# What is Web API and why to use it ?

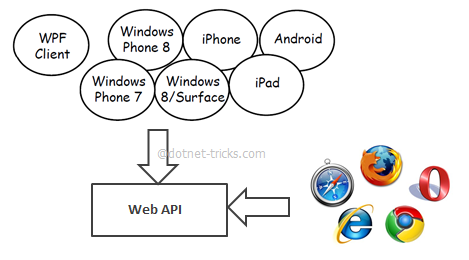
Asp.Net Web API is a framework for building HTTP services that can be consume by a broad range of clients including browsers, mobiles, iphone and tablets. It is very similar to ASP.NET MVC since it contains the MVC features such as routing, controllers, action results, filter, model binders, IOC container or dependency injection. But it is not a part of the MVC Framework. It is a part of the core ASP.NET platform and can be used with MVC and other types of Web applications like Asp.Net WebForms. It can also be used as an stand-alone Web services application.

## Why Asp.Net Web API (Web API) ?

Today, a web-based application is not enough to reach it's customers. People are very smart, they are using iphone, mobile, tablets etc. devices in its daily life. These devices also have a lot of apps for making the life easy. Actually, we are moving from the web towards apps world.

So, if you like to expose your service data to the browsers and as well as all these modern devices apps in fast and simple way, you should have an API which is compatible with browsers and all these devices.

For example twitter,facebook and Google API for the web application and phone apps.



Web API is the great framework for exposing your data and service to different-different devices. Moreover 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 features of HTTP (like URIs, request/response headers, caching, versioning, various content formats) and you don't need to define any extra config settings for different devices unlike WCF Rest service.

## Web API Features

1. It supports convention-based CRUD Actions since it works with HTTP verbs GET,POST,PUT and DELETE.
2. Responses have an Accept header and HTTP status code.
3. Responses are formatted by Web API’s MediaTypeFormatter into JSON, XML or whatever format you want to add as a MediaTypeFormatter.
4. It may accepts and generates the content which may not be object oriented like images, PDF files etc.
5. It has automatic support for OData. Hence by placing the new [Queryable] attribute on a controller method that returns IQueryable, clients can use the method for OData query composition.
6. It can be hosted with in the applicaion or on IIS.
7. It also supports the MVC features such as routing, controllers, action results, filter, model binders, IOC container or dependency injection that makes it more simple and robust.

## Why to choose Web API ?

1. If we need a Web Service and don’t need SOAP, then ASP.Net Web API is best choice.
2. It is Used to build simple, non-SOAP-based HTTP Services on top of existing WCF message pipeline.
3. It doesn't have tedious and extensive configuration like WCF REST service.
4. Simple service creation with Web API. With WCF REST Services, service creation is difficult.
5. It is only based on HTTP and easy to define, expose and consume in a REST-ful way.
6. It is light weight architecture and good for devices which have limited bandwidth like smart phones.
7. It is open source.

# Difference between WCF and Web API and WCF REST and Web Service

The .Net framework has a number of technologies that allow you to create HTTP services such as Web Service, WCF and now Web API. There are a lot of articles over the internet which may describe to whom you should use. Now a days, you have a lot of choices to build HTTP services on .NET framework. In this article, I would like to share my opinion with you over Web Service, WCF and now Web API. For more information about Web API refers [What is Web API and why to use it ?](http://www.dotnet-tricks.com/Tutorial/webapi/VG9K040413-What-is-Web-API-and-why-to-use-it-?.html).

## Web Service

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

## WCF

1. It is also based on SOAP and return data in XML form.
2. It is the evolution of the web service(ASMX) and support various protocols like TCP, HTTP, HTTPS, Named Pipes, MSMQ.
3. The main issue with WCF is, its tedious and extensive configuration.
4. It is not open source but can be consumed by any client that understands xml.
5. It can be hosted with in the applicaion or on IIS or using window service.

## WCF Rest

1. To use WCF as WCF [Rest service](http://kellabyte.com/2011/09/04/clarifying-rest/) you have to enable webHttpBindings.
2. It support HTTP GET and POST verbs by [WebGet] and [WebInvoke] attributes respectively.
3. To enable other HTTP verbs you have to do some configuration in IIS to accept request of that particular verb on .svc files
4. Passing data through parameters using a WebGet needs configuration. The UriTemplate must be specified
5. It support XML, JSON and ATOM data format.

## Web API

1. This is the new framework for building HTTP services with easy and simple way.
2. Web API is open source an ideal platform for building REST-ful services over the .NET Framework.
3. Unlike WCF Rest service, it use the full featues of HTTP (like URIs, request/response headers, caching, versioning, various content formats)
4. 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.
5. It can be hosted with in the application or on IIS.
6. It is light weight architecture and good for devices which have limited bandwidth like smart phones.
7. 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

1. Choose WCF when you want to create a service that should support special scenarios such as one way messaging, message queues, duplex communication etc.
2. 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.
3. 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).
4. Choose Web API when you want to expose your service to a broad range of clients including browsers, mobiles, iphone and tablets.

# Asp.Net Web API Vs Asp.Net MVC

1. Asp.Net MVC is used to create web applications that returns both views and data but Asp.Net Web API is used to create full blown HTTP services with easy and simple way that returns only data not view.
2. Web API helps to build REST-ful services over the .NET Framework and it also support content-negotiation(it's about deciding the best response format data that could be acceptable by the client. it could be JSON,XML,ATOM or other formatted data), self hosting which are not in MVC.
3. Web API also takes care of returning data in particular format like JSON,XML or any other based upon the Accept header in the request and you don't worry about that. MVC only return data in JSON format using JsonResult.
4. In Web API the request are mapped to the actions based on HTTP verbs but in MVC it is mapped to actions name.
5. Asp.Net Web API is new framework and part of the core ASP.NET framework. The model binding, filters, routing and others MVC features exist in Web API are different from MVC and exists in the new System.Web.Http assembly. In MVC, these featues exist with in System.Web.Mvc. Hence Web API can also be used with Asp.Net and as a stand alone service layer.
6. You can mix Web API and MVC controller in a single project to handle advanced AJAX requests which may return data in JSON, XML or any others format and building a full blown HTTP service. Typically, this will be called Web API self hosting.
7. When you have mixed MVC and Web API controller and you want to implement the authorization then you have to create two filters one for MVC and another for Web API since boths are different.
8. Moreover, Web API is light weight architecture and except the web application it can also be used with smart phone apps.