PayCore Java Spring Bootcamp [2.3]

Homework 3#

28.01.2022

QUESTION 1: SOAP vs Restful?

Answer:

SOAP is a protocol which is made for communicate on Internet. In the other hand, Restful is an architecture style which is made for revolving around the resource where each component is a resource and a resource is accessed by a common interface using HTTP standard methods. They have many differences. For example:

* REST Because of SOAP is protocol, SOAP cannot use REST. But on the other hand REST is an architecture style and it can use protocols like SOAP or HTTP.
* If we look at security, SOAP can define its own security. On the other hand, RESTful web services inherits security measures from underlying transport.
* REST does not define too much standarts like SOAP.
* In SOAP web services, JAX-XS is used for java API. In RESTful web services, JAX-RS is used for java API.
* For exposing the business logic, SOAP uses service interfaces but REST uses URI.

QUESTION 2: Difference between acceptance test and functional test?

Answer:

Functional Testing: It is the type of test that confirms whether each function of the application we are testing is working in accordance with the given requirements.

Acceptence Testing: In order to verify the exact needs or what the customer needs, acceptence testing comes. It is used for defining whether the software or the application that you have build whether that software fulfills the customers requirement and it is fit for use whatever customers needs.

QUESTION 3: What is Mocking?

Answer:

Mocking is a popular mehod that is used for isolating systems which are tested. With other words, while we testing our code, we create fake objects instead of using our real objects and this is called mocking. By doing this, we can give the response we expect from that process to the code with the mock object. With mocking, if the service you wrote is connected to a component that has not been written yet; You can overcome this situation by typing mock.

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

Answer:

Code covarage is the ratio of the amount of tested code in the code we write to the whole code we write. If our covarage is higher, that means that we can use more unity tests for our code. It is used to identify mistakes. It also means, in entire set of code, how many things on it have unit tests. Having "100% code coverage" doesn't mean everything has been fully tested. This means that every line of code is tested, but not in every (common) case.

QUESTION 5: HTTP/POST vs HTTP/PUT?

Answer:

HTTP/POST is a method that used to create a new resource. HTTP/PUT is a method that used to update a existing resource or create a new resource. If we want to transfer our datas on server we can use HTTP/POST. We can say that HTTP/PUT is a little bit specialized method against HTTP/POST. What we can do with HTTP/PUT, we can also do with HTTP/POST.

QUESTION 6: What are the Safe and Unsafe methods of HTTP?

Answer:

Safe method is the HTTP method that will not change situation of server. If it is changing the situation of servers it is not considered as safe but considered as unsafe method. For example, GET method is unsafe method because when we make a request with the get method, we can clearly display this request on the browser. This situation is making troubles for our security system. POST method is used for making client informations private. It is a safe method. The information to be sent with the post method is included in the message body, therefore no information is displayed on the url address.

QUESTION 7: How does HTTP Basic Authentication work?

Answer:

HTTP Basic Authentication important for securing our datas. With using it, we basicly want to prevent attacks that we got from outside and try to make our datas safe.

The steps of how it works:

1. The user (client) sends a request to the server.
2. The server sends the HTTP status code as 401 and the basic value WWW-Authenticate variable in the HTTP Header in the HTTP header. The reason he's posting this is for basic authentication to start.
3. When the user receives this request, an input field like the one below appears.
4. User enters username and password values ​​here. The values ​​here go in the background by putting a colon between them and encrypting with base64.
5. The user sends this under the Authorization header under the HTTP protocol, as a basic value at the beginning.
6. The server, on the other hand, compares the base64-encrypted user name and password information in the Authorization header from the user with the base64-encrypted data within its body. If this comparison is true, it returns 200 as the HTTP status code. But if it is wrong, it returns 401 status code with WWW-Authenticate and if coding is not desired to redirect somewhere, it is returned to step 3.

QUESTION 8: Define Rest Template in Spring?

Answer:

Rest Template is a central class in Spring framework. It is for executing synchronous HTTP requests on the client side.

QUESTION 9: What is idempotant and which HTTP methods are idempotant?

Answer:

For HTTP, idempotant means multiple identical requests will have the same result. This HTTP methods are idempotant: GET, HEAD, OPTIONS, TRACE, PUT ve DELETE. Here is a table shows that which methods are idempotant and which are safe:

|  |  |  |
| --- | --- | --- |
| **HTTP Method** | **Safe** | **Idempotant** |
| GET | Yes | Yes |
| AHEAD | Yes | Yes |
| OPTIONS | Yes | Yes |
| TRACE | Yes | Yes |
| PUT | No | Yes |
| DELETE | No | Yes |
| POST | No | No |
| PATCH | No | No |

