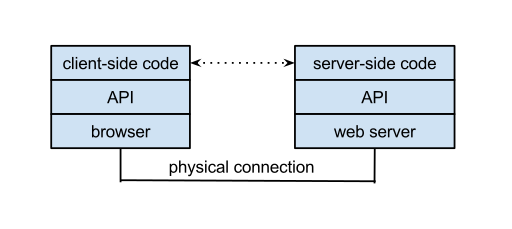
A web API is an application programming interface (API) for either a web server or a web browser. It is a web development concept, **usually limited to a web application's client-side** (including any web frameworks being used), and thus usually does not include web server or browser implementation details such as SAPIs or web browser engine APIs unless publicly accessible by a remote web application.



Something may Confuse：**Tweedy is a Python library for accessing the Twitter API**.

Web API的主要功能

1. 支持基于Http verb (GET, POST, PUT, DELETE)的CRUD (create, retrieve, update, delete)操作

    通过不同的http动作表达不同的含义，这样就不需要暴露多个API来支持这些基本操作。

2. 请求的回复通过Http Status Code表达不同含义，并且客户端可以通过[Accept header](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html)来与服务器协商格式，例如你希望服务器返回JSON格式还是XML格式。

3. 请求的回复格式支持 JSON，XML，并且可以扩展添加其他格式。

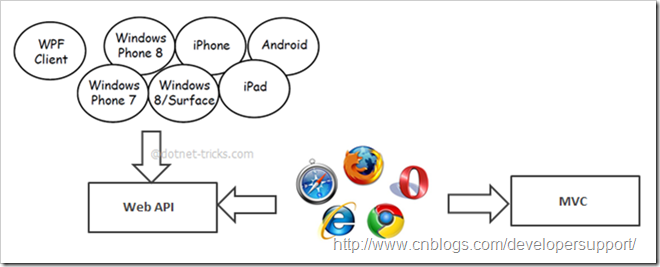
4. 原生支持[OData](http://www.odata.org/)。

5. 支持Self-host或者IIS host。

6. 支持大多数MVC功能，例如Routing/Controller/Action Result/Filter/Model Builder/IOC Container/Dependency Injection。

# Web API vs MVC

你可能会觉得Web API 与MVC很类似，他们有哪些不同之处呢？先上图，这就是他们最大的不同之处。

[](http://images.cnitblog.com/blog/502305/201309/06170419-f5c577398f2e43ffb9a380bf42fab2ab.png)

详细点说他们的区别，

* MVC主要用来构建网站，既关心数据也关心页面展示，而Web API只关注数据
* Web API支持格式协商，客户端可以通过Accept header通知服务器期望的格式
* Web API支持Self Host，MVC目前不支持
* Web API通过不同的http verb表达不同的动作(CRUD)，MVC则通过Action名字表达动作
* Web API内建于ASP.NET System.Web.Http命名空间下，MVC位于System.Web.Mvc命名空间下，因此model binding/filter/routing等功能有所不同
* 最后，Web API非常适合构建移动客户端服务

**Server-side**

A [server-side](https://en.wikipedia.org/wiki/Server-side) web API is a programmatic [interface](https://en.wikipedia.org/wiki/Interface_%28computing%29) consisting of one or more publicly exposed endpoints to a defined [request-response](https://en.wikipedia.org/wiki/Request-response) message system, typically expressed in [JSON](https://en.wikipedia.org/wiki/JSON) or [XML](https://en.wikipedia.org/wiki/XML), which is exposed via the web—most commonly by means of an [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)-based web server. [Mashups](https://en.wikipedia.org/wiki/Mashup_%28web_application_hybrid%29) are [web applications](https://en.wikipedia.org/wiki/Web_application) which combine the use of multiple server-side web APIs. [Webhooks](https://en.wikipedia.org/wiki/Webhook) are server-side web APIs that take as input a [Uniform Resource Identifier](https://en.wikipedia.org/wiki/Uniform_Resource_Identifier) (URI) that is designed to be used like a remote [named pipe](https://en.wikipedia.org/wiki/Named_pipe) or a type of [callback](https://en.wikipedia.org/wiki/Callback_%28computer_programming%29) such that the server acts as a client to dereference the provided URI and trigger an event on another server which handles this event thus providing a type of [peer-to-peer](https://en.wikipedia.org/wiki/Peer-to-peer) [IPC](https://en.wikipedia.org/wiki/Inter-process_communication).

**Endpoints**

Endpoints are important aspects of interacting with server-side web APIs, as they specify where resources lie that can be accessed by third party software. Usually the access is via a URI to which HTTP requests are posed, and from which the response is thus expected.

Endpoints need to be static, otherwise the correct functioning of software that interacts with it cannot be guaranteed. If the location of a resource changes (and with it the endpoint) then previously written software will break, as the required resource can no longer be found at the same place. As API providers still want to update their web APIs, many have introduced a versioning system in the URI that points to an endpoint, for example the Clarifai API: The endpoint for the tagging functionality within the web API has the following URI: "https://api.clarifai.com/v1/tag/". The "/v1/" part of the URI specifies access to the first version of the web API. If clarifai decides to update to version two, they can do this while still maintaining support for third party software that uses the first version.

### Documentation

Server-side web APIs are an interface for the outside world to interact with the business logic. For many companies this internal business logic and the intellectual property associated with it are what distinguishes them from other companies, and potentially what gives them a competitive edge. They do not want this information to be exposed. However, in order to provide a web API of high quality, there **needs** to be a sufficient level of documentation. One API provider that not only provides documentation, but also links to it in its error messages is Twilio.

However, there are now directories of popular documented server-side web APIs.