In this video we will discuss **versioning a Web API Service using a QueryString parameter**. This is continuation to [Part 35](http://csharp-video-tutorials.blogspot.com/2017/02/web-api-versioning-using-uri.html). Please watch [Part 35](http://csharp-video-tutorials.blogspot.com/2017/02/web-api-versioning-using-uri.html) from [Web API tutorial](https://www.youtube.com/playlist?list=PL6n9fhu94yhW7yoUOGNOfHurUE6bpOO2b)before proceeding.   
  
**Here is what we want**

|  |  |
| --- | --- |
| **URI** | **Should Return** |
| /api/students?v=1 | Version 1 Students |
| /api/students?v=2 | Version 2 Students |

Before we implement versioning using a querystring parameter. First let's understand how a controller is selected when a request is issued to a web api service. For example, let us understand how a controller is selected when a rquest is issued to the following URI  
/api/students/1  
  
In Web API, there is a class called DefaultHttpControllerSelector. This class has a method called SelectController() that selects the controller based on the information it has in the URI.   
  
  
  
**In the URI we have**  
1. The name of the controller, in this case students   
2. The id parameter value, in this case 1  
  
So from the URI, the SelectController() method takes the name of the controller in this case "Students" and finds "StudentsController" and returns it. This is the default implementation that we get out of the box.  
  
This default implementation will not work for us because, in our service we do not have controller that is named StudentsController. Instead we have  
1. StudentsV1Controller and  
2. StudentsV2Controller  
  
When a request is issued to the following URI, depending on the query string parameter "v" value we want to select the controller. If the value is 1, select StudentsV1Controller, and if it is 2, then select StudentsV2Controller.  
/api/students?v=1 

|  |  |
| --- | --- |
| **Query String "v" value** | **Controller to Select** |
| v=1 | StudentsV1Controller |
| v=2 | StudentsV2Controller |

Here are the steps to version Web API service using a query string parameter  
  
**Step 1 :** Since the default controller selector implementation provided by Web API does not work for us, we have to provide our own custom controller selector implementation. To do this  
**1.** Add a folder to the web api project. Name it "Custom"  
**2.** Add a class file to the folder. Name it "CustomControllerSelector". Copy and paste the following code. I have commented the code where necessary, so it is self explanatory

using System.Net.Http;

using System.Web;

using System.Web.Http;

using System.Web.Http.Controllers;

using System.Web.Http.Dispatcher;

namespace WebAPI.Custom

{

    // Derive from the DefaultHttpControllerSelector class

    public class CustomControllerSelector : DefaultHttpControllerSelector

    {

        private HttpConfiguration \_config;

        public CustomControllerSelector(HttpConfiguration config) : base(config)

        {

            \_config = config;

        }

        public override HttpControllerDescriptor

            SelectController(HttpRequestMessage request)

        {

            // Get all the available Web API controllers

            var controllers = GetControllerMapping();

            // Get the controller name and parameter values from the request URI

            var routeData = request.GetRouteData();

            // Get the controller name from route data.

            // The name of the controller in our case is "Students"

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

            // Default version number to 1

            string versionNumber = "1";

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

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

            {

                versionNumber = versionQueryString["v"];

            }

            if (versionNumber == "1")

            {

                // if version number is 1, then append V1 to the controller name.

                // So at this point the, controller name will become StudentsV1

                controllerName = controllerName + "V1";

            }

            else

            {

                // if version number is 2, then append V2 to the controller name.

                // So at this point the, controller name will become StudentsV2

                controllerName = controllerName + "V2";

            }

            HttpControllerDescriptor controllerDescriptor;

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

            {

                return controllerDescriptor;

            }

            return null;

        }

    }

}

**Step 2 :** The next thing that we need to do is, replace the default controller selector with our custom controller selector. This is done in WebApiConfig.cs file. Notice we are replacing IHttpControllerSelector, with our CustomControllerSelector. DefaultHttpControllerSelector implements IHttpControllerSelector, so that is the reason we are replacing IHttpControllerSelector.

config.Services.Replace(typeof(IHttpControllerSelector),

    new CustomControllerSelector(config));

**Step 3 :** Include the following default route in WebApiConfig.cs

config.Routes.MapHttpRoute(

    name: "DefaultRoute",

    routeTemplate: "api/{controller}/{id}",

    defaults: new { id = RouteParameter.Optional }

);

**Step 4 :** Remove [Route] attribute, from action methods in StudentsV1Controller and StudentsV2Controller   
