**1** – SOAP vs Restful?

SOAP: A Web Service is a structure that serves other clients over the HTTP protocol. SOAP, on the other hand, is a standard used to create a Web Service, regardless of the programming language.

SOAP consists of four basic functions. These are Envelope, Header, Body, Fault.

Envelope: It is the ROOT element of the SOAP structure and is mandatory.

Header: Used for authorization settings.

Body: Information about methods or the result of methods is found here.

Fault: As the name suggests, error codes and error reports of occurring errors are found here.

RESTFUL is a protocol. The comparison between RESTFUL and SOAP is not hardly true, because SOAP is API architect but RESTFUL is a service.

Main difference between these in usage is text transferring format. SOAP transfers just XML file but RESTFUL can transfer different type of text file such as JSON.

**2** - Difference between acceptance test and functional test?

Functional Test seems subtest. For example, if we check the “opening door” function, we observe to opening door or not. We don’t care “ the door opening which side?”, or” Is door locked? “, we check just a function.

Acceptance Test seems final test. The test question is “Does it fulfill user requests?”. In the same example, “Is the house livable?” or different users queries such as “bathroom size”.

**3** - What is Mocking?

Mocking is uses for Unit Test. For example, in a class takes a parameter and it call different api with this parameter, the response of api is a text and we already know it. The comparison of the response and answer is Mocking.

Shortly, it is services which is, test the function is getting true result with given specific sample.

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

Generally, it is 70-80%, because of time-cost-effect relationships. But it could be higher than 80% in the automotive or medicine industry. Independent from this reason, Development Cycle Status is affects it, for example if system is old and the creator are no longer available, the code coverage can be lower.

**5** – HTTP/POST vs HTTP/PUT?

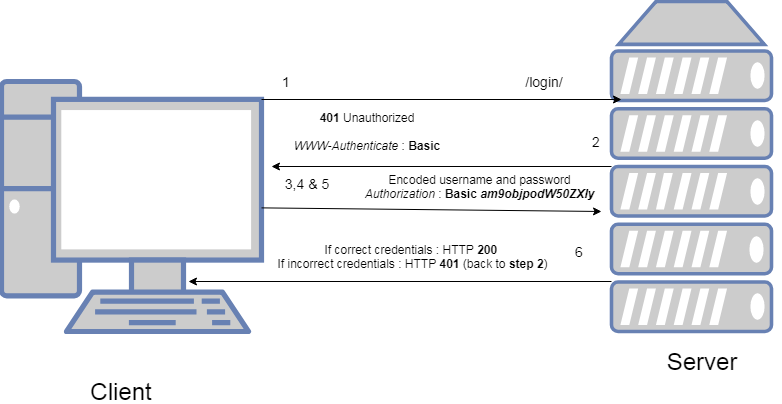
An HTTP PUT is supposed to accept the body of the request, and then store that at the resource identified by the URI. An HTTP POST is more general. It is supposed to initiate an action on the server. That action could be to store the request body at the resource identified by the URI, or it could be a different URI, or it could be a different action. PUT is like a file upload. A put to a URI affects exactly that URI. A POST to a URI could have any effect at all.

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

The Safe methods are GET, HEAD, or OPTIONS. These are thinkable read only. If a method can change anything, it is Unsafe method such as PUT and DELETE.

**7** - How does HTTP Basic Authentication work?

That seems a simple username-password login. While client connecting to server, Server wants an Authentication and the process is starts.



**8** - Define RestTemplate in Spring?

RestTemplate is library in Spring library and it carry out request of HTTP.

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

Idemotent methods are no affects to the server, in other words, an idempotent method should not have any side-effects (except for keeping statistics). In the question of 6, all the safe methods are idempotent but all idempotent methods are not safe methods. The [GET](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/GET), [HEAD](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/HEAD), [PUT](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/PUT), and [DELETE](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/DELETE) methods are idempotent.

**10** – What is DNS Spoofing? How to prevent?

While we are opening to internet, we are using our specific DNS, but open to public wifi networks could be unsafe because of DNS visibility. If we want to connect to “google.com” in public wifi, the website name converts to an IP, if IP defined different from the original IP, our website name is poison and we are not safe.

DNSSEC is using to protecting from DNS Spoofing. DNSSEC prevents DNS Spoofing by encrypting DNS.

**11** – What is content negotiation?

Content Negotiation is between client and server. The purpose is giving a different response with same URI.

**12** – What is statelessness in RESTful Web Services?

When server gave response and send the XML file to client, server does not keep the client information. It is advantage for less system load but it causes to send same information again to client.

**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 cyberattacked 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 authorized users without your explicit permission.

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.

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

These are communication with frontend(client) and backend(server). Client has e request and server gives a response. This communication components are Status Line, Headers, Body.

Status Line: The Status Line contains three important components – HTTP Version, HTTP Response Code, and a Reason-Phrase.

Response Header:The Response Header contains the information about the content that is being returned in response together with data about the Server that sent it.

Body:In case of a successful response, the body of the Response Message is used to serve the Client/User with the resource asked for in the request