In this video we will discuss **Content Negotiation in Web API**.   
  
  
  
One of the standards of the RESTful service is that, the client should have the ability to decide in which format they want the response - XML, JSON etc. A request that is sent to the server includes an Accept header. Using the Accept header the client can specify the format for the response. For example  
  
Accept: application/xml returns XML  
Accept: application/json returns JSON  
  
Depending on the Accept header value in the request, the server sends the response. This is called Content Negotiation.   
  
**So what does the Web API do when we request for data in a specific format**  
The Web API controller generates the data that we want to send to the client. For example, if you have asked for list of employees. The controller generates the list of employees, and hands the data to the Web API pipeline which then looks at the Accept header and depending on the format that the client has requested, Web API will choose the appropriate formatter. For example, if the client has requested for XML data, Web API uses XML formatter. If the client has requested for JSON data, Web API uses JSON formatter. These formatters are called Media type formatters.  
  
ASP.NET Web API is greatly extensible. This means we can also plugin our own formatters, for custom formatting the data.  
  
Multiple values can also be specified for the Accept header. In this case, the server picks the first formatter which is a JSON formatter and formats the data in JSON.  
Accept: application/xml,application/json  
  
You can also specify quality factor. In the example below, xml has higher quality factor than json, so the server uses XML formatter and formats the data in XML.  
application/xml;q=0.8,application/json;q=0.5  
  
If you don't specify the Accept header, by default the Web API returns JSON data.  
  
When the response is being sent to the client in the requested format, notice that the Content-Type header of the response is set to the appropriate value. For example, if the client has requested application/xml, the server send the data in XML format and also sets the Content-Type=application/xml.  
  
The formatters are used by the server for both request and response messages. When the client sends a request to the server, we set the Content-Type header to the appropriate value to let the server know the format of the data that we are sending. For example, if the client is sending JSON data, the Content-Type header is set to application/json. The server knows it is dealing with JSON data, so it uses JSON formatter to convert JSON data to .NET Type. Similarly when a response is being sent from the server to the client, depending on the Accept header value, the appropriate formatter is used to convert .NET type to JSON, XML etc.  
  
It's also very easy to change the serialization settings of these formatters. For example, if you want the JSON data to be properly indented and use camel case instead of pascal case for property names, all you have to do is modify the serialization settings of JSON formatters as shown below. With our example this code goes in WebApiConfig.cs file in App\_Start folder.

config.Formatters.JsonFormatter.SerializerSettings.Formatting =

                            Newtonsoft.Json.Formatting.Indented;

config.Formatters.JsonFormatter.SerializerSettings.ContractResolver =

    new CamelCasePropertyNamesContractResolver();