Most Asked Web API Interview Questions

Following is a list of frequently asked Web API interview questions and their best possible answers.

1) What is a Web API? / What do you understand by Web API?

In Web API, API stands for Application Programming Interface. It is a framework that helps users to develop or build HTTP services that can be used by various clients, including browsers and mobile devices. For example, The ASP.NET Web API helps us to build these services on the .NET framework.

2) Why use Web API? / Why is Web API preferred to use? / What is the reason behind using Web API?

We have many other technologies similar to Web API, but there are several genuine reasons to use Web API.

For example:

- Web API provides us the ability to create non-SOAP-based HTTP services.

- We can easily use it with HTTP verbs for Create, Read, Update and Delete operations.

- It is very lightweight, and its lightweight architecture makes it ideal for small bandwidth devices such as smartphones.

- It is based on HTTP and easy to define, expose and consume in a REST-ful way.

- It provides an ability to select response output in either JSON or XML and support for Open Data (OData) protocol.

3) Is it possible to use RESTful services using WCF? / Why is the use of Web API preferred over RESTful services?

Yes, it is possible to use RESTful services using WCF. We can easily develop RESTful services with WCF, but two main reasons make users choose Web API instead of RESTful services.

- Web API increases the TDD (Test Data-Driven) approach in the development of RESTful services.

- If we want to develop RESTful services in WCF, you surely need many config settings, URI templates, contracts & endpoints.

4) Is it true that ASP.NET Web API has replaced WCF?

No, it is not true at all that ASP.NET Web API has replaced WCF. ASP.NET Web API is another way to build non-SOAP-based services, i.e., plain XML or JSON string.

Web API is lightweight due to using HTTP only so. It cannot match the power and flexibility of WCF. Web API is perfect if you only need to use HTTP as your transport. But if you want to use a different protocol such as TCP or Named Pipes, you would have to use WCF.

5) What is the main difference between Web API and WCF?

WCF stands for Windows Communication Foundation. It is used for service-oriented application development supporting various transport protocols such as HTTP, TCP, MSMQ. The WCF clients must be able to understand XML. On the other hand, Web API is designed specifically for HTTP services, and due to its low-bandwidth, it supports non-SOAP services and most MVC features.

6) What are the main advantages of using Web API?

Following are the main advantages of using Web API:

- Content Negotiation
- Self-Hosting
- OData
- Filters
- Routing
- Model Bindings

7) What are the differences between Web API and MVC?

Following are the main differences between Web API and MVC (Model View Controller)

MVC	Web API
MVC is an application design model used to create web applications with a front-end and controls and back-end processes.	Web API is used to create HTTP services to interact with a variety of clients.
We can use the MVC to create web apps with user interfaces.	We use Web API to develop HTTP services.
In the MVC design pattern, the web application returns a view as well as data.	Web API returns the data only.
MVC can only map requests to action methods.	With Web API, we can map all requests to actions using HTTP verbs.
MVC can return data only in JSON format.	Web API can return data in JSON and XML, among other formats.

8) Which .NET framework supports Web API?

The .NET framework version 4.0 and above versions support web API.

9) What are the main return types supported in Web API?

Following are the main return types of Web API controller action:

- **Void:** It is used to return empty content.
- **HttpResponseMessage:** It is used to convert the response to an HTTP message.
- **IHttpActionResult:** It is used to internally call ExecuteAsync to create an HttpResponseMessage.
- **Other types:** It helps us to write the serialized return value into the response body.

10) Which open-source library is supported by Web API for JSON serialization?

Web API uses JSON.NET library for JSON serialization.

11) What is the biggest drawback of "Other Return Types" in Web API?

The biggest disadvantage of the "Other Return Types" in Web API is that it doesn't help us directly return an error code like a 404 error.

12) What do you understand by Web API Routing?

Web API Routing is a pattern matching just similar we do in MVC. All routes are registered in Route Tables.

For example:

```
Routes.MapHttpRoute(
Name: "ExampleWebAPIRoute",
routeTemplate: "api/{controller}/{id}
defaults: new { id = RouteParameter.Optional}
```

13) Is it possible to use ASP.NET Web API in applications created using other than .NET language?

Yes, we can easily use ASP.NET Web API in the applications created using another language than .NET, but those applications must have access/support to the HTTP protocol.

14) What do you understand by SOAP?

The full form of SOAP is Simple Object Access Protocol. It is an XML message format used in web service interactions to interchange data. It facilitates users to send messages over HTTP or JMS. It is also an XML-based messaging protocol for exchanging information among computers.

In other words, we can say that SOAP is a messaging protocol used for interchanging data in a decentralized and distributed environment.

15) How to use the Web API with ASP.NET Web Form?

