

Implementando Web APIs en ASP.NET MVC

Manual de estudiante



Miguel Muñoz Serafín



Implementando Web APIs en ASP.NET MVC

Manual de estudiante

Diciembre de 2017

Soporte técnico:

soporte@mail.ticapacitacion.com

<http://ticapacitacion.com/promociones/webapi.html>



Contenido

Acerca del módulo.....	6
Audiencia.....	6
Objetivos	6
Requerimientos	6
Lección 1 Desarrollando una API Web con ASP.NET Web API	7
¿Qué es Web API?	8
Creando una API Web con el Framework ASP.NET Web API	10
Valores devueltos por los métodos de Acción	17
Formato de los datos devueltos.....	21
Formateadores de contenido devuelto (Media Formatters)	22
Soporte a BSON en ASP.NET Web API 2.1.....	28
Servicios RESTful.....	30
Lección 2 Enrutamiento en ASP.NET Web API	33
Tablas de Enrutamiento	34
Variaciones de enrutamiento.....	37
Enrutamiento por el nombre de Acción.....	39
Atributo NonAction	40
Enrutamiento por atributo en ASP.NET Web API 2.....	41
Prefijos de Rutas.....	44
Restricciones en las Rutas	46
Restricciones personalizadas de rutas	48
Parámetros opcionales del URI y valores predeterminados	50
Nombres de Ruta	51
Orden de las Rutas	53
Lección 3 Invocando servicios ASP.NET Web API con Microsoft ASP.NET Web API Client.....	55
Configurando el servicio Web API para pruebas locales.....	56
Microsoft ASP.NET Web API Client	58
Obteniendo un resultado (HTTP GET)	61
Obteniendo un resultado desde aplicaciones ASP.NET MVC	63



Obteniendo un resultado desde aplicaciones Windows Presentation Foundation.....	64
Obteniendo un resultado desde aplicaciones de la Plataforma Universal de Windows	65
Obteniendo un resultado desde aplicaciones Xamarin	66
Utilizando formateadores de contenido en el método ReadAsync	67
Utilizando formateadores de contenido en aplicaciones ASP.NET MVC	68
Utilizando formateadores de contenido en aplicaciones Windows Presentation Foundation	69
Utilizando formateadores de contenido en aplicaciones de la Plataforma Universal de Windows. 70	
Utilizando formateadores de contenido en aplicaciones Xamarin	71
Creando un Recurso (HTTP POST)	72
Creando un Recurso desde aplicaciones ASP.NET MVC	74
Creando un Recurso desde aplicaciones Windows Presentation Foundation.....	75
Creando un Recurso desde aplicaciones de la Plataforma Universal de Windows	76
Creando un Recurso desde aplicaciones Xamarin	77
Actualizando un Recurso (HTTP PUT).....	78
Actualizando un Recurso desde aplicaciones ASP.NET MVC	79
Actualizando un Recurso desde aplicaciones Windows Presentation Foundation	80
Actualizando un Recurso desde aplicaciones de la Plataforma Universal de Windows	81
Actualizando un Recurso desde aplicaciones Xamarin	82
Eliminando un Recurso (HTTP DELETE)	83
Eliminando un Recurso desde aplicaciones ASP.NET MVC	84
Eliminando un Recurso desde aplicaciones Windows Presentation Foundation	85
Eliminando un Recurso desde aplicaciones de la Plataforma Universal de Windows.....	86
Eliminando un Recurso desde aplicaciones Xamarin	87
Enviando peticiones HTTP personalizadas	88
Enviando peticiones HTTP personalizadas desde aplicaciones ASP.NET MVC	89
Enviando peticiones HTTP personalizadas desde aplicaciones Windows Presentation Foundation	90
Enviando peticiones HTTP personalizadas desde aplicaciones de la Plataforma Universal de Windows.....	91
Enviando peticiones HTTP personalizadas desde aplicaciones Xamarin	92
Lección 4 Invocando servicios ASP.NET Web API con funciones JavaScript y jQuery	93
Obteniendo un resultado (HTTP GET)	94



