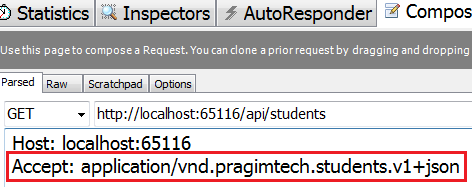
In this video we will discuss versioning a Web API Service using vendor specific media types. This is continuation to [Part 38](http://csharp-video-tutorials.blogspot.com/2017/03/web-api-versioning-using-accept-header.html). Please watch [Part 38](http://csharp-video-tutorials.blogspot.com/2017/03/web-api-versioning-using-accept-header.html) from [Web API tutorial](https://www.youtube.com/playlist?list=PL6n9fhu94yhW7yoUOGNOfHurUE6bpOO2b) before proceeding.   
  
  
  
So here is what we want to be able to do. Instead of using the standard media types like application/xml or application/json, we want to use our custom media type as shown below in fillder   
   
  
  
  
Notice in the media type, we have the version of the service we want. Custom media types have vnd prefix. vnd indicates that it is a vendor specific media type. So from our CustomControllerSelector class we want to read the version number from the custom media type we have specified.   
  
The changes that are required are commented, so it is self-explanatory. 

using System.Linq;

using System.Net.Http;

using System.Text.RegularExpressions;

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();

            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

            //string customHeader = "X-StudentService-Version";

            //if (request.Headers.Contains(customHeader))

            //{

            //    // If X-StudentService-Version:1 is specified twice in the request

            //    // then in versionNumber variable will get a value of "1,1"

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

            //    // Check if versionNumber string contains a comma, and take only

            //    // the first number from the comma separated list of version numbers

            //    if (versionNumber.Contains(","))

            //    {

            //        versionNumber = versionNumber.Substring(0, versionNumber.IndexOf(","));

            //    }

            //}

            // Get the version number from the Accept header

            // Users can include multiple Accept headers in the request

            // Check if any of the Accept headers has a parameter with name version

            //var acceptHeader = request.Headers.Accept.Where(a => a.Parameters

            //                    .Count(p => p.Name.ToLower() == "version") > 0);

            //// If there is atleast one header with a "version" parameter

            //if (acceptHeader.Any())

            //{

            //    // Get the version parameter value from the Accept header

            //    versionNumber = acceptHeader.First().Parameters

            //                    .First(p => p.Name.ToLower() == "version").Value;

            //}

            // Get the version number from the Custom media type

            // Use regular expression for mataching the pattern of the media

            // type. We have given a name for the matched group that contains

            // the version number. This enables us to retrieve the version number

            // using the group name("version") instead of ZERO based index

            string regex =

                @"application\/vnd\.pragimtech\.([a-z]+)\.v(?<version>[0-9]+)\+([a-z]+)";

            // Users can include multiple Accept headers in the request.

            // Check if any of the Accept headers has our custom media type by

            // checking if there is a match with regular expression specified

            var acceptHeader = request.Headers.Accept

                .Where(a => Regex.IsMatch(a.MediaType, regex,RegexOptions.IgnoreCase));

            // If there is atleast one Accept header with our custom media type

            if (acceptHeader.Any())

            {

                // Retrieve the first custom media type

                var match = Regex.Match(acceptHeader.First().MediaType,

                    regex, RegexOptions.IgnoreCase);

                // From the version group, get the version number

                versionNumber = match.Groups["version"].Value;

            }

            HttpControllerDescriptor controllerDescriptor;

            if (versionNumber == "1")

            {

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

            }

            else

            {

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

            }

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

            {

                return controllerDescriptor;

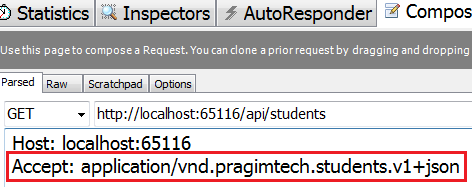
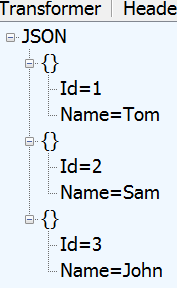
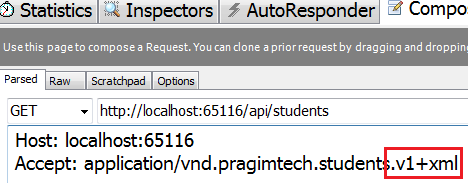
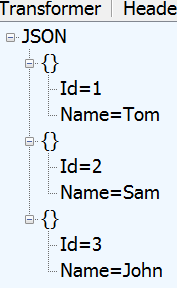
            }

            return null;

        }

    }

}

So at this point, if we issue a request using our custom media type from fiddler, we get version 1 of student objects in JSON format. If you specify version 2 (v2) instead of version 1 (v1), we get version 2 student objects as expected.   
   
  
Response from the web api service   
   
  
However, if we specify that we want **xml**format instead of **json**in the request as shown below. We still get **JSON**formatted result instead of **XML**formatted result.    
   
  
Response from the web api service   
   
  
This is because we have not added our custome media types to the respective media type formatters (JsonFormatter and XmlFormatter).   
  
To add the custom media types to the **JsonFormatter**, include the following 2 lines of code in WebApiConfig.cs file

config.Formatters.JsonFormatter.SupportedMediaTypes

    .Add(new MediaTypeHeaderValue("application/vnd.pragimtech.students.v1+json"));

config.Formatters.JsonFormatter.SupportedMediaTypes

    .Add(new MediaTypeHeaderValue("application/vnd.pragimtech.students.v2+json"));

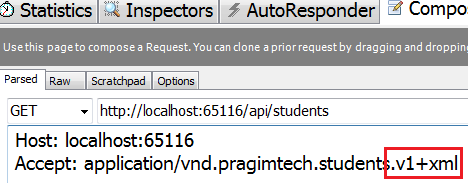
To add the custom media types to the **XmlFormatter**, include the following 2 lines of code in WebApiConfig.cs file

config.Formatters.XmlFormatter.SupportedMediaTypes

    .Add(new MediaTypeHeaderValue("application/vnd.pragimtech.students.v1+xml"));

config.Formatters.XmlFormatter.SupportedMediaTypes

    .Add(new MediaTypeHeaderValue("application/vnd.pragimtech.students.v2+xml"));

So at this point, let's issue another request using our custom media type from fiddler   
   
  
Notice we get version 1 of student objects in XML format as expected   
