

### 1) What is Web API?

WebAPI is a framework which helps you to build/develop HTTP services.

### 2) Why is Web API required? Is it possible to use RESTful services using WCF?

Yes, we can still develop RESTful services with WCF. However, there are two main reasons that prompt users to use Web API instead of RESTful services.

- Web API increases TDD (Test Data Driven) approach in the development of RESTful services.
- If we want to develop RESTful services in WCF, you surely need a lot of config settings, URI templates, contracts & endpoints for developing RESTful services using web API.

### 3) Why select Web API?

- It is used to create simple, non-SOAP-based HTTP Services
- It is also an easy method for creation with Web API. With WCF REST Services
- It is based on HTTP and easy to define, expose and consume in a REST-ful way.
- It is lightweight architecture and ideal for devices that have limited bandwidth like smartphones.

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

It's not at all true that ASP.NET Web API has replaced WCF. In fact, it is another way of building non-SOAP based services, i.e., plain XML or JSON string.

### 5) What are the advantages of Web API?

Advantages of Web API are:

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



---

ASP.NET Web API

## 6) What are main return types supported in Web API?

A Web API controller action can return following values:

- Void – It will return empty content
- HttpResponseMessage - It will convert the response to an HTTP message.
- IHttpActionResult - internally calls ExecuteAsync to create an HttpResponseMessage
- Other types - You can write the serialized return value into the response body

## 7) Web API supports which protocol?

Web App supports HTTP protocol.

## 8) Which .NET framework supports Web API?

NET 4.0 and above version supports web API.

## 9) Web API uses which of the following open-source library for JSON serialization?

Web API uses Json.NET library for JSON serialization.

## 10) By default, Web API sends HTTP response with which of the following status code for all uncaught exception?

500 - Internal Server Error

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

The biggest disadvantage of this approach is that you cannot directly return an error code like 404 error.

## 12) How do you construct HttpResponseMessage?

Following is the way to construct to do so,

```
public class TestController : ApiController
{
    public HttpResponseMessage Get()
    {
        HttpResponseMessage response = Request.CreateResponse(HttpStatusCode.OK, "value");
        response.Content = new StringContent("Testing", Encoding.Unicode);
    }
}
```

```
response.Headers.CacheControl = new CacheControlHeaderValue()  
  
{  
  
MaxAge = TimeSpan.FromMinutes(20)  
  
};  
  
return response;  
  
}  
  
}
```

### 13) What is Web API Routing?

Routing is pattern matching like in MVC.

All routes are registered in Route Tables.

For example:

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

### 14) What is SOAP?

SOAP is an XML message format used in web service interactions. It allows to send messages over HTTP or JMS, but other transport protocols can be used. It is also an XML-based messaging protocol for exchanging information among computers.

### 15) What is the benefit of using REST in Web API?

REST is used to make fewer data transfers between client and server which make it an ideal for using it in mobile apps. Web API also supports HTTP protocol. Therefore, it reintroduces the traditional way of the HTTP verbs for communication.

## 16) How can we use Web API with ASP.NET Web Form?

Web API can be used with ASP.NET Web Form

It can be performed in three simple steps:

1. Create a Web API Controller,
2. Add a routing table to Application\_Start method of Global.sax
3. Then you need to make a jQuery AJAX Call to Web API method and get data.

## 17) How to you can limit Access to Web API to Specific HTTP Verb?

Attribute programming plays a important role. It is easy to restrict access to an ASP.NET Web API method to be called using a particular HTTP method.

## 18) Can you use Web API with ASP.NET Web Form?

Yes, It is possible to use Web API with ASP.Net web form. As it is bundled with ASP.NET MVC framework. However, it can be used with ASP.NET Web Form.

## 19) How Can assign alias name for ASP.NET Web API Action?

We can give alias name for Web API action same as in case of ASP.NET MVC by using "ActionName" attribute as follows:

```
[HttpPost]
[ActionName("SaveStudentInfo")]
public void UpdateStudent(Student aStudent)
{
    StudentRepository.AddStudent(aStudent);
}
```

## 20) What is the meaning of TestApi?

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

## 21) Explain exception filters?

It will be executed when exceptions are unhandled and thrown from a controller method. The reason for the exception can be anything. Exception filters will implement "IExceptionFilter" interface.

## 22) How can we register exception filter from the action?

We can register exception filter from action using following code:

```
[NotImplExceptionFilter]

public TestCustomer GetMyTestCustomer(int custid)

{

//write the code

}
```

## 23) How you can return View from ASP.NET Web API method?

No, we can't return a view from ASP.NET Web API Method. Web API creates HTTP services that render raw data. However, it's also possible in ASP.NET MVC application.

## 24) How to register exception filter globally?

It is possible to register exception filter globally using following code-

```
GlobalConfiguration.Configuration.Filters.Add(new

MyTestCustomerStore.NotImplExceptionFilterAttribute());
```

## 25) Explain what is REST and RESTFUL?

REST represents REpresentational State Transfer; it is entirely a new aspect of writing a web app.

RESTFUL: It is term written by applying REST architectural concepts is called RESTful services. It focuses on system resources and how the state of the resource should be transported over HTTP protocol.