Realizando peticiones desde otros dominios.....	96
Habilitando CORS en ASP.NET Web API 2	97
Creando un Recurso (HTTP POST)	101
Actualizando un Recurso (HTTP PUT).....	102
Eliminando un Recurso (HTTP DELETE)	103
Enviando peticiones HTTP personalizadas	104



Acerca del módulo

La mayoría de las aplicaciones Web, aplicaciones de escritorio o aplicaciones móviles, requieren de una integración con sistemas externos. Una forma recomendada para implementar esa integración es a través de APIs Web. Debido a esto, es importante conocer la forma de crear y utilizar **APIs Web (Web APIs)** para poder implementar adecuadamente la interacción de las aplicaciones con sistemas externos.

Podemos utilizar APIs Web para implementar servicios de **Transferencia de Estado Representacional (Representational State Transfer o REST)** en una aplicación. Los servicios REST ayudan a reducir la sobrecarga de la aplicación y limitan los datos que son transmitidos entre los sistemas cliente y servidor. Para poder implementar de forma efectiva APIs Web estilo REST en una aplicación, es necesario conocer la forma de invocar a los Servicios de APIs Web desde aplicaciones ejecutándose en distintas plataformas, tales como, aplicaciones .NET, jQuery, Android, iOS, Windows Phone o Windows 10.

Audiencia

Este módulo está dirigido a desarrolladores con experiencia en el desarrollo de aplicaciones web con ASP.NET MVC y que desean implementar APIs web para ser consumidas por clientes Xamarin, .NET, UWP o JavaScript y jQuery.

Objetivos

Al finalizar este módulo, los participantes contarán con las habilidades y conocimientos para:

- Desarrollar un servicio ASP.NET Web API.
- Describir las distintas opciones de enrutamiento en ASP.NET Web API.
- Invocar un servicio Web API desde aplicaciones .NET.
- Invocar un servicio Web API desde aplicaciones Android, iOS y Windows Phone con Xamarin.
- Invocar un servicio Web API desde aplicaciones Universal Windows Platform (Windows 10)
- Invocar un servicio Web API desde aplicaciones JavaScript y jQuery.

Requerimientos

Para la realización de los ejercicios de este módulo, es necesario contar con lo siguiente:

- Un equipo de desarrollo con Visual Studio 2015 o posteriores. Puede utilizarse la versión gratuita **Visual Studio Community** que puede descargarse desde el siguiente enlace:

<https://www.visualstudio.com/community/>



Lección 1

Desarrollando una API Web con ASP.NET Web API

El Framework Web API de ASP.NET facilita la creación de APIs para aplicaciones móviles, aplicaciones de escritorio, servicios Web, aplicaciones Web y otras aplicaciones. Al crear una API Web permitimos que la información de nuestra aplicación Web esté disponible para que otros desarrolladores puedan utilizarla en sus sistemas.

Cada aplicación Web tiene una metodología funcional diferente, esta diferencia puede causar problemas de interoperabilidad entre las aplicaciones.

Los servicios REST tienen un diseño ligero y el Framework Web API ayuda a implementar servicios REST para solucionar los problemas de interoperabilidad. Para implementar servicios REST es necesario conocer cómo utilizar los diferentes métodos de enrutamiento que ASP.NET proporciona.

Objetivos de la lección

Al finalizar esta lección, los participantes contarán con las habilidades y conocimientos para:

- Describir qué es Web API.
- Crear una API Web para una aplicación Web MVC.
- Describir los servicios REST.
- Describir los distintos valores que puede devolver un servicio Web API.
- Crear un formateador de contenido personalizado.
- Describir los formatos de devolución de datos.
- Explorar una API Web utilizando Internet Explorer como cliente.



¿Qué es Web API?

