

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
 key princess castle | Making princess castle | Sewing Unicorn | Build lego smoothie van | Solve frozen elsa puzzle | Taj mahal jigsaw puzzle | Statue of liberty

FromBody and FromUri in Web API

Suggested Videos

Part 9 - Implementing PUT method in ASP.NET Web API

Part 10 - Custom method names in ASP.NET Web API

Part 11 - ASP.NET Web API query string parameters



In this video we will discuss the use of `FromBody` and `FromUri` attributes. Let us understand their use with an example.



Consider the following `Put()` method. This method updates the specified Employee details.

```
public HttpResponseMessage Put(int id, Employee employee)
{
    try
    {
        using (EmployeeDBEntities entities = new EmployeeDBEntities())
        {
            var entity = entities.Employees.FirstOrDefault(e => e.ID == id);
            if (entity == null)
            {
                return Request.CreateErrorResponse(HttpStatusCode.NotFound,
                    "Employee with Id " + id.ToString() + " not found to update");
            }
            else
            {
                entity.FirstName = employee.FirstName;
                entity.LastName = employee.LastName;
                entity.Gender = employee.Gender;
                entity.Salary = employee.Salary;

                entities.SaveChanges();

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

Ads by Google

sql tutorial

Ads by Google

rest api

web api c#

angular 2



PRAGIM
TECHNOLOGIES
 Training + Placements

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

Complete Tutorials

Angular tutorial for beginners

Angular 5 Tutorial for beginners

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

ASP.NET Core Tutorial

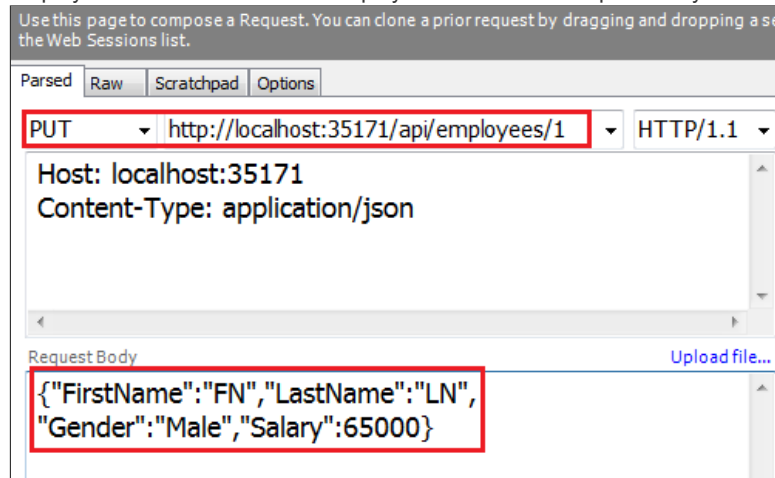
Angular 6 Tutorial

Angular CRUD Tutorial

Angular CLI Tutorial

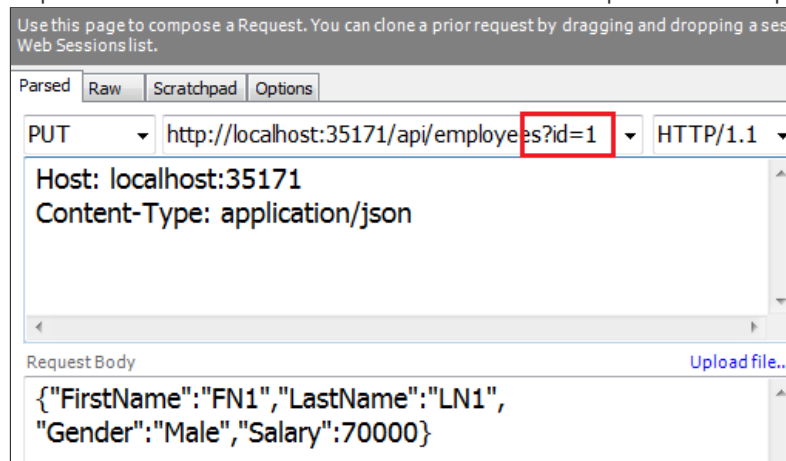
To update employee details whose Id is 1 we issue a Put request to the following URI
<http://localhost/api/employees/1>

If you are using Fiddler, the PUT request is as shown below. Notice the Id of the employee is in the URI and the employee data is in the request body.



At this point if we execute the request, the employee data is updated as expected.

Now let's include the Id as a query string parameter. In the first request Id is specified as part of route data. Notice in Fiddler we have included id parameter as a query string.



When we execute this request, the update succeeds as expected.

When a PUT request is issued, Web API maps the data in the request to the PUT method parameters in the EmployeesController. This process is called Parameter Binding.

Now let us understand the default convention used by Web API for binding parameters.

1. If the parameter is a simple type like int, bool, double, etc., Web API tries to get the value from the URI (Either from route data or Query String)
2. If the parameter is a complex type like Customer, Employee etc., Web API tries to get the value from the request body

