In this video we will discuss **how to generate links using route names in ASP.NET Web API**.    
  
  
  
Consider the following **StudentsController**. 

[RoutePrefix("api/students")]

public class StudentsController : ApiController

{

    static List<Student> students = new List<Student>()

    {

        new Student() { Id = 1, Name = "Tom" },

        new Student() { Id = 2, Name = "Sam" },

        new Student() { Id = 3, Name = "John" }

    };

    [Route("{id:int}")]

    public Student Get(int id)

    {

        return students.FirstOrDefault(s => s.Id == id);

    }

    public HttpResponseMessage Post(Student student)

    {

        students.Add(student);

        var response = Request.CreateResponse(HttpStatusCode.Created);

        return response;

    }

}

When a new item is created using the Post() method, along with status code 201 we also need to 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(Student student)

{

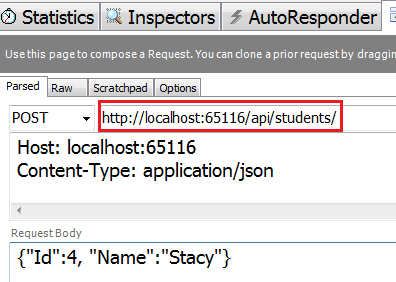
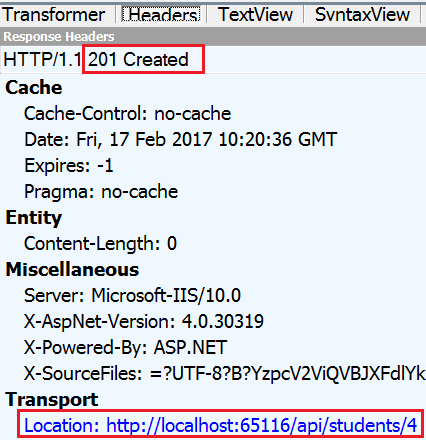
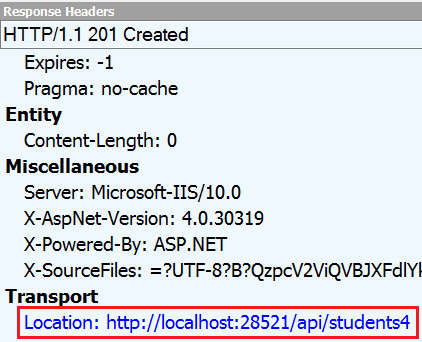
    students.Add(student);

    var response = Request.CreateResponse(HttpStatusCode.Created);

    response.Headers.Location = new Uri(Request.RequestUri + student.Id.ToString());

    return response;

}

At this point, issue a Post request using fiddler. Please note there is a forward slash (/) at the end of the URL to which we are issuing a POST request.   
   
  
Notice in the response header we have status code 201 Created and also the location i.e the URI of the newly created item.   
   
  
The link generation for the newly created item worked as expected because we have included a forward slash (/) in Fiddler when we issued the POST request. If we omit the forward slash at the end of the URI and issue the POST request, the link for the newly created item is not generated correctly.   
   
  
You might be thinking we can fix this issue by adding a forward slash in the POST() method as shown below. Please note we are appending a forward slash (/) to the RequestUri and then the ID of the student. 

public HttpResponseMessage Post(Student student)

{

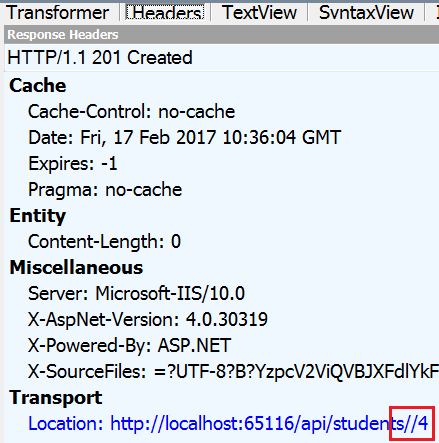
    students.Add(student);

    var response = Request.CreateResponse(HttpStatusCode.Created);

    response.Headers.Location = new Uri(Request.RequestUri + "/" + student.Id.ToString());

    return response;

}

The problem with this approach is that, we get 2 forward slashes (//) in the generated link, if we include a forward slash at the end of the URL to which we are posting in Fiddler   
   
  
We can very easily solve this issue of malformed URLs using Route Names. Every route in Web API has a name. We can use the route name to correctly generate the links.   
  
To specify a name for a route, set the Name property on the Route attribute.  

[Route("{id:int}", Name = "GetStudentById")]

public Student Get(int id)

{

    return students.FirstOrDefault(s => s.Id == id);

}

We can then use the Route name to generate the link as shown below. Please note we are using the route name ("GetStudentById"), to generate the link for the newly created item. 

public HttpResponseMessage Post(Student student)

{

    students.Add(student);

    var response = Request.CreateResponse(HttpStatusCode.Created);

    response.Headers.Location = new

        Uri(Url.Link("GetStudentById", new { id = student.Id }));

    return response;

}

At this point, the link generation for the newly created item works as expected irrespective of whether we have a forward slash (/) or not in Fiddler when issuing the POST request.  
  
So in summary, to generate links in ASP.NET Web API  
1. Set a name for the route using the Name property of the [Route] attribute

[Route("{id:int}", Name = "GetStudentById")]

public Student Get(int id)

{

    return students.FirstOrDefault(s => s.Id == id);

}

2. Use the name of the route to generate the link 

public HttpResponseMessage Post(Student student)

{

    students.Add(student);

    var response = Request.CreateResponse(HttpStatusCode.Created);

    response.Headers.Location = new

                Uri(Url.Link("GetStudentById", new { id = student.Id }));

    return response;

}