In this video we will discuss HTTP PUT, POST and DELETE verbs. This is continuation to [Part 2](http://csharp-video-tutorials.blogspot.com/2016/08/creating-web-api-project.html). Please watch [Part 2](http://csharp-video-tutorials.blogspot.com/2016/08/creating-web-api-project.html) from [ASP.NET Web API tutorial](https://www.youtube.com/playlist?list=PL6n9fhu94yhW7yoUOGNOfHurUE6bpOO2b) before proceeding.   
  
  
  
When we talk about a database table row, these are the following 4 actions that we can perform on the row  
C - Create a row  
R - Read a row  
U - Update a row  
D - Delete a row  
  
In the context of an ASP.NET Web API resource these 4 actions correspond to GET, POST, PUT and DELETE as shown in the table below

| **CRUD** | **HTTP Verb** |
| --- | --- |
| Create | POST |
| Read | GET |
| Update | PUT |
| Delete | DELETE |

Let us now understand some terms and concepts related to HTTP request and response system.  
  
**Request Verbs :** These HTTP verbs (GET, POST, PUT & DELETE) describe what should be done with the resource. For example do you want to create, read, update or delete an entity. GET, PUT, POST and DELETE http verbs are the most commonly used one's. For the complete list of the HTTP verbs, please check <http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>  
  
**Request Header :** When a client sends request to the server, the request contains a header and a body. The request header contains additional information such as what type of response is required. For example, do you want the response to be in XML or JSON.  
  
**Request Body :** Request Body contains the data to send to the server. For example, a POST request contains the data for the new item that you want to create. The data format may be in XML or JSON.  
  
**Response Body :** The Response Body contains the data sent as response from the server. For example, if the request is for a specific product, the response body includes product details either in XML or JSON format.  
  
**Response Status codes :** These are the HTTP status codes, that give the client details on the status of the request. Some of the common status codes are 200/OK, 404/Not Found, 204/No Content. For the complete list of HTTP status codes and what they mean, please visit <http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>  
  
We will use a tool called fiddler to perform POST, PUT & DELETE actions. You can download fiddler from the following link  
<https://www.telerik.com/download/fiddler>  
  
Modify the **ValuesController**as shown below, so that it can support POST, PUT and DELETE actions.

public class ValuesController : ApiController

{

    static List<string> strings = new List<string>()

    {

        "value0", "value1", "value2"

    };

    // GET api/values

    public IEnumerable<string> Get()

    {

        return strings;

    }

    // GET api/values/5

    public string Get(int id)

    {

        return strings[id];

    }

    // POST api/values

    public void Post([FromBody]string value)

    {

        strings.Add(value);

    }

    // PUT api/values/5

    public void Put(int id, [FromBody]string value)

    {

        strings[id] = value;

    }

    // DELETE api/values/5

    public void Delete(int id)

    {

        strings.RemoveAt(id);

    }

}

Post(), Put() and Delete() methods in our **ValuesController**are returning void. That is the reason we are getting status code 204 No Content. An action that returns void will send status code 204 No Content. With ASP.NET Web API, we can control what status codes these methods return. We will discuss how to do that in a later video.