So in our case, the id parameter is a simple type, so Web API tries to get the value from the request URI. The employee parameter is a complex type, so Web API gets the value from the request body.

We can change this default **parameter binding** process by using `[FromBody]` and `[FromUri]` attributes. Notice in the example below

[Angular 2 Tutorial](#)

[Design Patterns](#)

[SOLID Principles](#)

[ASP.NET Web API](#)

[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](#)

1. We have decorated id parameter with [FromBody] attribute, this forces Web API to get it from the request body
2. We have decorated employee parameter with [FromUri] attribute, this forces Web API to get employee data from the URI (i.e Route data or Query String)

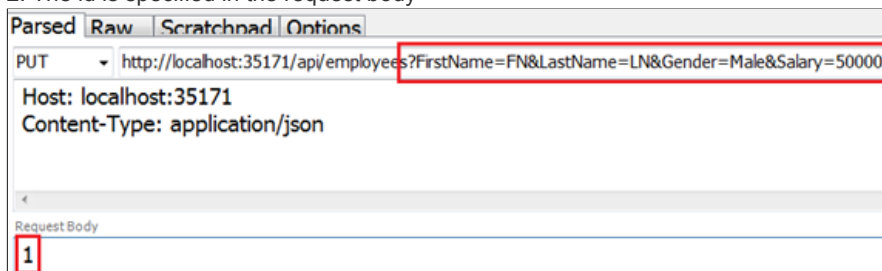
```
public HttpResponseMessage Put([FromBody]int id, [FromUri]Employee employee)
{
    try
    {
        using (EmployeeDBEntities entities = new EmployeeDBEntities())
        {
            var entity = entities.Employees.FirstOrDefault(e => e.ID == id);
            if (entity == null)
            {
                return Request.CreateErrorResponse(HttpStatusCode.NotFound,
                    "Employee with Id " + id.ToString() + " not found to update");
            }
            else
            {
                entity.FirstName = employee.FirstName;
                entity.LastName = employee.LastName;
                entity.Gender = employee.Gender;
                entity.Salary = employee.Salary;

                entities.SaveChanges();

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

Here is the request from Fiddler

1. Employee data is specified in the URI using query string parameters
2. The id is specified in the request body



When we execute the request the update succeeds as expected

WWW.PRAGIMTECH.COM

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

HTTP://FACEBOOK.COM/PRAGIMTECH

2 comments:

[ASP.NET MVC](#)

[Visual Studio Tips and Tricks](#)

Java Video Tutorials

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

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

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

Interview Questions

[C#](#)

[SQL Server](#)

[Written Test](#)



Noor January 23, 2017 at 2:20 AM

i get 405 error when ever iam trying to update or delete

[Reply](#)

Replies



Akhil Chintu June 3, 2017 at 3:07 AM

Noor, I too faced the same issue.

We need to Add the Verb types in the appplication config file, as the default HTTP Verbs are Just GET,HEAD,POST,DEBUG. So to allow Delete and Put Verb we need to Add Verbs in the File.

To do that .. Open Run Command and Paste the following path.

%userprofile%\documents\iisexpress\config\

Click on Enter.

There you will see Application.config file. Open it and modify bellow name Value Property Verb Values as Shown Below.

[Reply](#)

Enter your comment...



Comment as: **ahm7dkhalifa@** ▼

[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.