W03 HomeWork – Talha Arıç, talharic@gmail.com

1 – SOAP vs Restful ?

SOAP is a standard used to create web services independent of the programming language.

Envelope – It is the top (root) element of the SOAP structure and is mandatory.

Header – Used for authorization and SOAP settings. We can compare it to the head tag in HTML.

Body – It is the most important part in the SOAP structure. This section contains information about methods or information about the result of the method.

Fault – If any error occurs, information about the error (error code, error description) is included.

REST is an architecture that works over the HTTP protocol that provides communication between client and server. REST is a transfer method used in software built on service-oriented architecture. It provides communication between the client and the server by carrying XML and JSON data.

2 - Difference between acceptance test and functional test ?

Functional Testing: This is a validation activity; Have we developed a product that works correctly? Does the software meet the business requirements?

For such tests, we have test cases that cover all possible scenarios we can think of, even if that scenario is unlikely to exist in the "real world". We aim for maximum code coverage when doing this type of testing.

Acceptance Test: This is a validation activity; did we do the right thing? Is this what the customer really needs?

This is usually done in collaboration with the customer or by an internal customer representative (product owner). For such tests, we use test cases that cover typical scenarios where we expect the software to be used.

3 - What is Mocking ?

Mocking is one of the methods used by TDD, which is a popular software methodology, and in particular unit tests, to isolate the system they are testing. These methods can be broadly defined as test stunts. Test stunts replace other units on which the tested system is dependent. There are basically two reasons why this isolation is needed in unit tests:

Unit tests are often written to verify assumptions about the system they are testing.

Test stunts vary in their behavior and usage patterns. It can be said that the most used ones are dummy, fake, stub, spy and mock. The general factors that cause this diversity are related to whether these stuntmen do the job expected and how they behave while doing it.

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

Although 100% code coverage may appear like a best possible effort, even 100% code coverage is estimated to only expose about half the faults in a system. Low code coverage indicates inadequate testing, but high code coverage guarantees nothing.

In a large system, achieving 100% code coverage is generally not cost effective. Some reasons are listed below.

5 – HTTP/POST vs HTTP/PUT ?

PUT method is used to update resource available on the server. Typically, it replaces whatever exists at the target URL with something else. You can use it to make a new resource or overwrite an existing one. PUT requests that the enclosed entity must be stored under the supplied requested URI (Uniform Resource Identifier).

POST is a method that is supported by HTTP and depicts that a web server accepts the data included in the body of the message, which is requested. POST is often used by World Wide Web to send user generated data to the web server or when you upload file.

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

An HTTP method is safe when used to perform a read-only operation, such as retrieving information. In contrast, an unsafe HTTP method is used to change the state of an application, for instance to update a user's profile on a web application.

Common safe HTTP methods are GET, HEAD, or OPTIONS.

Common unsafe HTTP methods are POST, PUT and DELETE.

Allowing both safe and unsafe HTTP methods to perform a specific operation on a web application could impact its security, for example CSRF protections are most of the time only protecting operations performed by unsafe HTTP methods.

7 - How does HTTP Basic Authentication work ?

1-A client requests access to a protected resource.

2-The web server returns a dialog box that requests the user name and password.

3-The client submits the user name and password to the server.

4-The server authenticates the user in the specified realm and, if successful, returns the requested resource.

8 - Define RestTemplate in Spring ?

RestTemplate is a synchronous client to perform HTTP requests. It uses a simple, template method API over underlying HTTP client libraries such as the JDK HttpURLConnection, Apache HttpComponents, and others.

9 – What is idempotant and which HTTP methods are idempotant ?

An HTTP method is idempotent 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. In other words, an idempotent method should not have any side-effects (except for keeping statistics). 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 ?

DNS spoofing is the process of poisoning entries on a DNS server to redirect a targeted user to a malicious website under attacker control.

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). When a domain owner sets up DNS entries, DNSSEC adds a cryptographic signature to the entries required by resolvers before they accept DNS lookups as authentic.

11 – What is content negotiation ?

In HTTP, content negotiation is the mechanism that is used for serving different representations of a resource to the same URI to help the user agent specify which representation is best suited for the user (for example, which document language, which image format, or which content encoding).

12 – What is statelessness in RESTful Web Services ?

As per the REST architecture, the server does not store any state about the client session on the server-side. This restriction is called Statelessness. 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. If any such information is important then the client will send that as part of the current request.

13 - What is CSRF attack? How to prevent ?

Cross-site request forgery is a simple yet invasive malicious exploit of a website. It involves a cyberattacker adding a button or link to a suspicious website that makes a request to another site you're authenticated on. For example, a user is logged into their online banking platform which has poor security, and by clicking a "download" button on an untrusted site, it maliciously initiates a money transfer request on their behalf through their current online banking session. Compromised sites can reveal information or perform actions as an authorized user without your explicit permission.

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.

Strictly validated in every case before the relevant action is executed.

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

Have you ever thought about how the front-end of an application communicates with the backend to get data or perform certain operations? It is done through API Requests. API stands for Application Programming Interface. The communication between our client and the API is achieved using HTTP Request which is followed by a Response to the client. Both the Requests and Response follows a certain syntax and structure to ease the communication process.

Whenever our client application wants to communicate to the server, it sends out a message to the server using HTTP Protocols, which is also termed as the HTTP Request. Based on that message, the

server performs certain operations as demanded by the message and then replies to the client through a message, also knows as HTTP Response.