

1) What is Web API?

It is a framework which helps us to build/develop HTTP services. So there will be client server communication using HTTP protocol.

2) What are the advantages of using REST in Web API?

REST is always used to make less data transfers between client and server which makes REST an ideal for using it in mobile apps. Web API supports HTTP protocol thereby it reintroduces the old way of HTTP verbs (GET, POST, PUT and DELETE) for communication.

3) What is Representational state transfer or REST?

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

4) Explain Web API Routing?

Routing is the mechanism of pattern matching as we have in MVC. These routes will get registered in Route Tables. Below is the sample route in Web API –

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

5) List out the differences between WCF and Web API?

WCF

- It is a framework build for building or developing service oriented applications.
- WCF can be consumed by clients which can understand XML.
- WCF supports protocols like – HTTP, TCP, Named Pipes etc.
- "WebHttpBinding" to be enabled for WCF Rest.
- For each method there has to be attributes like – "WebGet" and "WebInvoke"
- For GET and POST verbs respectively.

Web API

- It is a framework which helps us to build/develop HTTP services
- Web API is an open source platform.
- It supports most of the MVC features which keep Web API over WCF.
- Unlike WCF Rest we can use full features of HTTP in Web API.
- Web API can be hosted in IIS or in application.

6) Can we unit test Web API and How?

Yes we can unit test Web API.

We can unit test the Web API using Fiddler tool. Below are the settings to be done in Fiddler –
Compose Tab -> Enter Request Headers -> Enter the Request Body and execute

7) What are the advantages of Web API?

Below are the list of support given by Web API –

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

8) Can we return view from Web API?

No. We cannot return view from Web API.

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

Attribute programming is used for this functionality. Web API will support to restrict access of calling methods with specific HTTP verbs. We can define HTTP verbs as attribute over method as shown below

```
[HttpPost]
public void UpdateTestProduct(Product p)
{
    TestProductRepository.AddProduct(p);
}
```

10) Can we use Web API with ASP.NET Web Forms?

Yes. We can use Web API with ASP.NET Webforms.

11) List out the steps to be made for Web API to work in Web Forms?

Below are the steps to be followed –

Creating new controller for Web API.

Adding routing table to "Application_Start" method in asax

Make a AJAX call to Web API actions.

12) Explain how to give alias name for action methods in Web API?

Using attribute "ActionName" we can give alias name for Web API actions. Example

```
[HttpPost]
[ActionName("AliasTestAction")]
public void UpdateTestProduct(Product p)
{
    TestProductRepository.AddProduct(p);
}
```



13) What is the difference between MVC Routing and Web API Routing?

There should be atleast one route defined for MVC and Web API to run MVC and Web API application respectively. In Web API pattern we can find "api/" at the beginning which makes it distinct from MVC routing. In Web API routing "action" parameter is not mandatory but it can be a part of routing.

14) Explain Exception Filters?

Exception filters will be executed whenever controller methods (actions) throws an exception which is unhandled. Exception filters will implement "IExceptionFilter" interface.

15) Explain about the new features added in Web API 2.0 version?

Below are the list of features introduced in Web API 2.0 –

- OWIN
- Attribute Routing
- External Authentication
- Web API OData

16) How can we pass multiple complex types in Web API?

Below are the methods to pass the complex types in Web API –

Using ArrayList
Newtonsoft JArray



17) Write a code snippet for passing arraylist in Web API?

Below is the code snippet for passing arraylist –

```
ArrayList paramList = new ArrayList();
Category c = new Category { CategoryId = 1, CategoryName =
"SmartPhones"};
```

```
Product p = new Product { ProductId = 1, Name = "Iphone", Price = 500,
CategoryId = 1 };
paramList.Add(c);
paramList.Add(p);
```



18) Give an example of Web API Routing?

Below is the sample code snippet to show 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 }
);
```


18) Give an example of Web API Routing?

Below is the sample code snippet to show 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 }  
);
```

19) Give an example of MVC Routing?

Below is the sample code snippet to show MVC Routing –

```
routes.MapRoute(  
    name: "MyRoute", //route name  
    url: "{controller}/{action}/{id}", //route pattern  
    defaults: new  
    {  
        controller = "a4academicsController",  
        action = "a4academicsAction",  
        id = UrlParameter.Optional  
    }  
);
```

20) How we can handle errors in Web API?

Below are the list of classes which can be used for error handling –

- `HttpResponseException`
- `Exception Filters`
- `Registering Exception Filters`
- `HttpError`



21) Explain how we can handle error from "HttpResponseException"?

This returns the HTTP status code what you specify in the constructor. Example :

```
public TestClass MyTestAction(int id)  
{  
    TestClass c = repository.Get(id);  
    if (c == null)  
    {  
        throw new HttpResponseException(HttpStatusCode.NotFound);  
    }  
    return c;  
}
```

22) How to register Web API exception filters?

Below are the options to register Web API exception filters –

- From Action
- From Controller
- Global registration



23) Write a code snippet to register exception filters from action?

Below is the code snippet for registering exception filters from action –

```
[NotImplExceptionFilter]
public TestCustomer GetMyTestCustomer(int custid)
{
    //Your code goes here
}
```



24) Write a code snippet to register exception filters from controller?

Below is the code snippet for registering exception filters from controller –

```
[NotImplExceptionFilter]
public class TestCustomerController : Controller
{
    //Your code goes here
}
```

25) Write a code snippet to register exception filters globally?