We can use the Web API with ASP.NET Web Form by performing the following three simple steps:

- First, create a Web API Controller.
- Now, add a routing table to the Application_Start method of Global. sax
- Now, we can make a jQuery AJAX Call to Web API method and get the data.

16) How can we restrict or limit access to Web API to Specific HTTP Verb?

We can easily restrict or limit access to an ASP.NET Web API method called using a particular HTTP method. In this process, attribute programming plays a crucial role.

17) What do you understand by the RESTful Services?

The full form of REST stands for the **Representational State Transfer**. Roy Fielding coined this term in 2000. RESTful is an architectural style for creating loosely coupled applications over HTTP. To make API to be RESTful, we have to follow the six constraints given below:

- **Client and Server Separation:** Clients and Servers are isolated in the RESTful services.
- **Stateless:** REST architecture is based on the HTTP Protocol, and the clients can cache the server response, but no client context would be stored on the server.
- **Uniform Interface:** It is used to allow a limited set of operations defined using the HTTP Verbs. For example, GET, PUT, POST, Delete, etc.
- **Cacheable:** RESTful architecture allows the response to be cached or not. Caching improves performance and scalability.
- **Code-On-Demand**

- **Layered System**

18) What do you understand by exception filters?

Exception filters are used to execute when exceptions are unhandled and thrown from a controller method. There may be several reasons for the exception. Exception filters implement the "IExceptionFilter" interface.

19) What do you understand by TestApi in Web API?

TestApi is a utility library of Web APIs. Using this library tester, developers can create testing tools and automated tests for a .NET application using data-structure and algorithms.

20) Is it possible to return View from Web API?

No, Web API does not return View, but they return the data. In Web API, APIController is used for returning the data. If you want to return a view from the controller class, you have to make sure you have used or inherited the Controller class.

21) What is Parameter Binding in ASP.NET Web API?

Parameter Binding is a process that specifies that when a Web API calls a method on a controller, it must set the values for the parameters.

By Default, Web API uses the following rules to bind the parameter:

- **FromUri:** If the parameter is of "Simple" type, the Web API tries to get the URI value. Simple Type includes.Net Primitive type like int, double, etc., DateTime, TimeSpan, GUID, string, any type which can convert from the string type.

- **FromBody:** If the parameter is of "Complex" type, the Web API only binds the values from the message body.

22) How can you register exception filter globally?

We can easily register exception filter globally by using the following code:

```
GlobalConfiguration.Configuration.Filters.Add(new
MyTestCustomerStore.NotImplExceptionFilterAttribute());
```

23) What is the difference between REST and RESTful?

Difference between REST and RESTful:

REST	RESTful		
The full form of REST is REpresentational State Transfer.	The term RESTful is written by applying REST architectural concepts called RESTful services.	It is a completely new aspect of writing a web app.	It mainly focuses on system resources and how the resource's state should be transported over HTTP protocol.

24) What is Content Negotiation in Web API?

Content Negotiation is the process of selecting the best representation for a given response from a set of various multiple representations available. Following are the two main headers that are responsible for the Content Negotiation:

- **Content-Type:** The content-type header is used to specify the server about the data; the server will receive from the client.

○ **Accept:** Another way for Content Negotiation is to use Accept-Header. It is used to specify the format of data requested by the client from a server.

25) Give an example to specify Web API Routing?

Following is an example of Web API Routing:

```
Config.Routes.MapHttpRoute(
name: "MyRoute,"//route name
routeTemplate: "api/{controller}/{action}/{id}",//as you can see "API" is at the beginning.
defaults: new { id = RouteParameter.Optional }
);
```

26) What do you understand by a Media-Type Formatter in ASP.NET Web API?

The Media-Type formatter is an abstract class acting as a parent class of JsonMediaTypeFormatter and XmlMediaTypeFormatter. These two classes, JsonMediaTypeFormatter (which is used to handle JSON format) and XmlMediaTypeFormatter (which is used to handle XML format), are derived from the Media-Type formatter.

The Media-Type formatter classes are also responsible for serializing the response data in the client's format.

27) What is the method to handle errors in Web API?

Several classes are available in Web API, such as HttpError, Exception Filters, HttpResponseException, and Registering Exception Filters, etc., to handle errors.

28) What are the several new features that come with ASP.NET Web API 2.0?

Following are the latest new features of ASP.NET Web API framework v2.0:

○ Attribute Routing

○ Open Web Interface NET

○ Cross-Origin Resource Sharing

○ External Authentication

○ HttpActionResult

○ Web API OData

29) How can you restrict access methods to specific HTTP verbs in Web API?

It is very easy to implement access restrictions in Web API with Attributes' help (like HTTP verbs). We can do it by defining HTTP verbs as an attribute to restrict access.

For example,

```
[HttpPost]
public void Method1(Class obj)
{
//Write logic here
```

30) What is Authorize Attribute? What is its use?

