HOMEWORK6
FATMA BETÜL UYAR

1 – What is the difference between manual testing and automated testing ?

- **Manual testing** is conducted **by a human**. No tool or software support.
- **Automated testing** is conducted **by tools** or **software automations**.
- **Automated testing** is **faster** and **more secure** than Manual testing.
- **Automation** does **not allow random** testing.
- **Automated testing** does **not involve human consideration**. So it can never give assurance of **user-friendliness.**

2 – What does Assert class ?

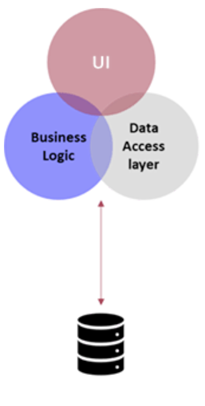
Assert is a method **useful** in determining **Pass or Fail status** of a **test** case. For example, if you want to test that a boolean condition is true or false, you can use assert.
Throws an **Assertion Error** if the associated assertion condition is not true. If the condition is true, program **flow goes on**. You can import assert like this;
import static org.junit.Assert.*;
Some assert methods;

- **assertNull**(object),
- **assertNotSame**(expected, actual),
- **assertEquals**(expected, actual)
- **assertArrayEquals**(airethematicArrary1, airethematicArrary2);

3 - How can be tested 'private' methods ?

From what I've read, testing private methods is not a very correct approach, *"Just call it."is more correct*. Instead of testing private methods, we can test the **public methods** that **call them**. Test our public methods and verify that they work, and our private methods will be covered, too.

4 – What is Monolithic Architecture ?



MONOLITHIC ARCHITECTURE

Monolithic is an **all-in-one architecture**, wherein all aspects of the software operate as a single unit. In other words, it is like a big container, wherein all the software components of an app are assembled.

Advantages;

- simple to develop
- simple to test
- simple to deploy
- simple to scale using the load balancer.

Disadvantages;

- hard to maintenance, I mean as the project grows,it is challenging to make changes fast and correctly.
- start-up time may be longer due the application size.
- If you do an update, you have to deploy the application again

5 - What are the best practices to write a Unit Test Case ?

- **Well-organized test practice**

Make your unit testing processes scalable and sustainable.

- **Well naming convention**

The better it is named, the easier it is to understand.

- **Trustworthy**

Successful tests should not output

- **More automated tests**

Automated tests helps to work efficiently, faster feedback

- **Only one condition at a time**

Apply to test a single use-case at a time

- **Should be isolated**

The test needs to be completely isolated from any other unit or dependencies.

- **Aims to %100 code coverage**

6 - Why does JUnit only report the first failure in a single test ?

If the unit test is too large and does a lot of things, multiple failures are reported. JUnit is designed to work best with a number of small tests. It executes each test within a separate instance of the test class. It reports failure on each test.

7 - What are the benefits and drawbacks of Microservices ?

Benefits ;

- Enables the **continuous delivery** and **deployment** of complex applications.
- **Simple scale**
- Services are smaller and **faster to test.**
- Services can be **deployed independently**.
- **Each team** can **develop**, **deploy** and **scale** their services independently of all of the other teams.
- The application **starts faster**, which makes developers more productive, and speeds up deployments
- **Microservices has its own database**
- **Improved fault isolation.** When the error occurred, only that service was affected.

Drawbacks;

- Additional **complexity** of creating a distributed system.
- More **difficult implement**
- More **difficult debug**
- More **difficult find** and **managing exception**
- Increased **memory consumption**.

8 - What is the role of actuator in spring boot ?

If we want to get **production-ready features** in an application, we can use the Spring Boot actuator.It provides **monitoring** of our application's endpoints. We can use **HTTP** and **JMX** endpoints to manage and monitor the Spring Boot application.

Actuator has 3 main feature.These are **endpoints**,**metrics** and **audit**.

In summary, Actuator **makes it easy** to collect metrics, understand traffic or know the status of the database.

9 - What are the challenges that one has to face while using Microservices ?

- **Data Consistency:**

It is getting hard to manage data consistency in distributed systems. It can be duplicate datas, so there may be redundancy across the data stores. For this challenge, we can use the SAGA pattern.

- **Security**

Because the data is distributed, it becomes difficult to maintain the confidentiality of the data.For security, we can use API Gateways. AWS, Spring Cloud Gateway,Jwt are some of them.

- **Communication**

Independently deployed microservices act as miniature standalone applications that communicate with each other.There are the different way to communicate microservices ;

a. Point to point using API Gateway
b. Messaging event driven platform using Kafka and RabbitMQ
c. Service Mesh

- **Testing**

Testing is much more complex in a microservices environment due to the different services, complex integration, and interdependencies.

- **Data Staleness**

The database should be always updated to give recent data. The API will fetch data from the recent and updated database.

- **Technological Diversity**

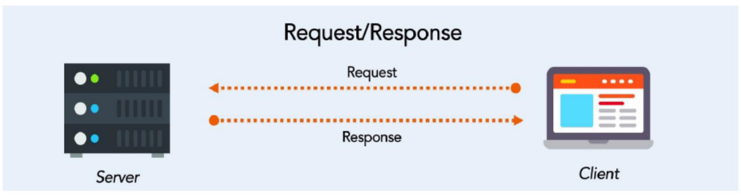
The microservice paradigm allows you to use different technological bases for each service. This, however, may result in having to use different tools for the same functionality – just because a different microservice is using a different technology.

Fault Tolerance, DevOps Support ,Monitoring & Performance…

10 - How independent microservices communicate with each other?

