Egill Anton Hlöðversson, egilla14, kt. 190193-3309
Edda Steinunn, eddasr15, kt. 241095-2909

Class assignment I

1. What is a server?
A computer containing some software a *client* communicates with via network.

2. What is a client?
A client is a computer application that accesses resource.

3. How do servers and clients interact with each other? Provide an example.
The most common way a server and a client interact with each other is when a client requests some resource from a server which then responds (with requested data). An example of this would be a web browser that sends a HTTP GET request to some URL (f.x. when I navigate to www.google.com via my browser it sends a HTTP get request to google servers for www.google.com), and the server responds with a HTTP response that contains website content (said google server serves my browser the content needed for my browser to render www.google.com)

4. What is the most used protocol for communication on the internet and why?
The HTTP protocol (Hypertext Transfer Protocol) is the most commonly used protocol for communications over the internet. The reason why it is so dominant for internet communications is that the World Wide Web was initially created with the HTTP protocol in mind for as means of communication, thus HTTP is well standardized and fairly lightweight.

5. What is the difference between server-side rendering and client-side rendering?
Server-side rendering refers to the type of server-client communication when server responses contain fully formed and rendered HTML content, in other words the server handles all heavy workload and content scraping before sending it to the client. ***Client-side rendering*** however refers to the type of server-client communication when server responds with raw data so

that client must do the heavy workload and content scraping as well as rendering the HTML.

6. What are the pros and cons of server-side rendering?

Pros:

- The initial page of the website loads faster
- Beneficial for static sites as client receives fully rendered content
- Good for Search Engine Optimization (SEO)
- Requires less work from the client which suites well for smaller and/or lightweight clients.

Cons:

- Difficult to request specific data and most often to load a batch of data is loaded
- More frequent server requests that results in more traffic to the server.
- Slow page rendering, instead of render each part of the website bit by bit, it needs to render all the HTML at once and the user has to wait until it has been loaded to interact.
- Full page reloads
- Non-rich site interactions

7. What are the pros and cons of client-side rendering?

Pros:

- The server does not need to do modifications and can send pure data
- Different clients can render data differently from the same resource
- Client-side rendering is not constrained by the programming language server uses, communications become not specific to some programming language
- Sites can be dynamic due to partial updates being enabled
- Websites are faster again due to partial updates being enabled

Cons:

- Data is generally not loaded until all JS files have been fetched
- Not beneficial for SEO (Search Engine Optimization)
- Initial load is slower than that of server-side rendering
- Client-side rendering often requires an external library

Egill Anton Hlöðversson, egilla14, kt. 190193-3309
Edda Steinunn, eddasr15, kt. 241095-2909

8. What are the benefits of using a web service?

Using a web service allows you maintain and distribute data to many different clients at once efficiently. Good web service architecture uses API that allows reuse of the same code the can be used on many client platforms and OS. Today web services generally allow scalability and can handle a lot of requests at once without affecting the functionality of clients.

9. Name all the different type of web services and provide a concise explanation of each type.

1. ***RESTful Resource Oriented Web Services*** are of web service architecture type in which the information needed for request or response is inferred by the HTTP method and any scoping information is specified in the URI. In other words, the HTTP method must match the method information for a service to be considered RESTful.
2. **RPC-Style Web Services** are of web service architecture type in which *“envelopes” of data* (which roughly means that data is encapsulated with method information and other information often in the form of XML) are received from client as requests and same type of data “envelopes” are sent back to client as response. SOAP is the most common web service of this architecture and are often doubly-encapsulated, i.e. contain data in “envelopes of envelopes”.
3. ***REST-RPC Hybrid Services*** are of web service architecture type which have attributes making them between RESTful architecture and RPC-Style architecture - i.e. they share some characteristics with both architecture but do not quite fit neither category. For example, a web service that transmits scoping information and method information via URI does not qualify as RESTful web service (due to the method being exposed in the URI), making it a REST-RPC Hybrid.

