HW#3 Gulnisa Aslan gulnisaslan@gmail.com

1 – SOAP vs Restful ?

SOAP is a **protocol**.	REST is an **architectural style**.
SOAP stands for **Simple Object Access Protocol**.	REST stands for **REpresentational State Transfer**.
SOAP **can't use REST** because it is a protocol.	REST **can use SOAP** web services because it is a concept and can use any protocol like HTTP, SOAP.
SOAP **uses services interfaces to expose the business logic**.	REST **uses URI to expose business logic**.
JAX-WS is the java API for SOAP web services.	**JAX-RS** is the java API for RESTful web services.
SOAP **defines standards** to be strictly followed.	REST does not define too much standards like SOAP.
SOAP **requires more bandwidth** and resource than REST.	REST **requires less bandwidth** and resource than SOAP.
SOAP **defines its own security**.	RESTful web services **inherits security measures** from the underlying transport.
SOAP permits XML data format only.	REST **permits different** data format such as Plain text, HTML, XML, JSON etc.
SOAP is **less preferred** than REST.	REST **more preferred** than SOAP.

2 - Difference between acceptance test and functional test ?

Functional testing - test the product, verifying that it has the qualities you've designed or build (functions, speed, errors, consistency, etc.)

Acceptance testing - test the product in its context, this requires (simulation of) human interaction, test it has the desired effect on the original problem(s).

3 - What is Mocking ?

Mocking uses unit tests.A lot of programming languages use mocking. Mocking's function, we don't use real classes. It creates fake class instance for us. When unit test write, We use to create fake istances

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

In particular Quality is very important from unit testing written so functionality coverage is very important.

5 – HTTP/POST vs HTTP/PUT ?

PUT	POST
PUT request is made to a particular resource. If the Request-URI refers to an already existing resource, an update operation will happen, otherwise create operation should happen if Request-URI is a valid resource URI (assuming client is allowed to determine resource identifier). **Example –** PUT /article/{article-id}	POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. It essentially means that POST request-URI should be of a collection URI. **Example –** POST /articles
PUT method is idempotent. So if you send retry a request multiple times, that should be equivalent to single request modification.	POST is NOT idempotent. So if you retry the request N times, you will end up having N resources with N different URIs created on server.
Use PUT when you want to modify a single resource which is already a part of resources collection. PUT overwrites the resource in its entirety. Use PATCH if request updates part of the resource.	Use POST when you want to add a child resource under resources collection.
Generally, in practice, always use PUT for UPDATE operations.	Always use POST for CREATE operations.

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

Http Safe Methods: Get , Head, Options
Http Unsafe Methods:Put, Delete, Post

7 - How does HTTP Basic Authentication work ?

HTTP Basic Authentication requires that the server request a user name and password from the web client and verify that the user name and password are valid by comparing them against a database of authorized users. When basic authentication is declared, the following actions occur:

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 ?

spring boot or spring application in section define @Bean annottation and defining class of RestTamplate with method and return new RestTamplate().
later when we want to use ,we use to controller RestTamplate with @Autowired annottation. In this way we define RestTamplatein spring application.

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

In the context of REST APIs, when making multiple identical requests has the same effect as making a single request – then that REST API is called **idempotent**.

When we design the REST APIs, we must realize that the API consumers can make mistakes. Consumers can write the client code in such a way that there can be **duplicate requests** coming to the API.

Idemponents Http method: Get,Put,Delete,Head,Options,Trace

10 – What is DNS Spoofing ? How to prevent ?

Domain Name System (DNS) Spoofing is also known as DNS cache poisoning. **DNS Spoofing** is an attack in which DNS records are altered to redirect users to a fraudulent website that may resemble the user's intended destination.

In layman terms, your computer is tricked into thinking that it is going to the correct IP address. Once the user has landed at the destination, the victim is prompted to log into their account. This gives the attacker an opportunity to steal the victim's personal information which may include credentials and confidential data.

