html.Partial() due to RenderPartial giving quick response to the output.

Returning a Partial view from the Controller's Action method:

1. **public** ActionResult PartialViewExample()
2. {
3. **return** PartialView();
4. }

**Render Partial View Using jQuery**  
Sometimes we need to load a partial view within a model popup at runtime, in this case we can render the partial view using JQuery element's load method.

1. <script type="text/jscript">
2. $('#partialView').load('/shared/PartialViewExample’);
3. </script>

**V**

# ActionResult Class

Represents the result of an action method.

public abstract class ActionResult

|  |  |  |
| --- | --- | --- |
| **Action Result** | **Helper Method** | **Description** |
| [ViewResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.viewresult(v=vs.118).aspx) | [View](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.view(v=vs.118).aspx#M:System.Web.Mvc.Controller.View) | Renders a view as a Web page. |
| [PartialViewResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.partialviewresult(v=vs.118).aspx) | [PartialView](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.partialview(v=vs.118).aspx#M:System.Web.Mvc.Controller.PartialView) | Renders a partial view, which defines a section of a view that can be rendered inside another view. |
| [RedirectResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.redirectresult(v=vs.118).aspx) | [Redirect](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.redirect(v=vs.118).aspx#M:System.Web.Mvc.Controller.Redirect(System.String)) | Redirects to another action method by using its URL. |
| [RedirectToRouteResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.redirecttorouteresult(v=vs.118).aspx) | [RedirectToAction](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.redirecttoaction(v=vs.118).aspx#M:System.Web.Mvc.Controller.RedirectToAction(System.String)) or [RedirectToRoute](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.redirecttoroute(v=vs.118).aspx" \l "M:System.Web.Mvc.Controller.RedirectToRoute(System.Object)) | Redirects to another action method. |
| [ContentResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.contentresult(v=vs.118).aspx) | [Content](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.content(v=vs.118).aspx#M:System.Web.Mvc.Controller.Content(System.String)) | Returns a user-defined content type. |
| [JsonResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.jsonresult(v=vs.118).aspx) | [Json](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.json(v=vs.118).aspx#M:System.Web.Mvc.Controller.Json(System.Object)) | Returns a serialized JSON object. |
| [JavaScriptResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.javascriptresult(v=vs.118).aspx) | [JavaScript](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.javascript(v=vs.118).aspx#M:System.Web.Mvc.Controller.JavaScript(System.String)) | Returns a script that can be executed on the client. |
| [HttpStatusCodeResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.httpstatuscoderesult(v=vs.118).aspx) | (None) | Returns a specific HTTP response code and description. |
| [HttpUnauthorizedResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.httpunauthorizedresult(v=vs.118).aspx) | (None) | Returns the result of an unauthorized HTTP request. |
| [HttpNotFoundResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.httpnotfoundresult(v=vs.118).aspx) | [HttpNotFound](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.httpnotfound(v=vs.118).aspx#M:System.Web.Mvc.Controller.HttpNotFound) | Indicates the requested resource was not found. |
| [FileResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.fileresult(v=vs.118).aspx) | [File](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.file(v=vs.118).aspx#M:System.Web.Mvc.Controller.File(System.Byte[],System.String)) | Returns binary output to write to the response. |
| [FileContentResult](https://msdn.microsoft.com/en-us/library/system.web.mvc.filecontentresult(v=vs.118).aspx) | [Controller.File(Byte[], String)](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.file(v=vs.118).aspx#M:System.Web.Mvc.Controller.File(System.Byte[],System.String)) or[Controller.File(Byte[], String, String)](https://msdn.microsoft.com/en-us/library/system.web.mvc.controller.file(v=vs.118).aspx#M:System.Web.Mvc.Controller.File(System.Byte[],System.String,System.String)) | Sends the contents of a binary file to the response |

public class JsonResult : ActionResult

## return View()

# RedirectToAction()

1. Return View doesn't make a new requests, it just renders the view without changing URLs in the browser's address bar.
2. Return RedirectToAction makes a new requests and URL in the browser's address bar is updated with the generated URL by MVC.
3. Return Redirect also makes a new requests and URL in the browser's address bar is updated, but you have to specify the full URL to redirect
4. Between RedirectToAction and Redirect, best practice is to use RedirectToAction for anything dealing with your application actions/controllers. If you use Redirect and provide the URL, you'll need to modify those URLs manually when you change the route table.
5. RedirectToRoute redirects to a specific route defined in the Route table.

BUNDLING

public static void RegisterBundles(BundleCollection bundles)

{ bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(

"~/Scripts/modernizr-\*"));

bundles.Add(new ScriptBundle("~/bundles/jquery").Include(

"~/Scripts/jquery-{version}.js"));

bundles.Add(new ScriptBundle("~/bundles/jqueryDataTables").Include(

"~/Scripts/jquery.dataTables.min.js",

"~/Scripts/dataTables.responsive.min.js"

));

bundles.Add(new StyleBundle("~/Content/css").Include(

"~/Content/bootstrap.css",

"~/Content/site.css"));

}

@Styles.Render("~/Content/css")

@Scripts.Render("~/bundles/modernizr")

## [WCF or ASP.NET Web APIs? My two cents on the subject](http://blogs.microsoft.co.il/idof/2012/03/05/wcf-or-aspnet-web-apis-my-two-cents-on-the-subject/)

<http://blogs.microsoft.co.il/idof/2012/03/05/wcf-or-aspnet-web-apis-my-two-cents-on-the-subject/>

What is the purpose of the WebAPIs?

the main goal was to support SOAP + WS-\* over a wide variety of transports.