## 26) Give me one 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 }

);
```

## 27) How can you handle errors in Web API?

Several classes are available in Web API to handle errors. They are `HttpError`, `Exception Filters`, `HttpResponseException`, and `Registering Exception Filters`.

## 28) What New Features comes with ASP.NET Web API 2.0?

The latest features of ASP.NET Web API framework v2.0 are as follows:

- Attribute Routing
- Cross-Origin Resource Sharing
- External Authentication
- Open Web Interface NET
- `HttpActionResult`
- Web API OData

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

With the help of Attributes (like HTTP verbs), It is possible to implement access restrictions in Web API.

It is possible to define HTTP verbs as an attribute to restrict access. Example:

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

## 30) How can you pass multiple complex types in Web API?

Two methods to pass the complex types in Web API –

Using `ArrayList` and `Newtonsoft array`

## 31) Write a code for passing 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);
```

### 32) Name the tools or API for developing or testing web api?

Testing tools for web services for REST APIs include:

1. Jersey API
2. CFX
3. Axis
4. Restlet

### 33) What is REST?

REST is architectural style. It has defined guidelines for creating services which are scalable. REST used with HTTP protocol using its verbs GET, PUT, POST and DELETE.

### 34) How to unit test Web API?

We can perform a Unit test using Web API tools like Fiddler.

Here, are some setting to be done if you are using

Fiddler –Compose Tab -> Enter Request Headers -> Enter the Request Body and execute

### 35) How can we restrict access to methods with specific HTTP verbs in Web API?

Attribute programming is widely used for this functionality. Web API also allows restricting access of calling methods with the help of specific HTTP verbs. It is also possible to define HTTP verbs as attribute over method.

### 36) What is the usage of DelegatingHandler?

DelegatingHandler is used in the Web API to represent Message Handlers before routing.

### 37) How can we register exception filter from the action?

We can register exception filter from action using following code

```
[NotImplExceptionFilter]

public TestCust GetMyTestCust (int custno)

{

//write the code

}
```

**38) Tell me the code snippet to show how we can return 404 errors from HttpError?**

Code for returning 404 error from HttpError

```
string message = string.Format("TestCustomer id = {0} not found", customerid);
```

```
return Request.CreateErrorResponse(HttpStatusCode.NotFound, message);
```

**39) Explain code snippet to register exception filters from controller?**

```
[NotImplExceptionFilter]
```

```
public class TestCustController : Controller
```

```
{
```

```
//Your code goes here
```

```
}
```

**40) Web API supports which protocol?**

Web App support HTTP protocol

**41) Which of the following .NET framework supports Web API?**

Web API is supported by NET 4.0 version

**42) Web API uses which library for JSON serialization?**

Web API uses Json.NET library for JSON serialization.

**43) By default, Web API sends HTTP response with which of the following status code for all uncaught exception?**

500 - Internal Server Error

**44) Explain method to handle error using HttpError in Web API?**

In WEB API HttpError used to throw the error info in the response body. "CreateErrorResponse" method is can also use along with this, which is an extension method defined in "HttpRequestMessageExtension."



#### 45) How can we register exception filter globally?

We can register exception filter globally using following code:

```
GlobalConfiguration.Configuration.Filters.Add (new MyTestCustomerStore.NotImplExceptionFilter Attribute());
```

#### 46) How to handle errors in Web API?

Several classes are available in Web API to handle errors. They are `HttpError`, `HttpResponseException`, `Exception Filters`, `Registering Exception Filters`.

#### 47) What is the benefit of WebAPI over WCF?

WCF services use the SOAP protocol while HTTP never use SOAP protocol. That's why WebAPI services are lightweight since SOAP is not used. It also reduces the data which is transferred to resume service. Moreover, it never needs too much configuration. Therefore, the client can interact with the service by using the HTTP verbs.

#### 48) State differences between MVC and WebAPI

MVC framework is used for developing applications which have User Interface. For that, views can be used for building a user interface.

WebAPI is used for developing HTTP services. Other apps can also be called the WebAPI methods to fetch that data.

#### 49) Who can consume WebAPI?

WebAPI can be consumed by any client which supports HTTP verbs such as GET, PUT, DELETE, POST. As WebAPI services don't need any configuration, they are very easy to consume by any client. In fact, even portable devices like Mobile devices can easily consume WebAPI which is certainly the biggest advantages of this technology.

#### 50) How can we make sure that Web API returns JSON data only?

To make Web API serialize the returning object to JSON format and returns JSON data only. For that you should add the following code in `WebApiConfig.cs` class in any MVC Web API Project:

```
//JsonFormatter  
  
//MediaTypeHeaderValue  
  
Config.Formatters.JsonFormatter.SupportedMediaTypes.Add(new MediaTypeHeaderValue("application/json"));
```

2

3

//JsonFormatter

//MediaTypeHeaderValue

```
Config.Formatters.JsonFormatter.SupportedMediaTypes.Add(new MediaTypeHeaderValue("application/json"))
```

[Guru99](#) Provides [FREE ONLINE TUTORIAL](#) on Various courses like

Java

MIS

MongoDB

BigData

Cassandra

Web Services

SQLite

JSP

Informatica

Accounting

SAP Training

Python

Excel

ASP Net

HBase

Project

Management

Test Management

Business Analyst

Ethical Hacking

PMP

Live Project

SoapUI

Photoshop

Manual Testing

Mobile Testing

Data Warehouse

R Tutorial

Tableau

DevOps

AWS

Jenkins

Agile Testing

RPA

JUnit

Software

Engineering

Selenium

CCNA

AngularJS

NodeJS

PLSQL

**Stay updated with new  
courses at Guru99  
Join our Newsletter**