Web API provides a built-in authorization filter known as Authorize Attribute. This filter is used to check whether the user is authenticated or not. If not, the user will see 401 Unauthorized HTTP Status Code.

31) Is it possible to pass multiple complex types in Web API?

Yes, we can easily pass multiple complex types in Web API by using the following two methods:

- Using ArrayList

- Using Newtonsoft array

32) What do you understand by basic HTTP Authentication?

Basic HTTP Authentication is a mechanism that is used for user authentication. Here, users are authenticated through the service in which the client passes username and password in the HTTP Authorization request headers. The credentials are formatted as the string "username:password," based on encoded.

33) How can you pass an ArrayList in Web API?

The following example shows the code how we can pass an ArrayList in Web API:

```
ArrayList paramList = new ArrayList();
Category c = new Category { CategoryId = 1, CategoryName ="MobilePhones"};
Product p = new Product { Productcode = 1, Name = "MotoG", Price = 15500, CategoryID = 1 };
paramList.Add(c);
paramList.Add(p);
```

34) How can the Web API route the HTTP request to the Controller ASP.NET MVC?

In ASP.NET Web API, an HTTP request is used to map to the controller. The Web API framework uses a routing table to determine which action is to invoke.

35) How can we handle an error using HttpError in Web API?

In Web API, the HttpError method is used to throw the response body's error info. Along with this, the "CreateErrorResponse" method can also be used, which is an extension method defined in "HttpRequestMessageExtension."

36) How can you register exception filter globally?

We can use the following code to register an exception filter globally:

```
GlobalConfiguration.Configuration.Filters.Add (new MyTestCustomerStore.NotImplExceptionFilterAttribute());
```

37) What do you understand by Exception handling in Web API?

Exception handling is a technique used to handle runtime error in the application code. There are several ways that we can use to handle the error in ASP.NET Web API. Following is a list of some of them:

- HttpResponseException

- HttpError

- Exception Filters etc.

38) What is Web API Versioning, and why is it used?

Web API Versioning is a technique in which Web API is arranged to cope with the business changes, and the API will not impact the client that is using/consuming the existing API. As we know, multiple clients can consume the Web API at a time, so Web API versioning is necessary and required as the business grows, and business requirement changes with the time.

39) How many ways are you to do Web API Versioning?

Following are the several ways to do Web API Versioning:

- URI

- Query String Parameter

- Custom Header Parameter

- Accept Header Parameter

40) What are the advantages of Web API over WCF?

WCF services use the SOAP protocol while Web API never uses the SOAP protocol. That's why Web API services are lightweight since it doesn't use SOAP. It also reduces the data which is transferred to resume service. The biggest advantage of Web API is that it never needs too much configuration. Therefore, the client can interact with the service by using the HTTP verbs.

41) What is the difference between MVC and Web API?

Differences between MVC and Web API:

MVC framework	Web API
The MVC framework is used for developing applications that have a User Interface.	WebAPI is used for developing HTTP services.
In the MVC framework, views are used for building a User Interface.	Other apps can also call the WebAPI methods to fetch that data.

42) Who can use or consume Web API?

Any client which supports HTTP verbs such as GET, PUT, DELETE, POST etc., can use or consume Web API. Web API services don't need any configuration, so it is very easy to consume them by any client. Even portable devices like mobile devices can easily consume them, and it is surely the biggest advantages of this technology.

43) What are the different HTTP methods used in Web API?

Following are the different types of HTTP methods used in Web API. We can use them according to the requirements.

- **GET:** The GET method is used to retrieve the information from the respective server using a given URI.

- **HEAD:** The HEAD method is the same as the GET method, but it is used to transfer the status line and header section.

- **PUT:** The PUT method is used to update values, and it can replace all current resources with the uploaded content.

- **POST:** A POST method is a request, and it is used to send data to the respective server.

- **DELETE:** The DELETE method is used to delete or remove all current resources given by a URI.

- **OPTIONS:** This method is used to describe the communication options for the target resource.

- **CONNECT:** The CONNECT method is used to establish a tunnel to the server identified by a given URI.

- **TRACE:** The TRACE method is used to perform a message loop-back test along the target resource path.

Note: From the list of all the above methods, the GET, PUT, POST, and DELETE are the most popular methods. All methods are case-sensitive, and we must use them in uppercase.

44) What do you understand by the HTTP Status Codes?

The Response Header of each API response consists of an HTTP Status Code. The HTTP Status Codes are the three-digit integers that contain the server response. Here, each number specifies a meaning. The first digit of the Status-Code defines the class of response. According to this first number, the HTTP Status Codes are categorized into five types.

