

# 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+

## Implementing post method in ASP.NET Web API

### Suggested Videos

Part 3 - HTTP GET PUT POST DELETE

Part 4 - ASP.NET Web API and SQL Server

Part 5 - ASP.NET Web API Content Negotiation



In this video we will discuss **implementing POST method in ASP.NET Web API**. Post allows us to create a new item.



We want to add a new Employee to the Employees table. Include the following Post() method in EmployeesController. Notice the Employee object is being passed as parameter to the Post method. The Employee parameter is decorated with [FromBody] attribute. This tells Web API to get employee data from the request body.

```
public void Post([FromBody] Employee employee)
{
    using (EmployeeDBEntities entities = new EmployeeDBEntities())
    {
        entities.Employees.Add(employee);
        entities.SaveChanges();
    }
}
```

### At this point build the solution. Fireup Fiddler and issue a Post request

1. Set the HTTP verb to POST
2. Content-Type: application/json. This tells that we are sending JSON formatted data to the server
3. In the Request Body, include the employee object that you want to add to the database in JSON format
4. Finally click execute

Ads by Google

Ads by Google

C# Tutorial

API Manager

Create Video

Delete Web

Infinix



**PRAGIM**  
TECHNOLOGIES  
Training + Placements

Best software training and placements in marathahalli, bangalore. For further details please call 09945699393.

### 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

Parsed	Raw	Scratchpad	Options
POST <input type="text" value="http://localhost:35171/api/employees/"/>			
Host: localhost:35171 Content-Type: application/json			
<div>Request Body</div> <pre>{   "FirstName": "Sam",   "LastName": "Wicht",   "Gender": "Male",   "Salary": 54000 }</pre>			

This works fine and adds the employee to the database as expected. The problem here is that since the return type of the Post() method is void, we get status code of **204 No Content**. When a new item is created, we should actually be returning status code 201 Item Created. With 201 status code we should also include the location i.e URI of the newly created item. To achieve this, change the implementation of the Post() method as shown below.

```
public HttpResponseMessage Post([FromBody] Employee employee)
{
    try
    {
        using (EmployeeDBEntities entities = new EmployeeDBEntities())
        {
            entities.Employees.Add(employee);
            entities.SaveChanges();

            var message = Request.CreateResponse(HttpStatusCode.Created, employee);
            message.Headers.Location = new Uri(Request.RequestUri +
                employee.ID.ToString());

            return message;
        }
    }
    catch (Exception ex)
    {
        return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ex);
    }
}
```

At this point, issue another Post request from fiddler. Notice in the response header we have status code **201 Created** and also the location i.e the URI of the newly created item.

[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](#)

Get SyntaxView	Transformer	Headers	TextView
Response Headers			
HTTP/1.1 201 Created			
<b>Cache</b> Cache-Control: no-cache Date: Tue, 06 Sep 2016 13:59:26 GMT Expires: -1 Pragma: no-cache			
<b>Entity</b> Content-Length: 83 Content-Type: application/json; charset=utf-8			
<b>Miscellaneous</b> Server: Microsoft-IIS/10.0 X-AspNet-Version: 4.0.30319 X-Powered-By: ASP.NET X-SourceFiles: =?UTF-8?B?YzpcRW1wbG95ZWVTZX			
<b>Transport</b> Location: <a href="http://localhost:35171/api/employees/9">http://localhost:35171/api/employees/9</a>			

Let's also modify `Get(int id)` method. At the moment when we issue a request for an employee id that does not exist, we get null back and the status code is **200 OK**. According to the standards of REST when an item is not found we should be returning **204 Not Found**. To achieve this modify the `Get(int id)` method as shown below.

```
public HttpResponseMessage Get(int id)
{
    using (EmployeeDBEntities entities = new EmployeeDBEntities())
    {
        var entity = entities.Employees.FirstOrDefault(e => e.ID == id);
        if (entity != null)
        {
            return Request.CreateResponse(HttpStatusCode.OK, entity);
        }
        else
        {
            return Request.CreateErrorResponse(HttpStatusCode.NotFound,
                "Employee with Id " + id.ToString() + " not found");
        }
    }
}
```

At this point, when we issue a request for an employee with ID = 101 which does not exist we get a **404** along with the message **"Employee with Id 101 not found"**

#### Here are the important points to remember

- If a method return type is `void`, by default status code **204 No Content** is returned.
- When a new item is created, we should be returning status code **201 Item Created**.
- With **201 status code** we should also include the location i.e URI of the newly created item.
- When an item is not found, instead of returning **NULL** and status code **200 OK**, return **404 Not Found** status code along with a meaningful message such as **"Employee with Id = 101 not found"**

Part 2 : [Video](#) | [Text](#) | [Slides](#)

Part 3 : [Video](#) | [Text](#) | [Slides](#)

#### Interview Questions

C#

SQL Server

Written Test

#### CBSE Class 9 Maths

Number Systems

#### CPT Tutorial

Part 1 : [Video](#) | [Text](#) | [Slides](#)

# WWW.PRAGIMTECH.COM

CLICK HERE FOR THE FULL  
ASP.NET WEB API TUTORIAL PLAYLIST

[HTTP://FACEBOOK.COM/PRAGIMTECH](http://facebook.com/pragimtech)



5 comments:



**Zokrit** September 29, 2016 at 7:49 AM

Thank you for this great tutorial. How can I get the last inserted Id if I am using a stored procedure for inserting. I added an output parameter to the stored proc but it always returns 0.

[Reply](#)



**afshid** October 12, 2016 at 1:20 PM

Hi

Many thanks for these great tutorials. I did the same as you did but when I tried to use post method to insert new employee I got this error: "HTTP/1.1 405 Method Not Allowed" Could you tell my how I can fix it?

[Reply](#)



**Unknown** January 11, 2017 at 3:05 PM

Hi

I am confused by the SQL reissue that you do... can you expand?

[Reply](#)

**Anonymous** March 7, 2017 at 7:26 AM

For Get request when an item is not found we have to return 404 Not Found. Typo error is there

[Reply](#)



**Md Shahbaz** November 15, 2017 at 2:19 AM

using above post code i'm getting wrong url  
Location: http://localhost:61438/api/Employee10  
using VS2013.

[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.