BERK BALCI Homework 3

1- Soap vs Restful ?

Security: While it can be easier and faster to provide security on SOAP, this can be complicated for REST. When it comes to security for REST, it is a bit weak compared to SOAP.

Data Structure: While we can work with REST with JSON, XML and even TEXT, we should use XML with SOAP. REST can be more useful in this way. With JSON, you can perform operations with smaller data. If data sizes are important for your application, using REST will be suitable for you.

Application Speed: If you want your application to run faster, it will be beneficial to use REST.

Support: Developer tools for SOAP are better. We can find more resources to help us use it. REST has less documentation than SOAP.

Also REST architecture is very flexible and lightweight compared to SOAP, the amount of data moved is less, and it is easier to integrate.

2- Difference between acceptance test and functional test ?

Functional tests check the results of a given set of data by comparing them to a set of results. Functional tests are not concerned with intermediate results or side effects. They are only result oriented. Once the X operation is done, they don't care what the result is.

The purpose of acceptance tests; It is the control of whether the necessary operations can be performed so that the user group to which the software will be presented can accept the software as running. Rather than detecting where there are errors in the system, "The system cannot perform the expected operations." used to mean.

3- What is Mocking ?

The general use of mocks can be defined as verifying whether the operations expected to be performed by the system under test are performed on the dependency it replaces. The difference of mock objects from spy objects is that while both follow the operations on them, mocks integrate this process into the validation part of the tests.

4- What is a reasonable code coverage % for unit tests (and why) ?

With that being said it is generally accepted that % 80 covarege is a good goal to aim for. Trying to reach a higher coverage might turn out to be costly, while not necessary producing enough benefit.

5- HTTP/POST vs HTTP/PUT ?

POST and PUT are both HTTP methods used to send data to the server. POST is only used to send data to a specific resource and what to do with the data is up to the server. In PUT, the same resource is accessed with the same address and if there is content, it is replaced with the incoming data, if there is no content, new content is created. With PUT, it is mostly preferred to send file-based content to the server.

6- What are the Safe and Unsafe methods of HTTP ?

Safe HTTP methods : GET, HEAD or OPTIONS

Unsafe HTTP methods : Put and Delete

7- How does HTTP Basic Authentication work ?

HTTP Basic Authentication is a method for the client to provide a username and password when making a request. In this authentication method, which can be set up for a specific file or the entire workspace on the server side, the client must enter the correct login information.

8- Define RestTemplate in Spring ?

RestTemplate is the default class in Spring library to handle synchronous HTTP requests on client side.

9- What is idempotent and which HTTP methods are idempotent ?

Idempotent is an HTTP method .If an identical request can be made once or several times in a row with the same effect while leaving the server in the same state. Implemented correctly, the Get , Head , Put and Delete methods are idempotent, but not the POST method.All safe methods are also idempotent.

10- What is DNS Spoofing ? How to prevent ?

Domain Name Server (DNS) spoofing (a.k.a. DNS cache poisoning) is an attack in which altered DNS records are used to redirect online traffic to a fraudulent website that resembles its intended destination.

Any user that accesses the internet from public Wi-Fi is vulnerable to DNS spoofing. To protect from DNS spoofing, internet providers can use DNSSEC (DNS security).

11- What is content negotiation ?

Content negotiation refers to mechanisms defined as part of HTTP that make it possible to serve different versions of a document in the same URI so that user agents can determine which version best fits their specifications.

12- What is statelessness in RESTful Web Services ?

Statelessness means that every HTTP request happens in complete isolation. When the client makes an HTTP request, it includes all information necessary for the server to fulfill the request. The server never relies on information from previous requests from the client.

13- What is CSRF attack? How to prevent ?

A key design principle that protects you from CSRF attacks is using GET requests for only view or read-only actions. These types of requests should not transform data and must only display recorded data. This limits the number of requests that are vulnerable to CSRF attacks.

The most robust way to defend against CSRF attacks is to include a CSRF token within relevant requests. The token should be: Unpredictable with high entropy, as for session tokens in general. Tied to the user's session.

14- What are the core components of the HTTP request and HTTP response ?

Core components of HTTP requests are:

HTTP Version – Indicates version

Request Body – Represents message content

Request Header – Contains metadata, such as cache settings and client type, for the HTTP request message

URI – Identifies the resource on the server

Verb – Indicates HTTP methods such as GET, POST, and PUT

Core components of HTTP response are :

HTTP Version – Indicates the present version of HTTP

Response Body – Represents the response message content

Response Header – Consists of metadata, like content length and server length, for the HTTP response message

Status/Response Code – Indicates the server status for the requested resource