10. What are the benefits of RPC web services?

Generally RPC web services support:

- WS-Security, thus communication may be more secure
- WS-AtomicTransactions (ACID compliant) which may enable atomic synchronous actions

- WS-ReliableMessaging, providing features of retry logic if communication fails
- Access to WSDL (Web Service Description Language), meaning less or no need for documentation and facilitation of writing and maintaining services

11. What are the benefits of RESTful web services?

- RESTful web services generally support a wide range of data formats whereas RPC web services generally only uses XML
- Performance and stability is better (f.x. due to more lightweight communications)
- RESTful services use HTTP more efficiently and correctly, whereas in RPC-Style web services the HTTP method is disregarded and treated as irrelevant information
- With RESTful services mapping response is considered responsibility of consumers

12. What is SOAP? How does a SOAP web service differentiate from a RESTful web service?

The acronym SOAP stands for subjective, objective, assessment, and plan. It an XML based messaging protocol used by SOAP web services. The biggest difference between SOAP and RESTful web services is that SOAP uses a messaging layer to encapsulate its data and often doubly encapsulates it while REST has no messaging layer. Additionally, with SOAP web services, the HTTP method is often disregarded as irrelevant and the method is specified in the “envelope” (encapsulation) while with RESTful the HTTP method is actually used as method.

13. What is a WSDL? What is the purpose of a WSDL?

The acronym WSDL stands for ***Web Service Description Language*** and is language which enables automated code-generation such that a web service that supports WSDL can describe what it does, how it does it, and how consumers of that web service can go about fetching resources or data from it. The purpose of WSDL is to facilitate documentation of web service, it’s maintenance, changes to the service without breaking the client code and they make it easy to consume web services by reducing amount of code of client application

Egill Anton Hlöðversson, egilla14, kt. 190193-3309
Edda Steinunn, eddasr15, kt. 241095-2909

14. Why is documentation useful for web services?

Documentations are useful for web services so that developers that are interested to connect to and to use the web service know how to setup a client that can communicate with the server and be able to request its resources. Documentation also facilitates the maintenance and changing of the web service for the web service's developers.

15. What are some common documentation libraries for web services? Name a few.

The most common enterprise solution is called *Swagger* and is even recommended by the *.NET Core* team at *Microsoft*, but there are also some free open source solution such as *Pronovix*, and *npm*.

16. What is authentication?

The process in which user credentials are compared to those stored within a system database. If the credentials match the user is a valid user within the system, otherwise the user is not a valid user and should not be able to access the system.

17. What is authorization?

Authorization determines what the user is allowed to see or do within the system.

18. What is the difference between authentication and authorization? Are they describing the same thing?

Authentication and authorization are both are security control methods within a system but have two distinguished responsibilities. Authentication is a method used to protect the system from invalid users (that is, users that are not allowed within the system), and authorization is to protect specific data accessibility from users (whether they're allowed within the system) and can be used to restrict system user access to specific resources.

19. Name at least three common security vulnerabilities in web services and describe each vulnerability briefly.

Cross-site scripting attacks (XSS attacks) are a security vulnerability that usually happens when either web applications or web services do not properly validate client input, resulting in malicious code running due to lack of encode validation. Ex: a user can be redirected to a fishing site instead of intended site.
SQL injection attacks are the type of security vulnerability that occurs when a user executes a SQL queries directly, e.g. if the user sends a data through a search form which is used directly in a SQL query. This query might include SQL commands such as DROP DATABASE
Cross-Site Request Forgery (CSRF) (commonly phishing) is a security vulnerability in which the attacker tricks the victim into submitting malicious request. This can happen for example when user clicks a link to a scam website which bait the user to type in personal information the attacker can use (with malevolent intent)

20. What is the difference between HTTP and HTTPS? What does the extra “S” stand for?

HTTP and HTTP are web protocols that client-server systems use to communicate over the internet and differ in terms of security. HTTPS encrypts that data between the client and the server whereas with HTTP everything is sent in clear text, thus communications between server and client become more secure with HTTPS. The “S” distinguishes the two and stands for *Secure*.

NOTE: THIS ASSIGNMENT WAS SOLVED USING COURSE LECTURES AND COURSE LECTURE SLIDES PROVIDED BY TEACHER AS REFERENCE.