API es un acrónimo de **Application Programming Interface** (Interfaz de Programación de Aplicaciones). Tradicionalmente, de forma local, una API se expone a través de archivos DLL. En la Web, una API se expone a través de Servicios Web que permiten que las aplicaciones cliente obtengan y realicen operaciones con los datos que el servicio expone. Existen distintos tipos de Servicios Web que se caracterizan principalmente por la forma en que realizan la comunicación con el cliente y el formato en que intercambian información. Un ejemplo de estos servicios son los Servicios Web que utilizan **SOAP** (Simple Object Access Protocol) como protocolo de comunicación e intercambio de datos en formato XML. En la actualidad, una opción para exponer APIs en la Web es mediante Servicios **REST**. **REST** es un acrónimo de **Representational State Transfer** (Transferencia de Estado Representacional) y es un estilo de arquitectura de software para crear APIs que utilicen HTTP como su método de comunicación subyacente. El término **REST** se originó en el año 2000, en una tesis doctoral sobre la Web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP.

ASP.NET Web API es un Framework que forma parte de ASP.NET MVC y que permite construir APIs habilitadas para **REST**. Las APIs habilitadas para REST ayudan a que sistemas externos utilicen la lógica de negocios implementada en una aplicación, incrementando la reutilización de dicha lógica.

Web API facilita la comunicación de dos vías entre el cliente y el servidor a través de tareas tales como:

- Indicar a una aplicación realizar una tarea específica.
- Leer datos.
- Actualizar datos.

Web API permite que los desarrolladores obtengan información de negocio mediante el uso de **REST** sin crear peticiones XML complicadas tal como en los servicios SOAP. Web API recibe las peticiones directamente en los URLs eliminando de este modo la necesidad de peticiones complicadas. Por ejemplo, el siguiente URL obtiene información de una entidad **Product** con el identificador **1**.

<http://northwind.com/api/products/1>

Web API utiliza el URL de las peticiones y obtiene resultados en formato **JSON**. **JSON** es un acrónimo de **JavaScript Object Notation** y es un formato ligero para intercambio de datos. Para las personas resulta de fácil lectura y escritura mientras que para las máquinas les resulta fácil analizarlo y generarlo. **JSON** está basado en un subconjunto del lenguaje de programación JavaScript. Es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son familiares para los programadores de lenguajes de la familia C, incluyendo C++, C#, Java, JavaScript, Perl, Python y muchos otros. Estas propiedades hacen que **JSON** sea ideal para el intercambio de datos.



Para mayor información acerca de **JSON** se recomienda visitar el siguiente enlace:

Introducing JSON

<http://json.org>

El siguiente código muestra una respuesta de una petición Web API en formato JSON.

```
[{"Id":1,"Name":"Chai","Category":"Beverages","Price":18.0000},{ "Id":2,"Name":"Chang","Category":"Beverages","Price":19.0000},{ "Id":3,"Name":"Aniseed Syrup","Category":"Condiments","Price":10.0000}]
```

REST y Web API permiten que todos los diferentes tipos de aplicaciones, incluyendo aplicaciones para dispositivos móviles, interactúen con servicios. En particular, REST y Web API proporcionan los siguientes beneficios para aplicaciones móviles:

