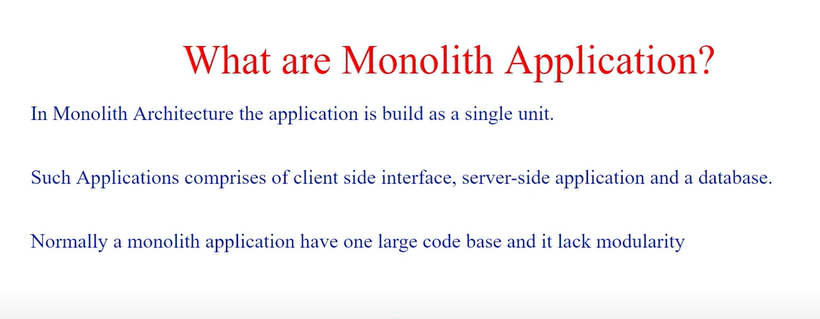
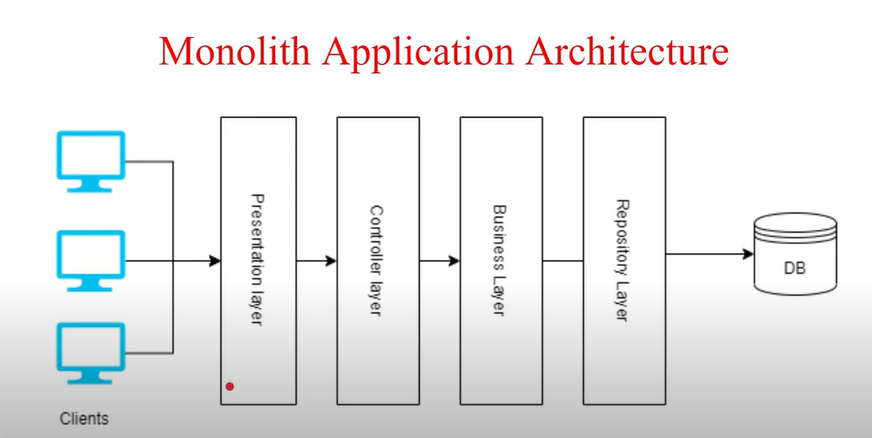
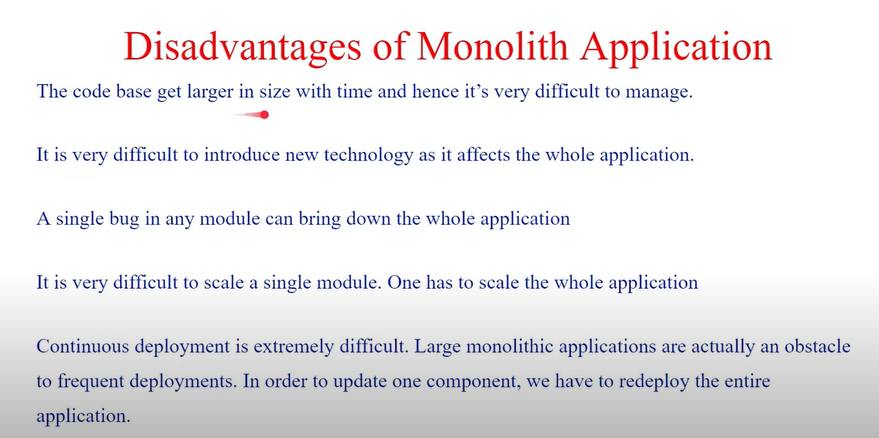
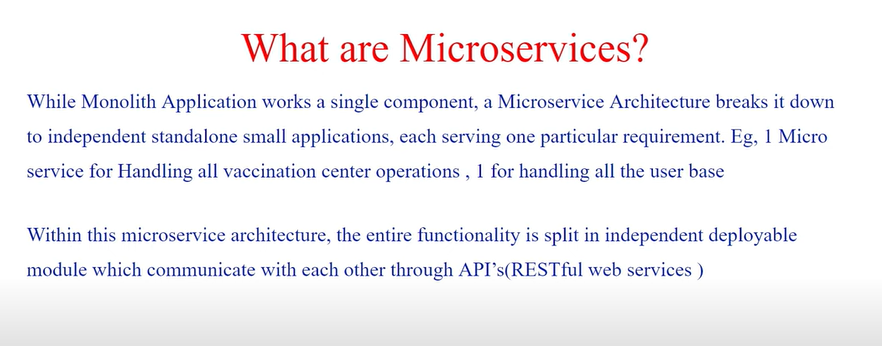
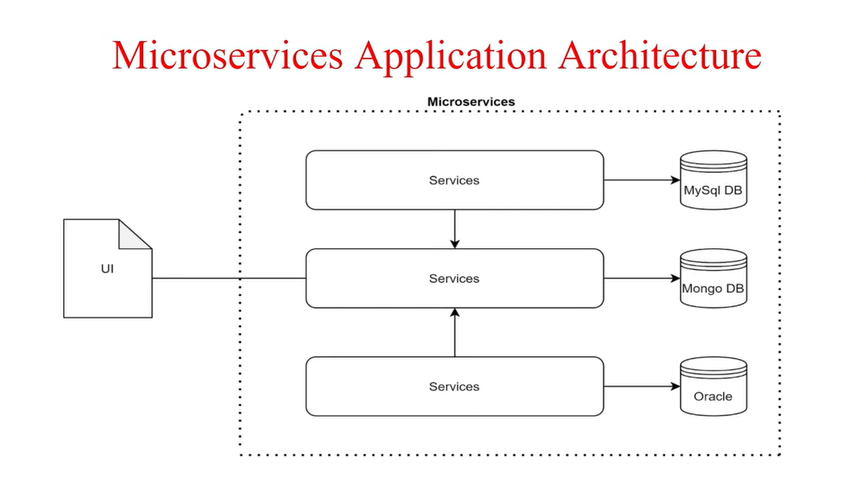
**Micro-Services Tutorials:**



