HW#3 Talha Alçorba

1-)SOAP vs Restful ?

Rest is more flexible and useful than XML-based SOAP, as it supports data types such as JSON, HTML, and XML.

Soap has to use a W3C standard WSDL language to generate its requests. Rest does not require this, requests can be created via the URI. REST is easier to use and design because it doesn't need a language.

In terms of security, there are ready-made functions for SOAP, and its more complexity makes it more advantageous in this regard.

Both use the HTTP protocol and provide communication between server-client. However, HTTP is required for REST and SOAP can also support protocols such as TCP/IP STMP.

REST uses simple HTTP method for caching request and is easier than SOAP. SOAP makes mixed XML requests for this process.

2-)Difference between acceptance test and functional test ?

Functional test : 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. We use any test environment we can capture at the moment, it doesn't need to be "production" calibrated as long as it's available.

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. This test should be performed in a "production-like" environment on the same or similar hardware as the customer will use.

3-)What is Mocking ?

Mocking is one of the methods used by the popular software methodology, TDD, and in particular unit testing, to isolate the system they are testing. These methods can be broadly defined as test doubles. 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:

a-)Unit tests are often written to verify assumptions about the system itself that they are testing.

b-)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) ?

With that being said it is generally accepted that 80% coverage 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. The first time you run your coverage tool you might find that you have a fairly low percentage of coverage.

5-)HTTP/POST vs HTTP/PUT ?

PUT method is called when you have to modify a single resource while POST method is called when you have to add a child resource.PUT method response can be cached but you cannot cache POST method responses.You can use UPDATE query in PUT whereas you can use create query in POST.In PUT method, the client decides which URI resource should have, and in POST method, the server decides which URI resource should have.PUT works as specific while POST work as abstract. If you send the same PUT request multiple times, the result will remain the same but if you send the same. POST request multiple times, you will receive different results. PUT method is idempotent whereas POST method is not idempotent.

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

It stands for Hyper Text Transfer Protocol, which allows you to move your hardware without implementing any useful material in transmitter and hosting. You're browsing with a website up the road, and in general, you're open to threats to protect yourself with security programs.

So what is Https? https, which stands for Secure Hypertext Transfer Protocol, adds a security measure to send and receive any information encrypted between the transmitter and the server. This is SSL Certificate. This SSL Certificate, which must be included in banks, e-commerce sites offering online shopping services and similar websites, uses different encryption methods to send and store personal information and card information written on the forms on the payment pages. Thus, it provides a secure online shopping opportunity. In other words, e-commerce sites with https:// at the beginning have SSL Certificate and guarantee card security in online shopping.

7-)How does HTTP Basic Authentication work ?

HTTP basic authentication is a simple authentication where a server can request authentication from an identification. The dictionary sends authentication to the server in an Authorization header. credentials are masked with base64.

If a browser makes a protected anonymous request, the server sends an HTTP response with code 401, a reason string indicating the authentication failure, and a WWW-Authenticate submission.

8-)Define RestTemplate in Spring ?

Many Java projects and microservices today use the Spring @RestController annotation to easily create an endpoint to expose it as a REST service. Accessing these resources can be done in various ways, and one of the most used in Spring Framework is to use the RestTemplate class found in the Spring Framework Web project.

I would like to share with you an example of a service where you can make HTTP calls to services exposed via the @RestController annotation using RestTemplate, which gives you a simple way of error handling of these calls. Having a good structure of the code needed, we can achieve a good result, then follow the following four steps:

a-)Configuration features

b-)Service route numbering

c-)RestCaller (our service)

d-)Default Error Handler

9-)What is idempotant and which HTTP methods are idempotant ?

If a method is called once and the result is the same when it is called more than once, it is a dempotent method.

Options, Get, Head, Put and Delete are idempotent methods of HTTP.

10-)What is DNS Spoofing ? How to prevent ?

DNS spoofing, also known as DNS cache poisoning, is a computer security attack in which corrupted data is placed in the DNS resolution cache by corrupting Domain Name System data. Makes the nameserver return incorrect results, eg IP address. Thus, the attacker can redirect traffic to their own computer (or another computer).

Many cache poisoning attacks against DNS servers can be avoided by relying less on information passed to them by other DNS servers and ignoring DNS records that are not directly relevant to the query. For example, BIND 9.5.0-P1 [1] and higher perform these checks.[1] Providing cryptographically secure random numbers and the randomness of the source port DNS requests come from, a 16-bit cipher and source port can greatly reduce the likelihood of successful DNS poisoning attacks.

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, a RESTful Web Service should not keep a client state on the server. This restriction is called Statelessness. It is the responsibility of the client to pass its context to the server and then the server can store this context to process the client's further request.

13-)What is CSRF attack? How to prevent ?

CSRF vulnerability; It is one of the services other than the users who use the web. We can say that this vulnerability, which occurs in systems that are not controlled from which source and how they are directed to the application, is a software that escapes the target of a programmer who is really coding a target. This vulnerability, abbreviated CSRF or XSRF, is also known as “Session Riding”.

Important requests sent by the user to the system should be received with the POST method. The most popular method to prevent Cross-Site Request Forgery (CSRF) is to provide the user with a unique randomly generated "token". Called CSRF Token or Synchronizer Token, this method works like this: The web server creates a token. (This token is regenerated as the transaction is made.) The token is stored in the form as confidential information. The user performs the POST operation.

Token information is included in the POST data. The web application compares the token generated by the system with the token sent in the request. If the token data matches, it is understood that the request was sent by the real user and the request is approved. If there is no match, the request is denied. In this way, malicious requests are blocked.

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

A simple request credit from a customer computer consists of credits:

a-)To get a resource, a request is required information, a request is GET /content/page1.html, a resource called /content/page1.html is requested from the server.

b-)Titles (Example – Accept-Language: EN).

c-)An empty line.

d-)body at will.

All lines must end with a carriage return and line feed. The blank space must contain the carriage return and line feed that are not in it.

A simple response from the server includes the following components:

a-)HTTP Status Code (For example, HTTP/1.1 301 Moved Permanently means that the requested resource has been permanently moved and redirected to another resource).

b-)Headings (Example – Content Type: html)

c-)An empty line.

d-)An optional message body.

All lines in the server response must end with a carriage return and line feed. As with the request, the blank line in the response must have only a carriage return and a line feed, without spaces.