# ASP.NET Web API

The ASP.NET Web API is an extensible framework for building HTTP based services that can be accessed in different applications on different platforms such as web, windows, mobiles etc.

It works more or less the same way as ASP.NET MVC web application except that it sends data as a response instead of html view. It is like a webservice or WCF service but the exception is that it only supports HTTP protocol.

ASP.NET Web API framework includes new HttpClient to communicate with Web API Server. HttpClient can be used in ASP.MVC server side, Windows From application, Console application or other apps.

# WebApiConfig.cs

The WebApiConfig.cs is configuration file for Web API. You can configure routes and other things for web API, same like RouteConfig.cs is used to configure MVC routes. It also creates Web API controller ValuesController.cs by default.

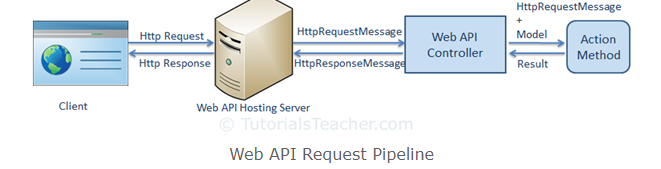
# Web API Controller

Web API Controller is similar to ASP.NET MVC Controller. It handles incoming HTTP requests and send response back the caller.

Web API Controller must be derived from System.Web.Http.ApiController class. All the public methods of the controller are called action methods.

# Configure Web API

Web API supports code based configuration. It can be configured in web.config file.



# Web API Routing

Web API routing is similar to ASP.NET MVC Routing. It routes an incoming HTTP request to particular action method on a web API controller.

Web API supports two types of routing:

1. Convention-based Routing in WebApiConfig.cs config.Routes.MapHttpRoute.
2. Attribute Routing (supported in Web API 2): use [Route()] attribute on any controller or action method

# Parameter Binding

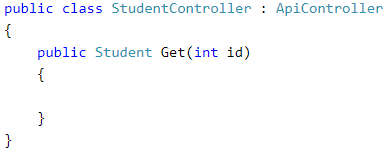
Web API binds action method parameters either with URL’s query string or with request body depending on the parameter type.

By default, Web API gets the value of a primitive parameter from the query string and complex type parameter from the request body.

.NET primitive types are int, bool, string, datetime, decimal or any type that can be converted from string type.

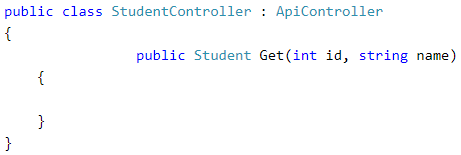
1. **Get action method with primitive parameter.**

As the below picture, the Get action method include id parameter of int type. So, Web API will try to extract the value of id from the query string of requested URL, convert it into int and assign it to id parameter of Get action method. For example, if an HTTP request is <http://localhost/api/student?id=1>, then value of id parameter will be 1.



Note: Query string parameter name and action method parameter name must be the same (case-insensitive).

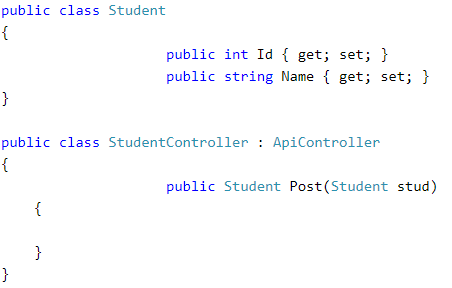
1. **Get action method with Multiple Primitive parameters**



<http://localhost/api/student?id=1&name=steve>

Note: Query string parameter names must match with the name of an action method parameter. However, they can be in different order.

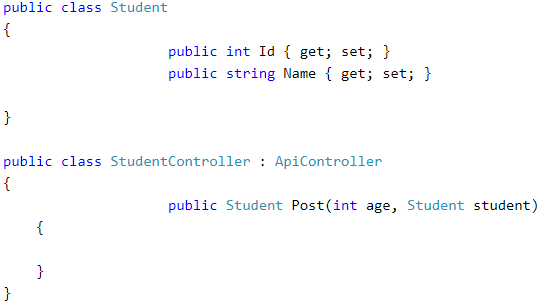
1. **Post action method with complex type parameters**



<http://localhost/api/student>

Web API will try to get the values of stud parameter from HTTP request body. Web API will extract the JSON object from the Request body and convert it into Student object automatically.