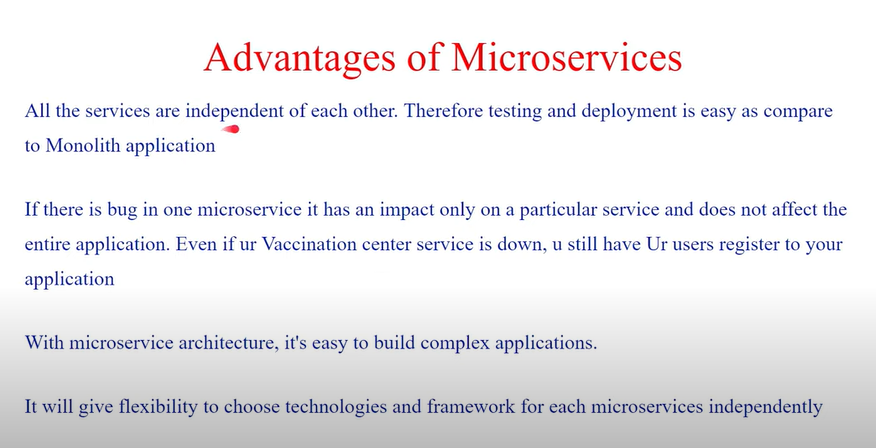


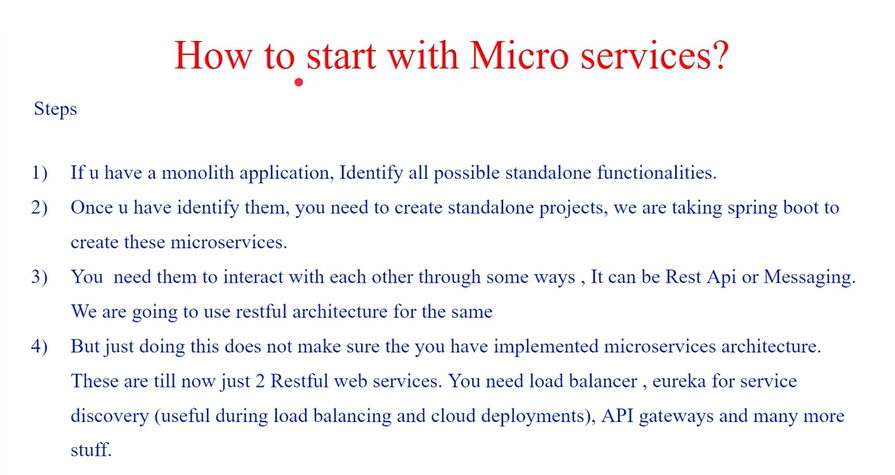


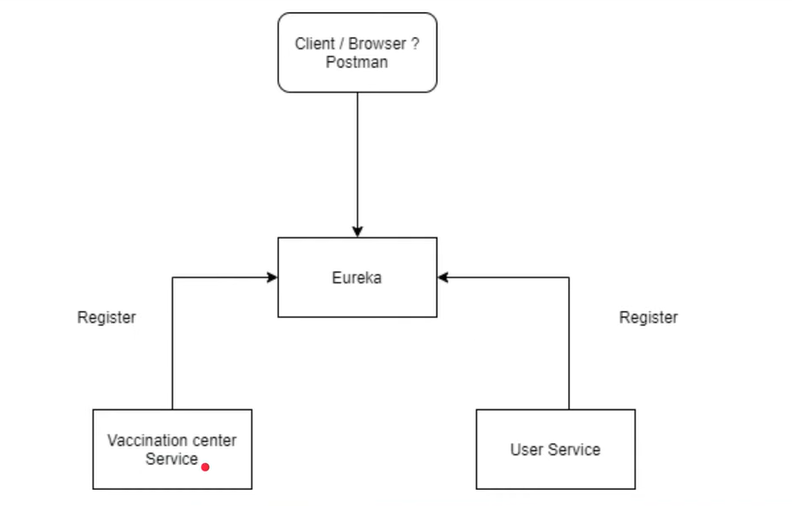




* This is the very basic architecture it has more components which we are covered in later slides