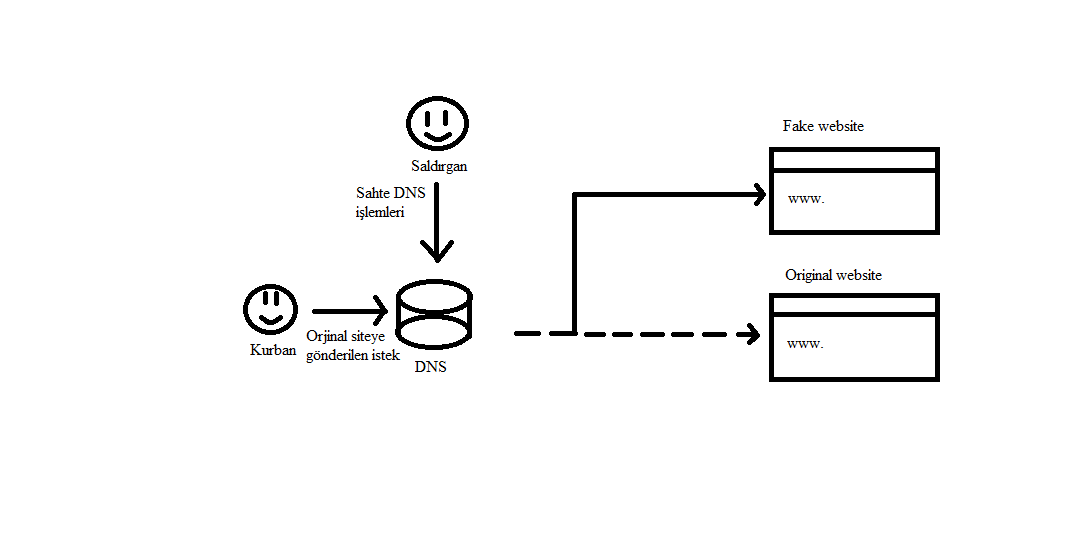
QUESTION 10: What is DNS Spoofing? How to prevent?

Answer:

DNS is used to translate a domain name to a specific IP address. DNS spoofing generally covers cyber attacks where hackers redirect web traffic to fake web servers and phishing websites.

How this attacks works:

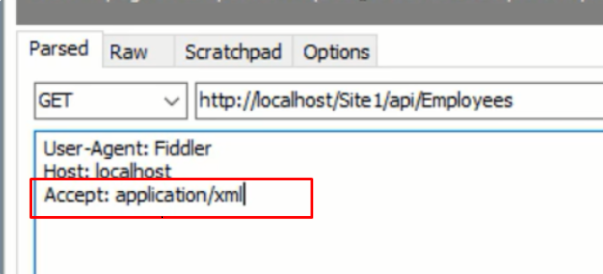
By gaining access to the DNS cache, attackers replace the real IP address with the IP address of a fake website, allowing users to reach the fake website that was specially crafted for fraud instead of the real website. DNS spoofing is very difficult to detect as the fake website looks exactly the same as the original website the user is trying to access.



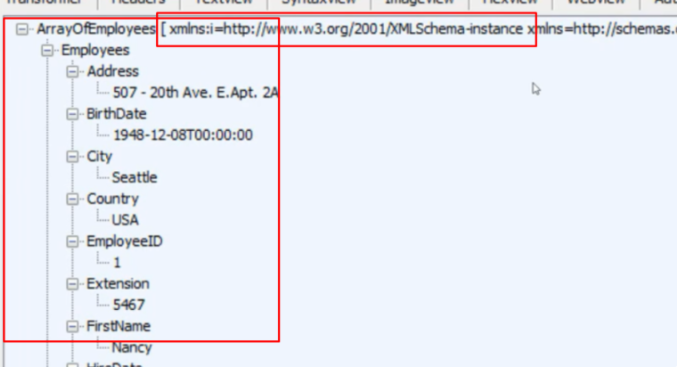
QUESTION 11: What is content negotiation?

Answer:

It can be said that content negotation is a content agreement between client and server. The purpose of content negotation is to be able to serve content in different document types with the same URI. This is one of the conveniences that HTTP has given us. For example if we have json and xml support, we can text Accept:application/xml to our header and transform our response toxml file. If we don’t do that, its response will be json as default. Lets see an example:



In here we defined our exception as xml file. The response will be like this:



QUESTION 12: What is statelessness in RESTful Web Services?

Answer:

In REST architacture, there are rules. One of them is called statelessness. It means that a RESTful Web Service should not keep a client state on the server.

QUESTION 13: What is CSRF attack? How to prevent?

Answer:

CSRF is the use of a user's session logged in any web application to perform operations against the user's wishes. It is a security problem that cause many things. It is a dangerous situation and its harm is parallelled with the user's authorization which is using the application at the moment. Let's say that someone who is checking their bank account is also checking their mail on the side tab. When the target person clicks on a malicious link sent by the attacker, the attacker can access the bank page and data open in the target's side tab, although the target user is in the mail tab. Because the target's session on the bank page is still active, the information needed by the attacker is still available in the browser. As long as the target user does not log out, the attacker can secretly carry out transactions such as "money transfer, credit withdrawal" that can cause serious damage in the background. However, for this scenario to happen, the target user does not necessarily have to click on a malicious link. If the website used has this vulnerability, As soon as the page loads, a door is opened for the attacker, so to speak. Or, the target user's session can be infiltrated as soon as the page loads, even without the condition of clicking, through any malicious site.

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

Answer:

HTTP request has 5 main components. And these components are; Method, URI, HTTP Version, Request Header, Request Body. These components are representing something. Method component tells what methods the request operation represents. URI is used for uniquely identifying the resources on the server. HTTP Version indicates what version of HTTP protocol you are using. Request header has the details of the request metadata such as client type, the content format supported, message format, cache settings, etc. Request body represents the actual message content to be sent to the server.

HTTP response has 4 main components. And these components are; Response Status Code, HTTP Version, Response Header, Response Body. These components are representing something. Response Status Code represents the server response status code for the requested resource. We can give an example about this. For example, 400 represents a client-side error. Another component is HTTP Version. It indicates the HTTP protocol version. Response header has the metadata of the response message. This data that we are describing right now can describe what is the content length, content type, response date, what is server type, etc. The last component is Response Body. Response Body contains what is the actual resource/message returned from the server.