Following is the list of HTTP Status Codes with their meaning:

HTTP Status Codes	Status	Meaning
1xx	Informational	It specifies that the request has been received and the process is continuing.
2xx	Success	It specifies that the action was successfully received, understood, and accepted.
3xx	Redirection	It specifies that we have to take further actions to complete the request.
4xx	Client Error	It specifies that the request contains incorrect syntax or cannot be fulfilled.
5xx	Server Error	It specifies that the server failed to fulfil a valid request.

45) What are some commonly seen HTTP Status Codes?

Following is the list of some commonly seen HTTP Status Codes:

- 200 (Request is Ok)
- 201 (Created)
- 202 (Accepted)
- 204 (No Content)
- 301 (Moved Permanently)
- 400 (Bad Request)
- 401 (Unauthorized)
- 403 (Forbidden)
- 404 (Not Found)
- 500 (Internal Server Error)
- 502 (Bad Gateway)
- 503 (Service Unavailable) etc.

46) What do you understand by Internet Media Types?

The Internet Media Type is a file identification mechanism on the MIME encoding system. It is also known as the "MIME type." It has become the de facto standard for identifying content on the Internet.

For example, suppose we receive an email from a server with an attachment, then the server embeds the media type of the attachment in the message header. So, the browser can launch the appropriate helper application or plug-in.

47) What is the difference between XML and JSON?

Following are the differences between XML and JSON:

XML	JSON
The full form of XML is eXtensible Markup Language. The full form of JSON is JavaScript Object Notation.	
It is a markup language like HTML and is designed to store and transport data.	JSON is often used when data is sent from a server to a web page. It is also used to store and transport data.
XML doesn't do anything, but it can store data in a specific format.	JSON is a self-describing, easy-to-understand, lightweight format for storing and transporting data.

48) In which .NET framework the ASP .NET Web API was introduced?

The first version of ASP.NET Web API was introduced in .NET Framework 4. After that, all the later versions of the .NET Framework support the ASP.NET Web API.

49) What are the main differences between WCF vs. MVC vs. Web API?

WCF vs. MVC vs. ASP.NET Web API:

WCF	MVC	Web API
WCF stands for Windows Communication Foundation. It is a framework that is used to build or develop service-oriented applications.	MVC stands for Model View Controller. It is used to create a web application that returns both data and view (HTML).	Web API is an open-source platform and framework used to create HTTP-based services that only return data and not view.
It is designed to exchange standard SOAP-based messages.	MVC is used to facilitate the easy rendering of HTML.	It supports most of the MVC features, which keep Web API over WCF.
WCF can be consumed by clients who can understand XML.	MVC is used to create a web app in which we can build web pages.	The Web API framework is used to generate HTTP services that cover more clients by generating raw format data, for example, plain XML or JSON string.
WCF supports a variety of transport protocols such as HTTP, TCP, Named Pipes or MSMQ, etc.	JSONit will return JSONResult from the action method.	Web API creates simple HTTP services that render raw data.
WCF ships out with the .NET Framework.	All requests are mapped to the respective action methods.	Web API returns XML or JSON to the client.
WCF service is good for Message Queue, duplex communication, one-way messaging.	We can return a view from MVC. In an ASP.NET MVC application, requests are mapped to Action Methods.	All requests are mapped to actions using HTTP verbs.

50) How can we unit test the Web API?

We can use the Fiddler tool to unit test the Web API. Follow the steps given below and do the setting in the Fiddler tool:

Compose Tab -> Enter Request Headers -> Enter the Request Body and execute.

51) How can you return JSON from Web API Service irrespective of the Accept header value?

We can easily return JSON from Web API Service irrespective of the Accept header value by removing the XmlFormatter from the Register() method of WebApiConfig.cs file, which is present inside the App_Start folder. Use the following piece of code to remove XmlFormatter completely. It will always return data in JSON format irrespective of the Accept header value in the client request.

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

52) What is the difference between Controller and ApiController?

Controller is used to rendering our normal views.

Example:

```
public class TweetsController : Controller {
  // GET: /Tweets/
  [HttpGet]
```

```
  public ActionResult Index() {
    return Json(Twitter.GetTweets(), JsonRequestBehavior.AllowGet);
  }
}
```

ApiController is used when we have only to return data that is serialized and sent to the client.

Example:

```
public class TweetsController : ApiController {
  // GET: /Api/Tweets/
  public List<Tweet> Get() {
    return Twitter.GetTweets();
  }
}
```

Interview Tips	**Job/HR Interview Questions**
Company Interview Questions & Procedure	**JavaScript Interview Questions**
Java Basics Interview Questions	**Java OOPs Interview Questions**
Servlet Interview Questions	**JSP Interview Questions**
Spring Interview Questions	**Hibernate Interview Questions**
PL/SQL Interview Questions	**SQL Interview Questions**
Oracle Interview Questions	**Android Interview Questions**
jQuery Interview Questions	**MySQL Interview Questions**