There is also a need to also support non-SOAP services, especially over HTTP, where you can harness the power of the HTTP protocol to create HTTP services: services that are activated by simple GET requests, or by passing plain XML over POST, and respond with non-SOAP content such as plain XML, a JSON string, or any other content that can be used by the consumer.

Support for non-SOAP services was very much needed in WCF back then, mostly because some clients, such as web browsers, were not that suitable to handle SOAP messages (plenty of XML parsing and DOM manipulation).

So in WCF 3.5 we got the [WebHttpBinding](http://msdn.microsoft.com/en-us/library/system.servicemodel.webhttpbinding.aspx) – a new binding that helped us create this kind of non-SOAP service over HTTP, better known as a RESTful service.

That was the main goal of the Web APIs, known back then as the WCF Web APIs: to stop looking at HTTP through the eyes of WCF – as just a transport protocol to pass requests. Rather, it allows us to look at it as the real application-level protocol it is – a rich, interoperable, resource-oriented protocol. The purpose of the Web APIs was to properly use URIs, HTTP headers, and body to create HTTP services for the web, and for everyone else that wished to embrace HTTP as its protocol and lifelong friend.

Why do we need ~~REST~~ HTTP services? What’s wrong with SOAP-over-HTTP?

The world of SOAP and the world of HTTP services are very different. SOAP allows us to place all the knowledge required by our service in the message itself, disregarding its transport protocol, whether it is TCP, HTTP, UDP, PGM, Named Pipes… But unlike TCP, UDP and the other [level 4-5](http://en.wikipedia.org/wiki/OSI_model) protocols, HTTP is an application-level protocol, and as such it offers a wide variety of features:

* It supports verbs that define the action – query information using GET, place new information and update existing using POST or PUT, remove information using DELETE etc.
* It contains message headers that are very meaningful and descriptive – headers that suggest the content type of the message’s body, headers that explain how to cache information, how to secure it etc.

Is there still use for WCF? when should I choose Web APIs over WCF?

Recall my points from before – HTTP is a lot more than a transport protocol; use SOAP across the board and consider HTTP as no more than another way to pass messages.

* If your intention is to create services that support special scenarios – one way messaging, message queues, duplex communication etc, then you’re better of picking WCF
* If you want to create services 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 transports are unavailable, then you’re better off with WCF and using both SOAP-based bindings and the WebHttp binding.
* If you want to create resource-oriented services over HTTP that can use the full features of HTTP – define cache control for browsers, versioning and concurrency using ETags, pass various content types such as images, documents, HTML pages etc., use URI templates to include Task URIs in your responses, then the new Web APIs are the best choice for you.
* If you want to create a multi-target service that can be used as both resource-oriented service over HTTP and as RPC-style SOAP service over TCP – talk to me first, so I’ll give you some pointers.

# DataAnnotations Attributes

The [System.ComponentModel.DataAnnotations](https://msdn.microsoft.com/en-us/library/system.componentmodel.dataannotations(v=vs.100).aspx) attributes can be used with the Entity Data Model (EDM), LINQ to SQL, and other data models. You can also create custom validation attributes. For more information, see [How to: Customize Data Field Validation in the Data Model Using Custom Attributes](https://msdn.microsoft.com/en-us/library/cc668224(v=vs.100).aspx).

1. Add a class to your project to contain the partial class definitions. For more information, see [How to: Customize Data Field Validation in the Data Model](https://msdn.microsoft.com/en-us/library/cc488527(v=vs.100).aspx).
2. Add a namespace declaration to the partial class that matches the namespace of the data model that you are using.
3. using System.ComponentModel.DataAnnotations;
4. namespace MvcDA {
5. [MetadataType(typeof(ProductMD))]
6. public partial class Product {
7. public class ProductMD {
8. [StringLength(50),Required]
9. public object Name { get; set; }
10. [StringLength(15)]
11. public object Color { get; set; }
12. [Range(0, 9999)]
13. public object Weight { get; set; }
14. // public object NoSuchProperty { get; set; }
15. }
16. }
17. }

# [How to cache data in a MVC application](http://stackoverflow.com/questions/343899/how-to-cache-data-in-a-mvc-application)

|  |  |
| --- | --- |
| 47down voteaccepted | Reference the System.Web dll in your model and use System.Web.Caching.Cache  public string[] GetNames()  {  string[] names = Cache["names"] as string[];  if(names == null) //not in cache  {  names = DB.GetNames();  Cache["names"] = names;  }  return names;  }  A bit simplifie |