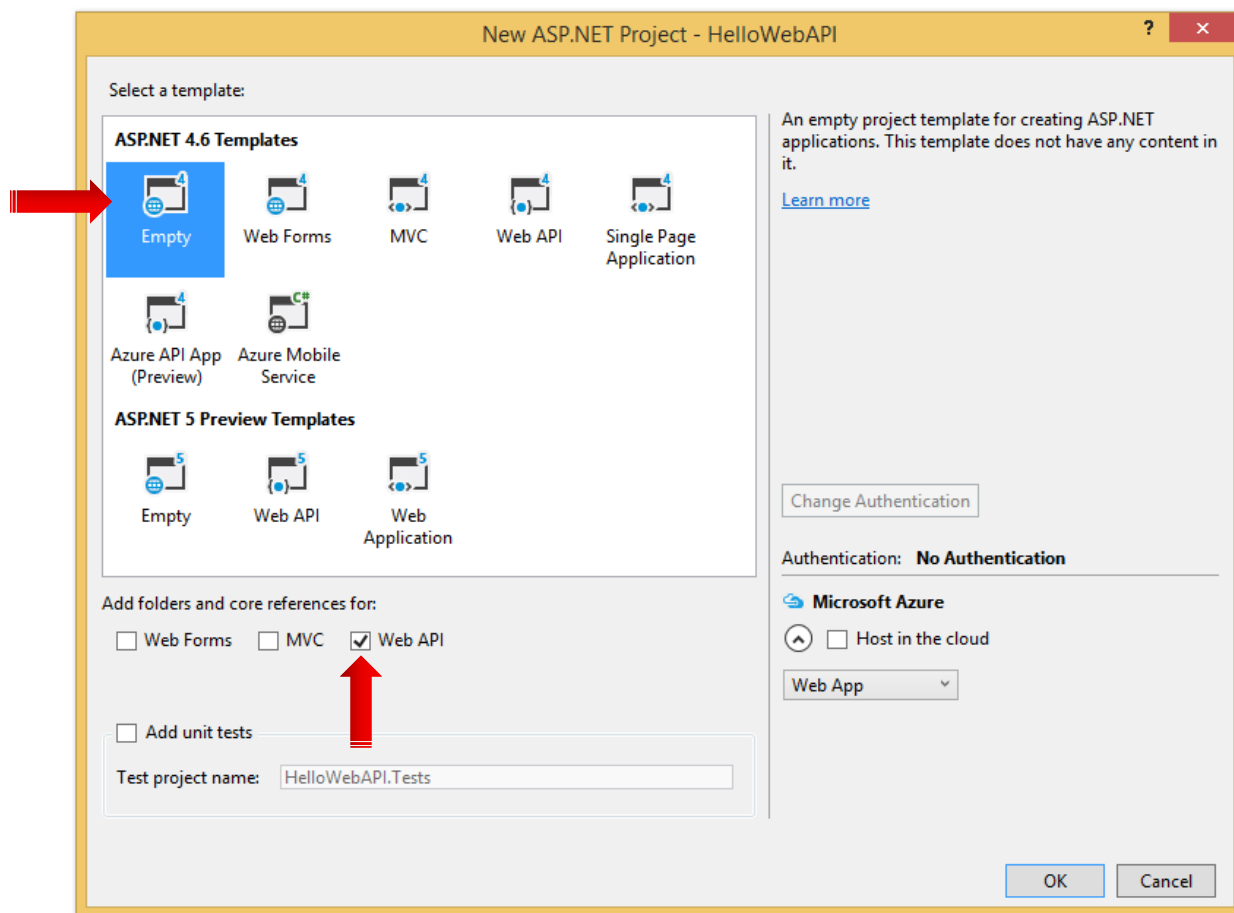
- Reducen el poder de procesamiento necesario para crear mensajes de petición complejos para obtener datos.
- Mejoran el rendimiento de la aplicación reduciendo la cantidad de intercambio de datos entre el cliente y el servidor.

