**Summary – MS**

MS common structure

Controller -> service -> repository

Controller -> service -> another MS

* Controllers are the “gate” of the application. It consists of annotated methods which are the root of the REST services. Controllers usually injected with services.
* Services are usually the middle layer that connects between controllers and repositories or other MS. Meaning, that they are responsible to activate repository operations based on the REST service that was called. They are also responsible to call other MSs if needed.
* Repositories are responsible to perform DB actions of CRUD.

MS common architecture

MS ‘B’

MS ‘A’

Configuration MS

Eureka MS (discovery MS)

Discovery MS

Eureka Server is an application that holds the information about all client-service applications. Every MS will register into the Eureka server and Eureka server knows all the client applications running on each port and IP address. Eureka Server is also known as Discovery Server.

Special annotations

* @Service (annotates class) – defines a class that it's holding the business logic. So there's not any other specialty except using it in the service layer.
* @HystrixCommand(fallbackMethod, commandProperties) (annotates method) – defines a method that uses a fallback mechanism in case it fails. Usually annotates methods that call other MSs. fallbackMethod defines a method inside that class that will be called in case all the conditions in commandProperties will be met. commandProperties are properties that define conditions for fallback mechanism.
* @HystrixProperty(name, value) (annotates a property inside commandProperties that’s inside @HystrixCommand) – defines a condition that the fallback mechanism will follow. For example: name="execution.isolation.thread.timeoutInMilliseconds" and value=”2000” - this is the timeout that determines if the MS that we are calling is down.
* @Value(<property value>) (annotates a class member) – defines a field that will be populated with a value from a property file. The value should be written with ‘$’ and {}. For example: "${api.key}".
* @EnableEurekaClient (annotates a class) – registers the MS as an instance to the Eureka server.
* @EnableEurekaServer (annotates a class) – defines the MS as a Eureka server (discovery server).