How to prevent?

DNSSEC: Domain Name System (DNS) Spoofing is also known as DNS cache poisoning. DNS Spoofing is an attack in which DNS records are altered to redirect users to a fraudulent website that may resemble the user's intended destination.

In layman terms, your computer is tricked into thinking that it is going to the correct IP address. Once the user has landed at the destination, the victim is prompted to log into their account. This gives the attacker an opportunity to steal the victim's personal information which may include credentials and confidential data.

Use Encryption

Use encryption like SSL/TLS which would prevent or mitigate the possibility of a website being compromised by DNS Spoofing. This way a user can verify whether the server is legitimate and belongs to the original owner of the website.

Use HTTPS

Only trust URLs that contain "https" which legitimizes a website. If the indication of "https" appears to be in flux, consider the possibility of a potential DNS Spoofing Attack

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 (REpresentational "State" Transfer) architecture, the server does not store any state about the client session on the server-side. This restriction is called Statelessness.

Each request from the client to the server must contain all of the necessary information to understand the request. The server cannot take advantage of any stored context on the server.

13 - What is CSRF attack? How to prevent ?

Cross-site request forgery attacks (CSRF or XSRF for short) are used to send malicious requests from an authenticated user to a web application. The attacker can't see the responses to the forged requests, so CSRF attacks focus on state changes, not theft of data. Successful CSRF attacks can have serious consequences, so let's see how CSRF works and how you can prevent it.

An attacker can launch a CSRF attack when he knows which parameters and value combination are being used in a form. Therefore, by adding an additional parameter with a value that is unknown to the attacker and can be validated by the server, you can prevent CSRF attacks. Below is a list of some of the methods you can use to block cross-site request forgery attacks.

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

HTTP REQUEST:
Verb: Indicate Http methods such as Get, Post, Delete, Put etc.
Uri: Uniform Resource Identifier (URI) to identity the resource on server.
HTTP Version: Indicate HTTP version, for example HTTP v1.1.
Request Header: Contains metadata for the HTTP Request message as key-value pairs.For example, client (or browser) type, format supported by client, format message body cache setting etc.
Request Body: Message content or Resource representation.

HTTP RESPONSE:
Status/Response Code: Indicate Server status for the requested resource.For example 404 means resource not found and 200 means response is ok.
HTTP Version: Indicate HTTP version, for example HTTP v1.1.
Response Header: Contains metadata for the HTTP Response message as key-value pairs.For example, content length, content type, response date, server type etc.
Response Body: Response message content or Resource representation.

1- https://www.javatpoint.com/soap-vs-rest-web-services
2- https://stackoverflow.com/questions/3370334/difference-between-acceptance-test-and-functional-test#:~:text=A%20functional%20test%20verifies%20that,that%20the%20software%20assists%20with.
3- --
4- https://stackoverflow.com/questions/90002/what-is-a-reasonable-code-coverage-for-unit-tests-and-why
5- https://www.geeksforgeeks.org/diffrence-between-put-and-post-http-requests/ OR https://restfulapi.net/rest-put-vs-post/
6- https://developer.mozilla.org/en-US/docs/Glossary/Safe/HTTP
7- https://docs.oracle.com/cd/E19879-01/819-3669/6n5sg7cfb/index.html#bncbp
8- https://www.youtube.com/watch?v=hzOf41B-xtE&ab_channel=JavaBrains
9- https://restfulapi.net/idempotent-rest-apis/
10- https://www.purevpn.com/wifi-vpn/threats/dns-spoofing
11- https://developer.mozilla.org/en-US/docs/Web/HTTP/Content_negotiation

12-https://restfulapi.net/statelessness/#:~:text=Statelessness%20means%20that%20every%20HTTP,previous%20requests%20from%20the%20client.

13- https://www.netsparker.com/blog/web-security/csrf-cross-site-request-forgery/

14-https://www.tutorialspoint.com/restful/
restful_interview_questions.htm