Creando una API Web con el Framework ASP.NET Web API

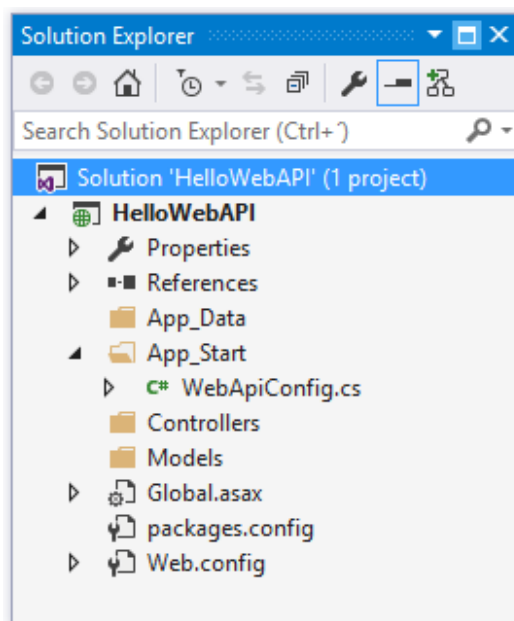
ASP.NET proporciona una plantilla para proyectos **ASP.NET Web API** que facilitan la creación de APIs Web. Para implementar **ASP.NET Web API** utilizando Visual Studio, podemos seguir los siguientes pasos:

1. Abrir **Visual Studio** y seleccionar la opción **File/New Project** de la barra de menús.
2. En la ventana **New Project**:
 - a. Seleccionar la plantilla **ASP.NET Web Application**.
 - b. Proporcionar el nombre del proyecto.
 - c. Dar clic en **OK** para continuar.
3. En la ventana **New ASP.NET Project**, seleccionar alguna de las siguientes plantillas:
 - a. **Empty**. Esta es una plantilla sin contenido. Para implementar ASP.NET Web API con esta plantilla es recomendable seleccionar la casilla **Web API** de la sección **Add folders and core references for** para generar la estructura de directorios necesaria, así como para agregar las referencias de los Assemblies requeridos.
 - b. **Web API**. Esta plantilla crea la estructura de directorios y agrega las referencias de los Assemblies necesarios además de código implementando funcionalidad básica.
 - c. **Single Page Application**. Esta es una plantilla utilizada para crear aplicaciones Web HTML5 con clientes JavaScript que consumen ASP.NET Web API.
 - d. **Azure API App**. Esta plantilla proporciona una plataforma para hospedar APIs REST en **Microsoft Azure**.
 - e. **Azure Mobile Service**. Esta plantilla permite implementar ASP.NET Web API como soporte (backend) a aplicaciones móviles que utilizan **Microsoft Azure Mobile Services**.

La siguiente imagen muestra la ventana **New ASP.NET Project** donde se ha seleccionado la plantilla **Empty** para implementar **ASP.NET Web API**. Debemos notar que se ha seleccionado la casilla **Web API** de la sección **Add folders and core references for** para generar la estructura de directorios necesaria, así como para agregar las referencias de los Assemblies requeridos.

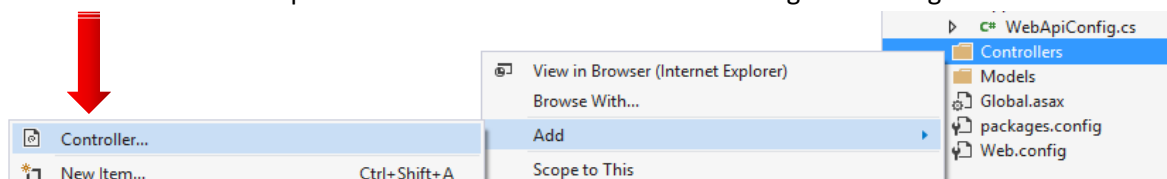


Después de que el proyecto haya sido creado, se mostrará una estructura de directorios similar a la siguiente.



Después de crear el proyecto, será necesario agregar una clase API Controller que derive de **ApiController**.

Para agregar una nueva clase API Controller podemos seleccionar la opción **Add/Controller...** del menú contextual de la carpeta **Controllers** como se muestra en la siguiente imagen.



En la ventana **Add Scaffold** podemos seleccionar alguna de las opciones **Web API 2**.

Las siguientes imágenes muestran las ventanas **Add Scaffold** y **Add Controller** para crear la clase API Controller **Products** vacía a partir de la plantilla **Web API 2 Controller - Empty**.