**What are the type of routing available in WebApi?**

**Convention-based Routing**- In the convention-based routing, Web API uses route templates to determine which controller and action method to execute. At least one route template must be added into route table in order to handle various HTTP requests.

**Attribute Routing** - Attribute routing is supported in Web API 2. As the name implies, attribute routing uses [Route()] attribute to define routes. The Route attribute can be applied on any controller or action method.

In order to use attribute routing with Web API, it must be enabled in WebApiConfig by calling config.MapHttpAttributeRoutes() method.

**What is the different between 1.0 and 2.0 webapi?**

Version 2.0, the Web API framework has been enhanced to support the following features:

IHttpActionResult return type

Support Attribute Routing

Support for Cross-Origin requests using CORS

Securing ASP.NET Web API using OAuth 2.0

Support for $expand, $select in OData Service

OWIN self-host

HttpRequestContext

**How Api's call in angular?**

Angular provides the HttpClient module which allows developers to send HTTP requests and make API calls to remote HTTP servers.

Angular provides a simplified client HTTP API for Angular applications, the HttpClient service class in @angular/common/http.

The HTTP client service offers the following major features.

The ability to request typed response objects.

Streamlined error handling.

Testability features.

Request and response interception.

**What is Web API?**

Web API (Application Programming Interface) is a framework that helps to

Create resource-oriented services using the full features of HTTP

Exposing services to a variety of clients easily like browsers or mobile devices, etc.

**Difference between .Net Core and .Net Framework**

**Open Source**

**Cross Platform**

#### Why use Web API?

#### **To create non-SOAP-based HTTP services**

#### Use of familiar HTTP verbs for Create, Read, Update and Delete operations

#### Its lightweight architecture making it ideal for small bandwidth devices such as smartphones

#### ability to select response output in either JSON or XML along with support for Open Data (OData) protocol

**What are the advantages of Web API?**

Advantages of Web API are:

* It works the HTTP way using standard HTTP verbs like GET, POST, PUT, DELETE, etc. for all CRUD operations
* Complete support for routing
* Response generated in JSON or XML format using MediaTypeFormatter
* It has the ability to be hosted in IIS as well as self-host outside of IIS
* Supports Model binding and Validation
* Support for OData
* Filters
* Content Negotiation

**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

#### What is the difference between Web API and WCF?

#### WCF - is for service-oriented application development supporting a variety of transport protocols (e.g., HTTP, TCP, MSMQ). WCF clients must be able to understand XML.

#### What are some of the differences between Web API and MVC?

#### MVC returns a view as well as data. However, Web API returns only the data.

#### MVC maps requests to action methods.Web API, requests map to actions using HTTP verbs.

#### What is REST?

REST, or Representational State Transfer, is an architectural style that uses a set of standards to make it easy for systems to communicate over the internet using HTTP.

RESTful, are stateless in that clients and servers know nothing about the other's state.

They are also modular, meaning clients and servers, user interfaces and data storage, can be maintained independently.

A Web API may or may not be RESTful".

**\*Client-server separation**: Clients and servers have no dependence upon one another. Clients need only know the URI (Uniform Resource Identifier) of the server resource they need to access.\*

**\*Stateless**: The client's request to the server should be self-contained. The server does not need to maintain any information about the client in order to complete the request.\*

**\*Uniform interface**: Each resource made available to clients must have only one logical URI and provide a way to get additional data. There should be a common, consistent approach to accessing and modifying resources.\*

**\*Cacheable**: For the sake of performance and scalability, the client should be able to store certain server responses (e.g., lists of items that do not change frequently).\*

**\*Layered system:** A layered system divides APIs, data storage, and request authentication between three separate servers. The client is usually unaware of whether it has connected directly to the end server, or to one of the intermediary servers.\*

**\*Code-on-demand**: This constraint is optional. If necessary, instead of responding with static representations of resources in XML or JSON, the server can send executable code".\*

**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

**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. Restlets
5. Postman
6. Fiddler

**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.