# Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.
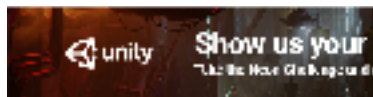
Support us    .Net Basics    C#  SQL    ASP.NET    ADO.NET    MVC    Slides    C# Programs    Subscribe    Buy DVD    G+

## HTTP GET PUT POST DELETE

**Suggested Videos**
Part 1 - What is ASP.NET Web API
Part 2 - Creating a Web API Project

In this video we will discuss HTTP PUT, POST and DELETE verbs. This is continuation to Part 2. Please watch Part 2 from ASP.NET Web API tutorial 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.

**SOLID Principles**
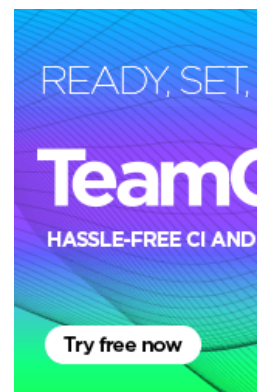
SOLID Introduction | Text | Slides

Single Responsibility | Text | Slides

**Important Videos**

The Gift of Education

Web application for your business

How to become .NET developer

Resources available to help you

**Dot Net Video Tutorials**

Angular CLI Tutorial

Angular 2 Tutorial

Design Patterns

ASP.NET Web API

**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.

```csharp
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.

Bootstrap

AngularJS Tutorial

jQuery Tutorial

JavaScript with ASP.NET Tutorial

JavaScript Tutorial

Charts Tutorial

LINQ

LINQ to SQL

LINQ to XML

Entity Framework

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Dot Net Interview Questions

**Slides**

Entity Framework

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

**Java Video Tutorials**

Part 1 : Video | Text | Slides

## Interview Questions

C#

SQL Server

Written Test

## CBSE Class 9 Maths

Number Systems

## CPT Tutorial

G+

## 7 comments:

**a gouri kumar** August 30, 2016 at 9:52 PM

Nice Video

Reply

**Muhammad Rehbar Sheikh** August 31, 2016 at 1:38 AM

Thank you venkat sir. Good Explanation.

Reply

**Adnane Adnan** September 1, 2016 at 6:07 AM

thank you so much sir :D

Reply

**Atul Tripathi** September 2, 2016 at 1:59 PM

Hi Sir, Can you please help in creating a SQL script to delete old files on shared drives. It has to be done via SQL.

Reply

**mehedi hasan** October 15, 2016 at 10:37 AM

Whats the need of PUT and Delete as we also do this using POST method?
Can you please explain the difference between put and post?

Reply

**Muhammad Imran** December 1, 2016 at 10:03 AM

Put is Used To Add data row in table and post is used to update that data rowl

Reply

**himanshu pareek** May 29, 2017 at 10:38 AM

"Postman" plugin is very useful in Chrome to test the webapi.

Reply

Enter your comment...

**Comment as:**    Unknown (Goo  ▾                              Sign out

Publish        Preview                                    ☐ Notify me

If you like this website, please share with your friends on facebook and Google+ and
recommend us on google using the g+1 button on the top right hand corner.

## Links to this post
Create a Link

**Newer Post**                    **Home**                    **Older Post**

Subscribe to: Post Comments (Atom)

Powered by Blogger.