**Q. What is Web Service in .NET?**

**ANS**:

A web service is a web-based functionality accessed using the protocols of the web to be used by the web applications and uses standard XML Messaging for communication. XML is used to encode all communications to a Web service. For example, a client invokes a Web service by sending an XML message, and then waits for a corresponding XML response. Because all communication is in XML, Web services are not tied to any one operating system or programming.

### Q. What are the advantages of Web Services?

**ANS**:

* **Interoperability:** Web services are accessible over network and runs on HTTP/SOAP protocol and uses XML/JSON to transport data, hence it can be developed in any programming language. Web service can be written in java programming and client can be PHP and vice versa.
* **Reusability**: One web service can be used by many client applications at the same time.
* **Loose Coupling**: Web services client code is totally independent with server code, so we have achieved loose coupling in our application.
* Easy to deploy and integrate, just like web applications.
* Multiple service versions can be running at same time.

### Q. What are different types of Web Services?

### ANS:

There are two types of web services:

1. **SOAP Web Services**: Runs on SOAP protocol and uses XML technology for sending data.
2. **Restful Web Services**: It’s an architectural style and runs on HTTP/HTTPS protocol almost all the time. REST is a stateless client-server architecture where web services are resources and can be identified by their URIs. Client applications can use HTTP GET/POST methods to invoke Restful web services.

### Q. What are different types of Web Services?

### ANS:

The basic web services platform is XML + HTTP. All the standard web services work using the following components −

* SOAP (Simple Object Access Protocol)
* UDDI (Universal Description, Discovery and Integration)
* WSDL (Web Services Description Language)

**Q. What is the extension for Web Service?**

**ANS**: .asmx

**Q. What is SOAP?**

**ANS**: SOAP stands for Simple Access Object Protocol.

SOAP is an XML-based protocol for exchanging information between computers.

**Q. What is the namespace for Web Service?**

**ANS**: System.Web.Services

**Q. What is WSDL?**

**ANS**:

WSDL stands for Web Service Description Language. WSDL is an XML based document that provides technical details about the web service. Some of the useful information in WSDL document is: method name, port types, service end point, binding, method parameters etc. WSDL is an XML-based language for describing web services and how to access them.

**Q. What are the Components of WSDL?**

**ANS**:

Some of the different tags in WSDL xml are:

* xsd:import namespace and schemaLocation: provides WSDL URL and unique namespace for web service.
* message: for method arguments
* part: for method argument name and type
* portType: service name, there can be multiple services in a wsdl document.
* operation: contains method name
* soap:address for endpoint URL.

**Q. What is UDDI?**

**ANS**:

UDDI is an acronym for Universal Description, Discovery, and Integration. UDDI is used for locating the web service. UDDI is acronym for Universal Description, Discovery and Integration. UDDI is a directory of web services where client applications can look up for web services. Web Services can register to the UDDI server and make them available to client applications.

**Q. What is REST web service?**

**ANS**:

REST is the acronym for Representational State Transfer. REST is an architectural style for developing applications that can be accessed over the network. REST architectural style was brought in light by Roy Fielding in his doctoral thesis in 2000.REST is a stateless client-server architecture where web services are resources and can be identified by their URIs. Client applications can use HTTP GET/POST methods to invoke Restful web services. REST doesn’t specify any specific protocol to use, but in almost all cases it’s used over HTTP/HTTPS. When compared to SOAP web services, these are lightweight and don’t follow any standard. We can use XML, JSON, text or any other type of data for request and response.

Restful web services supported HTTP methods are – GET POST, PUT, DELETE and HEAD.

### Q. Compare SOAP and REST web services?

**ANS**:

|  |  |  |
| --- | --- | --- |
| SR No | SOAP | REST |
| 1 | SOAP is a standard protocol for creating web services. | REST is an architectural style to create web services. |
| 2 | SOAP is acronym for Simple Object Access Protocol. | REST is acronym for Representational State Transfer. |
| 3 | SOAP uses WSDL to expose supported methods and technical details. | REST exposes methods through URIs, there are no technical details. |
| 4 | SOAP web services and client programs are bind with WSDL contract | REST doesn’t have any contract defined between server and client |
| 5 | SOAP web services and client are tightly coupled with contract. | REST web services are loosely coupled. |
| 6 | SOAP learning curve is hard, requires us to learn about WSDL generation, client stubs creation etc. | REST learning curve is simple; POJO classes can be generated easily and works on simple HTTP methods. |
| 7 | SOAP supports XML data format only | REST supports any data type such as XML, JSON, image etc. |
| 8 | SOAP web services are hard to maintain, any change in WSDL contract requires us to create client stubs again and then make changes to client code. | REST web services are easy to maintain when compared to SOAP, a new method can be added without any change at client side for existing resources |
| 9 | SOAP web services can be tested through programs or software such as Soap UI. | REST can be easily tested through CURL command, Browsers and extensions such as Chrome Postman. |

### Q. What is use of javax.xml.ws.Endpoint class?

### ANS:

Endpoint class provides useful methods to create endpoint and publish existing implementation as web service. This comes handy in testing web services before making further changes to deploy it on actual server.

### Q. What is ASP.NET WEB API?

### ANS:

ASP.NET Web API is a framework that simplifies building HTTP services for broader range of clients (including browsers as well as mobile devices) on top of .NET Framework.

### Q. What are the advantages of ASP.NET WEB API?

### ANS:

Using ASP.NET Web API has a number of advantages, but core of the advantages 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

#### Q. What New Features are introduced in ASP.NET Web API 2.0?

#### ANS:

More new features introduced in ASP.NET Web API framework v2.0 are as follows:

* Attribute Routing
* External Authentication
* CORS (Cross-Origin Resource Sharing)
* OWIN (Open Web Interface for .NET) Self Hosting
* IHttpActionResult
* Web API OData

#### Q How to Restrict Access to Web API Method to Specific HTTP Verb?

#### ANS:

Attribute programming plays its role here. We can easily restrict access to an ASP.NET Web API method to be called using a specific HTTP method. For example, we may require in a scenario to restrict access to a Web API method through HTTP POST only as follows:

[HttpPost]

public void UpdateStudent(Student aStudent)

{

StudentRepository.AddStudent(aStudent);

}

#### Q Can we use Web API with ASP.NET Web Form?

#### ANS:

Yes, ASP.NET Web API is bundled with ASP.NET MVC framework but still it can be used with ASP.NET Web Form.

It can be done in three simple steps as follows:

1. Create a Web API Controller
2. Add a routing table to Application\_Start method of Global.asax
3. Make a jQuery AJAX Call to Web API method and get data

**Q What are the Http methods used to implement a RESTful API?**

#### ANS:

**POST** Creates a new resource.

**GET** Retrieves a resource.

**PUT** Updates an existing resource.

**DELETE** Deletes a resource.

**Q .Can we do unit test Web API??**

#### ANS:

#### Web API can be unit test by using Fiddler tool. Following is the settings to be updated in Fiddler: Compose Tab -> Enter Request Headers -> Enter the Request Body and execute