In this video we will discuss **versioning a Web API Service using a custom version header**. This is continuation to [Part 36](http://csharp-video-tutorials.blogspot.com/2017/02/web-api-versioning-using-querystring.html). Please watch [Part 36](http://csharp-video-tutorials.blogspot.com/2017/02/web-api-versioning-using-querystring.html) from [Web API tutorial](https://www.youtube.com/playlist?list=PL6n9fhu94yhW7yoUOGNOfHurUE6bpOO2b)before proceeding.   
  
In our previous video, we have implemented a CustomControllerSelector. At the moment, this CustomControllerSelector is retrieving the version number for a query string parameter.   
  
  
  
To implement versioning using a custom version header, all we have to do is change the logic slightly int the CustomControllerSelector class to read the version number from the custom version header instead of from a query string parameter   
  
  
  
The changes that are required are commented, so it is self-explanatory.

using System.Linq;

using System.Net.Http;

using System.Web.Http;

using System.Web.Http.Controllers;

using System.Web.Http.Dispatcher;

namespace WebAPI.Custom

{

    public class CustomControllerSelector : DefaultHttpControllerSelector

    {

        private HttpConfiguration \_config;

        public CustomControllerSelector(HttpConfiguration config) : base(config)

        {

            \_config = config;

        }

        public override HttpControllerDescriptor

            SelectController(HttpRequestMessage request)

        {

            var controllers = GetControllerMapping();

            var routeData = request.GetRouteData();

            var controllerName = routeData.Values["controller"].ToString();

            // Default the version number to 1

            string versionNumber = "1";

            // Comment the code that gets the version number from Query String

            // var versionQueryString = HttpUtility.ParseQueryString(request.RequestUri.Query);

            // if (versionQueryString["v"] != null)

            // {

            //     versionNumber = versionQueryString["v"];

            // }

            // Get the version number from Custom version header

            // This custom header can have any name. We have to use this

            // same header to specify the version when issuing a request

            string customHeader = "X-StudentService-Version";

            if (request.Headers.Contains(customHeader))

            {

                versionNumber = request.Headers.GetValues(customHeader).FirstOrDefault();

            }

            HttpControllerDescriptor controllerDescriptor;

            if (versionNumber == "1")

            {

                controllerName = string.Concat(controllerName, "V1");

            }

            else

            {

                controllerName = string.Concat(controllerName, "V2");

            }

            if (controllers.TryGetValue(controllerName, out controllerDescriptor))

            {

                return controllerDescriptor;

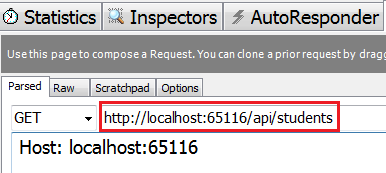
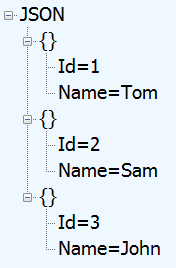
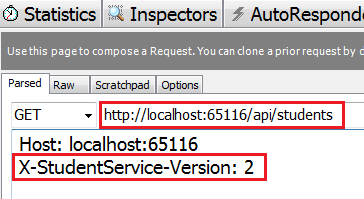
            }

            return null;

        }

    }

}

At this point, build the solution and issue a request to /api/students using Fiddler. Notice we have not specified our custom version header in the request. We get back StudentV1 objects, this is because we have set version 1 as the default in our code.   
  
   
  
Response from the service   
   
  
Let's issue another request using our custom version header as shown below.   
  
   
  
Response from the service. Since we have specified the version as 2, we got StudentV2 objects back.   
