Module 6

1. Understanding HTTP Verbs:

- HTTP verbs tell the server what to do with the data identified by the URL.
- The HTTP method is supplied in the request line and specifies the operation that the client has requested.
- If you're going to follow the REST architecture and the HTTP protocol, you must choose from the verbs available in that protocol, the primary or most-commonly-used HTTP verbs are POST, GET, PUT, and DELETE.

GET

This method is used to retrieve a representation of a resource. A GET request is considered safe because as HTTP specifies, these requests are used only to read data and not change it.

So in short there are no side effects and GET requests can be re-issued without worrying about the consequences.

The disadvantage of GET requests is that they can only supply data in the form of parameters encoded in the URI or as cookies in the cookie request header.

POST

POST is used when the processing you wish to do on the server should be repeated or when creating a new resource or for a POST to the parent when the service associates the new resource with the parent or for assigning an ID, etcetera.
It is used to create new resources. For some resources, it may be used to alter the internal state.

PUT

PUT is used to create a resource, not in a general case but when the resource ID is chosen by the client instead of by the server, then only PUT is used.
Simply PUT updates data in the repository, replacing any existing data with the supplied data.

DELETE

A DELETE request is as simple as its name implies; it just deletes, it is used to delete a resource identified by a URI and on successful deletion, it returns HTTP status 200 (OK) along with a response body.

The important and unique thing about a DELETE request is that the client cannot be guaranteed that the operation has been carried out, even if the status code returned from the origin server indicates that the action has been completed successfully.

2. Understanding JSON Structure:

- JSON is an open-standards document format for human-readable and machine-understandable serialization and deserialization of data.
- Simply, it is used for data-interchange. The benefit of JSON is that it has a very compact size as compared to XML documents of the same purpose and data.
- JSON stores the data in the form of key/value pairs. Another benefit of JSON is it is language-independent.
- You can work with JSON data in almost any programming language that can handle string objects.
- Almost every programming framework that I can think of most of the programming frameworks support JSON-based data-interchange. For example, ASP.NET Web API supports JSON format data transfer to-and-from the server.

- The basic structure of JSON document is very much simple. Every document in JSON must have either an object at its root, or an array.

- Object and array can be empty there is no need for it to contain anything in it, but there must be an object or an array.

JSON data types

- JSON data types are the valid values that can be used in JSON files. In this section I want to clarify the values that they can hold, how you can use them in your own project data-interchange formats.

JSON Object

- At the root of the JSON document, there needs to be either a JSON object or a JSON array. In JavaScript and in many other programming languages an object is denoted by curly braces; “{ }”. JSON uses the same notation for denoting the objects in the JSON file.

JSON Arrays

Like JavaScript, JSON also supports storing the objects in a sequence, series or list; what-ever you would like to put it as. Arrays are not complex structures in JSON document. They are,

1. Simple.
2. Start with a square bracket, and end on the same.
3. Can contain values in a sequence.
4. Each value is separated by a comma (not a colon).
5. Value can be of any type.

JSON Strings

- String typed values are the values which contain the character-type values. In many programming environments, strings are called, arrays of characters.
- In JavaScript, you can use either single quotes or double quotes to wrap the string values.
- But, in JSON specification you should always consider using double quotation.
- Using single quotation may work and should work in JavaScript environment but in the cases when you have to share the data over the network to a type-safe programming environment such as where C++ or C# may be used as the programming language.