Below is the code snippet for registering exception filters globally –

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



26) How to handle error using HttpStatusCodeResult?

HttpError will be used to throw the error info in response body. "CreateErrorResponse" method is used along with this, which is an extension method defined in "HttpRequestMessageExtensions".

27) Write a code snippet to show how we can return 404 error from HttpStatusCodeResult?

Below is the code snippet for returning 404 error from HttpStatusCodeResult –

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

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

28) How to enable tracing in Web API?

To enable tracing place below code in –"Register" method of WebAPIConfig.cs file.

```
config.EnableSystemDiagnosticsTracing();
```



29) Explain Authentication in Web API?

Web API authentication will happen in host. In case of IIS it uses Http Modules for authentication or we can write custom Http Modules. When host is used for authentication it used to create principal, which represent security context of the application.

30) Explain ASP.NET Identity?

This is the new membership system for ASP.NET. This allows to add features of login in our application.

Below are the list of features supported by ASP.NET Identity in Web API –

- One ASP.NET Identity System
- Persistence Control

31) What are Authentication Filters in Web API?

Authentication Filter will let you set the authentication scheme for actions or controllers. So this way application can support various authentication mechanisms.

32) How to set the Authentication filters in Web API?

Authentication filters can be applied at the controller or action level. Decorate attribute – “IdentityBasicAuthentication” over controller where we have to set the authentication filter.

33) Explain method – “AuthenticateAsync” in Web API?

“AuthenticateAsync” method will create “IPrincipal” and will set on request. Below is the sample code

snippet for “AuthenticateAsync” –

```
Task AuthenticateAsync(  
    HttpContext mytestcontext,  
    CancellationToken mytestcancellationToken  
)
```



34) How to set the Error Result in Web API?

Below is the sample code to show how to set error result in Web API –

```
HttpResponseMessage myresponse = new HttpResponseMessage(HttpStatusCode.Unauthorized);  
myresponse.RequestMessage = Request;  
myresponse.ReasonPhrase = ReasonPhrase;
```

35) Explain method – “ChallengeAsync” in Web API?

“ChallengeAsync” method is used to add authentication challenges to response. Below is the method signature –

```
Task ChallengeAsync(  
    HttpContext mytestcontext,  
    CancellationToken mytestcancellationToken  
)
```



36) What are media types?

It is also called MIME, which is used to identify the data. In HTML, media types are used to describe message format in the body.

37) List out few media types of HTTP?

Below are the list of media types –

- Image/Png
- Text/HTML
- Application/Json



38) Explain Media Formatters in Web API?

Media Formatters in Web API can be used to read the CLR object from our HTTP body and Media formatters are also used for writing CLR objects of message body of HTTP.

39) How to serialize read-only properties?

Read-Only properties can be serialized in Web API by setting the value "true" to the property – "SerializeReadOnlyTypes" of class – "DataContractSerializerSettings".



40) How to get Microsoft JSON date format ?

Use "DateFormatHandling" property in serializer settings as below –

```
var myjson = GlobalConfiguration.Configuration.Formatters.JsonFormatter;  
  
myjson.SerializerSettings.DateFormatHandling =  
Newtonsoft.Json.DateFormatHandling.MicrosoftDateFormat;
```

41) How to indent the JSON in web API?

Below is the code snippet to make JSON indenting –

```
var mytestjson = GlobalConfiguration.Configuration.Formatters.JsonFormatter;  
  
mytestjson.SerializerSettings.Formatting = Newtonsoft.Json.Formatting.Indented;
```



42) How to JSON serialize anonymous and weakly typed objects?

Using "Newtonsoft.Json.Linq.JObject" we can serialize and deserialize weakly typed objects.

43) What is the use of "IgnoreDataMember" in Web API?

By default if the properties are public then those can be serialized and deserialized, if we do not want to serialize the property then decorate the property with this attribute.



44) What is “Under-Posting” and “Over-Posting” in Web API?

“Under-Posting” – When client leaves out some of the properties while binding then it’s called under – posting.

“Over-Posting” – If the client sends more data than expected in binding then it’s called over-posting.

45) How to handle validation errors in Web API?

Web API will not return error to client automatically on validation failure. So its controller’s duty to check the model state and response to that. We can create a custom action filter for handling the same.



46) Give an example of creating custom action filter in Web API?

Below is the sample code for creating custom action filter –

```
public class MyCustomModelAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(HttpActionContext actionContext)
    {
        if (actionContext.ModelState.IsValid == false)
        {
            //Code goes here
        }
    }
}
```



47) How to apply custom action filter in WebAPI.config?

Add a new action filter in “Register” method as shown –

```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        config.Filters.Add(new MyCustomModelAttribute());
        // ...
    }
}
```



48) What is BSON in Web API?



It’s a binary serialization format. “BSON” stands for “Binary JSON”. BSON serializes objects to key-value pair as in JSON. Its light weight and its fast in encode/decode.

49) How parameter binding works in Web API?

Below are the rules followed by WebAPI before binding parameters –

If it is simple parameters like – bool,int, double etc. then value will be obtained from the URL. Value read from message body in case of complex types.

50) Why to use "FromBody" in Web API?

This attribute is used to force Web API to read the simple type from message body. "FromBody" attribute is along with parameter. Example :

```
public HttpResponseMessage Post([FromBody] int customerid, [FromBody] string customername) { ... }
```