1. **Post action method with primitive and complex type parameters**



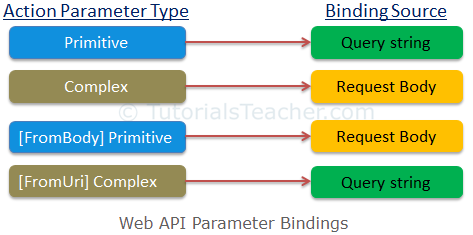
<http://localhost/api/student?age=25>

By default, Web API will get the id parameter from query string and student parameter from the request body.

Parameter binding for Put and Patch method will be the same as post method in Web API.

1. **[FromUri] and [FromBody]**

Use [FromUri] attribute to force Web API to get the value of complex type from query string and [FromBody] attribute to get the value from the request body, opposite to the default rules.



# 7.Action Return Type

The Action method return will be embedded in the Web API response sent to the client.

The Web API action method can have following return types:

1. Void
2. Primitive type or complex type
3. **HttpResponseMessage**

Web API controller always returns an object of HttpResponseMessage to the hosting infrastructure.

The advantage of sending HttpResponseMessage from an action method is that you can configure a response your way.

1. **IHttpActionResult**

The IHttpActionResult was introduced in Web API 2 (.NET 4.5), which is similar to ActionResult class in ASP.NET MVC.

# 8.Web API Request/Response Data Formats

Media data (aka MIME type) specifies the format of the data as type/subtype e.g. text/html, application/json, image/jpeg etc.

In HTTP request, MIME type is specified in the request header using Accept and Content-type attribute.

* The Accept header attribute specifies the format of response data which the client expects.
* The Content-type attribute specifies the format of the data in the request body so that receiver can parse it into appropriate format.

Accept and Content-type are both headers sent from a client (browser) to a service.

* Accept header is a way for a client to specify the media type of the response content it is expecting.
* Content-type is a way to specify the media type of request being sent from the client to the server.

# 9.Media Type Formatters

The Web API handles JSON and XML by default. But, how does it handle other different formats?

Media type formatters are classes responsible for serializing request/response data so that Web API can understand the request data format and send data in the format which client expects.

# 10.NOTE

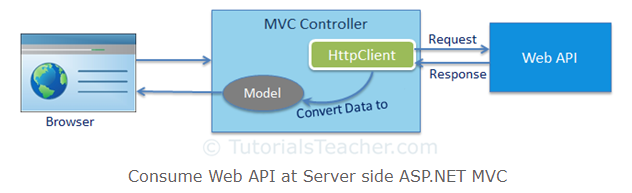
(1). We should not return EF entity objects from the Web API. It is recommended to return DTO (Data Transfer Object) from Web API. If we create Web API project with MVC, we can also use MVC classes which will be used in both MVC and Web API.

ViewModel classes or DTO classes are just for data transfer from Web API controller to clients.

(2). ASP.NET Web API controller can include multiple action GET/PUT… methods with different parameters and types.

# 11.Consume Web API in ASP.NET MVC

We can use HttpClient in the MVC controller to consume Web API in ASP.NET MVC. HttpClient sends a request to the Web API and receives a response. Then we need to convert the response data to a model and then render it into a view.



NOTE: AngularJS or any other JavaScript framework can be used in MVC view and can access Web API directly from the view using AJAX.

# 12. Consume Web API in AngularJS

Web API can be accessed directly from the UI at client side using AJAX capabilities of any JavaScript framework.

# 13.Example explanation

* Create new project ASP.NET web Application – select Web API – tick MVC+ Web API
* Create ADO.NET Entity Data Model: right click the project – Add New Item – select ADO.NET Entity Data Model – connection to the database

Usually, the DbContext is in the xxx.Context.cs file.

* Create viewmodel classes or DTO (Data Transfer Object) for data transfer from Web API controller to client.
* Create Web API Controller which extends ApiController class.
  + Create GET Action
  + Create POST Action
  + Create PUT Action
  + Create DELETE Action

After that, you can use Postman to test all the action methods.

* Consume Web API in ASP.NET MVC
  + Create MVC controller which extends Controller class.
  + Create corresponding view

C:\Users\linghui.ye\source\repos\MyWebApi