

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 G+

ASP.NET Web API MediaTypeFormatter

Suggested Videos

Part 3 - HTTP GET PUT POST DELETE

Part 4 - ASP.NET Web API and SQL Server

Part 5 - ASP.NET Web API Content Negotiation

1 Oracle PI Sql Download the 30 day trail version for PL

2 Start Download - View PDF Convert From Doc to App! download from

In this video we will discuss

- What is MediaTypeFormatter
- How to return only JSON from ASP.NET Web API Service irrespective of the Accept header value
- How to return only XML from ASP.NET Web API Service irrespective of the Accept header value
- How to return JSON instead of XML from ASP.NET Web API Service when a request is made from the browser
- Point you to an article that describes how to return CSV formatted data from ASP.NET Web API

1 Oracle PI Sql Download the 30 day trail version for PL

2 هل أنت وحيد ؟ - تعرف على صديقات جديدات و لوحتك الإلكترونية / اوبراشات / promo.w

What is MediaTypeFormatter

[MediaTypeFormatter](#) is an abstract class from which [JsonMediaTypeFormatter](#) and [XmlMediaTypeFormatter](#) classes inherit from. [JsonMediaTypeFormatter](#) handles JSON and [XmlMediaTypeFormatter](#) handles XML.

How to return only JSON from ASP.NET Web API Service irrespective of the Accept header value

Include the following line in **Register()** method of **WebApiConfig.cs** file in **App_Start** folder. This line of code completely removes **XmlFormatter** which forces ASP.NET Web API to always return JSON irrespective of the Accept header value in the client request. Use this technique when you want your service to support only JSON and not XML.

With this change, irrespective of the Accept header value (application/xml or application/json), the Web API service is always going to return JSON.

```
config.Formatters.Remove(config.Formatters.XmlFormatter);
```

How to return only XML from ASP.NET Web API Service irrespective of the Accept header value

Include the following line in **Register()** method of **WebApiConfig.cs** file in **App_Start** folder. This line of code completely removes **JsonFormatter** which forces ASP.NET Web API to always return XML irrespective of the Accept header value in the client request. Use this technique when you want your service to support only XML and not JSON.

Ads by Google

Ads by Google

C# Tutorial

Create Video

Video Call

Video How To



Threat Intelligence Man:

Cyber Security Solutions

Phishing Solutions, Threat Intelligence, Brand Protection And More

lookingglasscyber.com/Cyber-Security

PRAGIM
TECHNOLOGIES
Training + Placements

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

SOLID Principles

SOLID Introduction | Text | Slides

Single Responsibility | Text | Slides

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

Angular CLI Tutorial

Angular 2 Tutorial

Design Patterns

ASP.NET Web API

```
config.Formatters.Remove(config.Formatters.JsonFormatter);
```

With this change, irrespective of the Accept header value (application/xml or application/json), the Web API service is always going to return XML.

How to return JSON instead of XML from ASP.NET Web API Service when a request is made from the browser.

So here is what we want the service to do

1. When a request is issued from the browser, the web API service should return JSON instead of XML.
2. When a request is issued from a tool like fiddler the Accept header value should be respected. This means if the Accept header is set to application/xml the service should return XML and if it is set to application/json the service should return JSON.

There are 2 ways to achieve this

Approach 1 : Include the following line in Register() method of WebApiConfig.cs file in App_Start folder. This tells ASP.NET Web API to use JsonFormatter when a request is made for text/html which is the default for most browsers. The problem with this approach is that Content-Type header of the response is set to text/html which is misleading.

```
config.Formatters.JsonFormatter.SupportedMediaTypes
    .Add(new MediaTypeHeaderValue("text/html"));
```

Approach 2 : Include the following class in WebApiConfig.cs file in App_Start folder.

```
public class CustomJsonFormatter : JsonMediaTypeFormatter
{
    public CustomJsonFormatter()
    {
        this.SupportedMediaTypes.Add(new MediaTypeHeaderValue("text/html"));
    }

    public override void SetDefaultContentHeaders(Type type, HttpContentHeaders
headers, MediaTypeHeaderValue mediaType)
    {
        base.SetDefaultContentHeaders(type, headers, mediaType);
        headers.ContentType = new MediaTypeHeaderValue("application/json");
    }
}
```

Register the formatter: Place the following line in Register() method of WebApiConfig.cs file in App_Start folder

```
config.Formatters.Add(new CustomJsonFormatter());
```

With these 2 changes, when a request is issued from the browser you will get JSON formatted data and the Content-Type header of the response is also set to application/json. If you are using tools like fiddler and if you set Accept header to application/xml you will still get XML formatted data.

ASP.NET Web API is an extensible framework. This means you can also plugin your own custom formatter. For example, if you want the response to be in CSV format, you can create custom CSVMediaTypeFormatter that inherits from the base abstract class MediaTypeFormatter . The following article describes how to do this.

<http://www.tugberkugurlu.com/archive/creating-custom-csvmediatypeformatter-in-asp-net-web-api-for-comma-separated-values-csv-format>

[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](#)[ASP.NET MVC](#)[Visual Studio Tips and Tricks](#)

Java Video Tutorials

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

WWW.PRAGIMTECH.COM

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

HTTP://FACEBOOK.COM/PRAGIMTECH



3 comments:



Unknown March 22, 2017 at 10:31 AM

Can we remove XMLFormatter for get JSON result.

[Reply](#)

[Replies](#)



himanshu pareek June 17, 2017 at 2:18 AM

yes..as he shown in the video..

[Reply](#)



Suresh Molabanti May 23, 2017 at 2:42 AM

To get the result xml formatter and content-type is also application/xml:--

Plz find the below code

```
public class CustomXmlFormatter : XmlMediaTypeFormatter
{
    public CustomXmlFormatter()
    {
        this.SupportedMediaTypes.Add(new MediaTypeHeaderValue("text/html"));
    }

    public override void SetDefaultContentHeaders(Type type, HttpContentHeaders headers, MediaTypeHeaderValue mediaType)
    {
        base.SetDefaultContentHeaders(type, headers, mediaType);
        headers.ContentType = new MediaTypeHeaderValue("application/xml");
    }
}
```

Register customxmlFormatter in register method in wepapi.config.cs:-
config.Formatters.Add(new CustomXmlFormatter());

[Reply](#)

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

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

Interview Questions

[C#](#)

[SQL Server](#)

[Written Test](#)

CBSE Class 9 Maths

[Number Systems](#)

CPT Tutorial

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

Enter your comment...

Comment as: Unknown (Goo ▾)

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.