There are 3 ways for the communication;

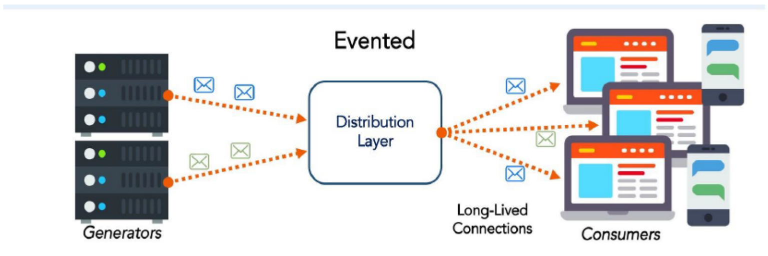
- **Request-Driven (Synchronous):**



This architecture can be designed using different technologies, for example Rest.

A service makes a direct **request** to **another service** that it needs data from. If we use Rest services, the requests will GET, POST, DELETE and UPDATE requests.

- **Event- Driven (Asynchronous):**



Event Driven architecture aims that the services that need to communicate with each other can complete the process in the distributed system by throwing an event and handling this event. Inter-service events are handled via a central **Event Bus**. All event traffic will pass through this bus.

- Hybrid

Event and Request , it contains both.

11 - What do you mean by Domain driven design ?

Domain Driven Design is an approach that helps us solve and manage the complexity in our project, and also allows us to make our project sustainable.

Ubiquitous Language

Every service that we will use in our project must have an equivalent in the domain. Thus, everyone involved in the project can speak this common language and understand each other.

Bounded Context are structures that are separated from each other and their boundaries are determined.

Diğer servisler olmadan, onlara bağlı olmaksızın geliştirilebilen, çalıştırılabilen, bağımsız otonom birimlerdir.

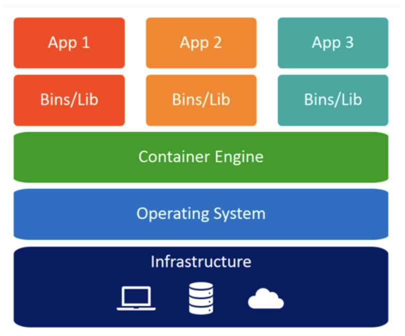
Entity : These are structures that have a unique value of their own.

Value Object : Structures that do not have a unique value of their own.

Aggregate Root: It is the coexistence of related entities.

12 – What is container in Microservices ?

- Containers are the result of a packing mechanism that **decouples applications from the environment** where they would normally run.
- Containerizing allows applications to be **deployed much easier**, use **fewer resources to run**, and **stay isolated from an environment** where they could cause or be affected by issues.
- Containers provide **a lightweight**, **fast**, and **isolated infrastructure** to run your applications

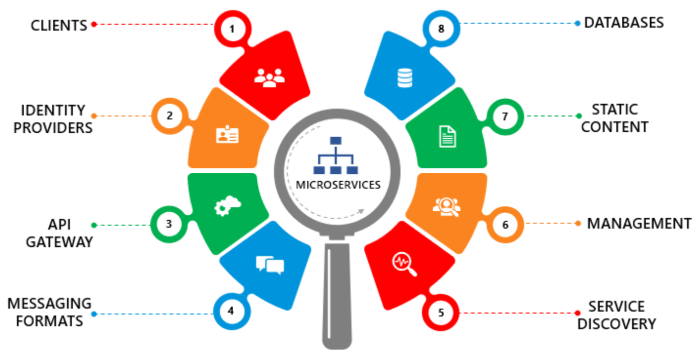


Containers

- The **application**, **dependencies**, **libraries**, **binaries**, and **configuration files** are usually bundled into the container, providing **an easy solution to migrating** your application anywhere.

- The average container size is usually **less than 100MB**, as opposed to a couple of gigabytes used by a virtual machine.

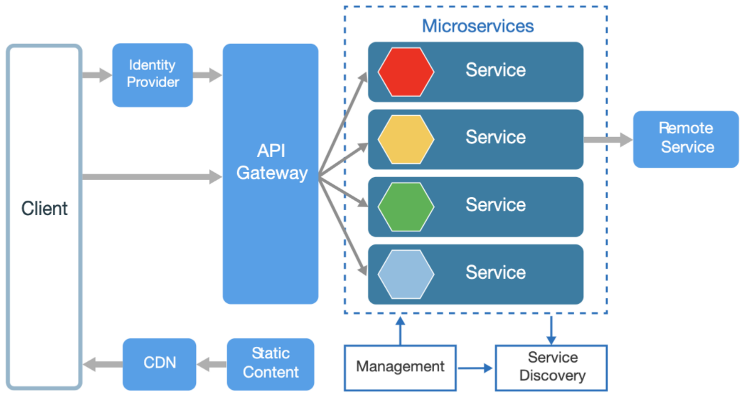
13 - What are the main components of Microservices architecture ?



Main components are;

- clients
- identity providers
- api gateway
- messaging formats
- databases
- static content
- management
- service discovery

14 - How does a Microservice architecture work?



1. Clients

The architecture **starts** with different types of clients

2. Identity Providers

Client's requests are then passed on to the identity providers who authenticate the requests of clients and communicate the requests to API Gateway.

3. API Gateway

API Gateway acts as an entry point for the clients to forward requests to appropriate microservices.

4. Messaging Formats

There are two types of messages, synchronous and asynchronous.

5. Data Handling

Well, each Microservice owns a private database to capture their data and implement the respective business functionality.

6. Static Content

After the Microservices communicate within themselves, they deploy the static content to a cloud-based storage service that can deliver them directly to the clients via Content Delivery Networks (CDNs).

7. Management