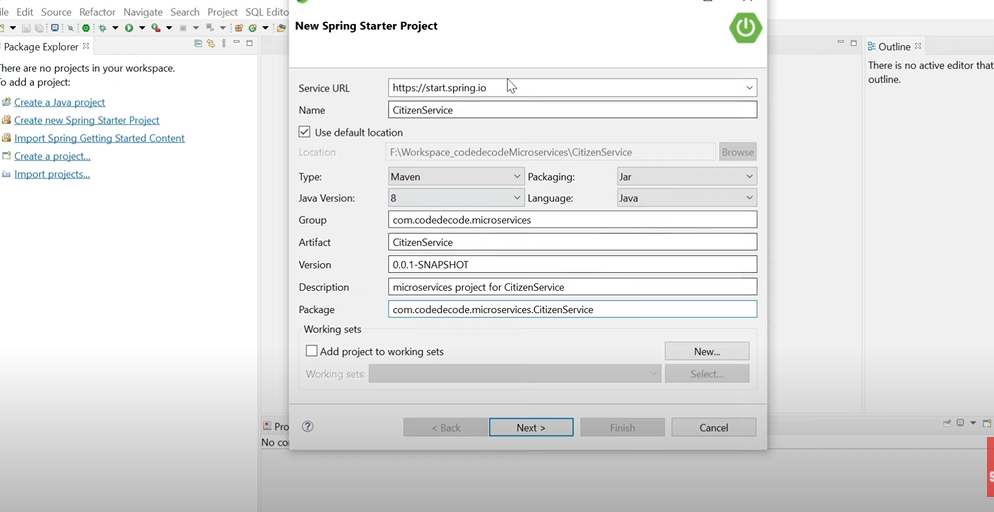


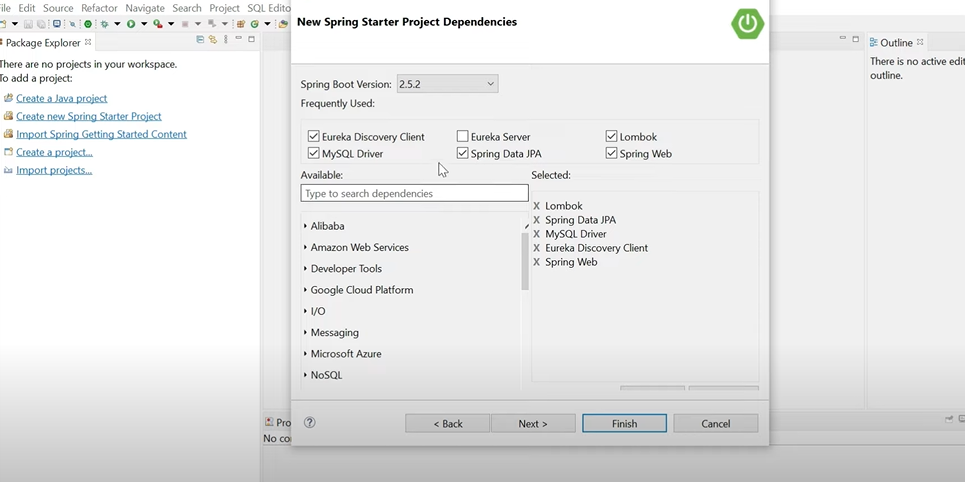
* We are going to create three microservices using spring boot and rest APIs
* **Vaccination Centre Service**: To handle vaccination centre related functionalities
* **User/Citizen Service**: To handle User related functionalities
* **Eureka Service**: Basically it takes the corresponding URL which is being hit by the services to interact with the other services and based on the load (Load balancing) it hit the required URL and responds accordingly.

Or it handled the load balancing situations and handles the dynamic URLs.

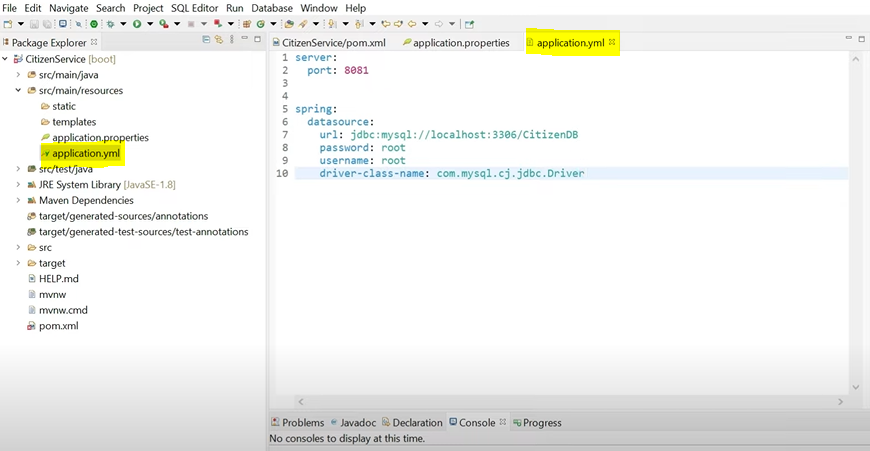


* Let’s create the project: **User/Citizen Service:**

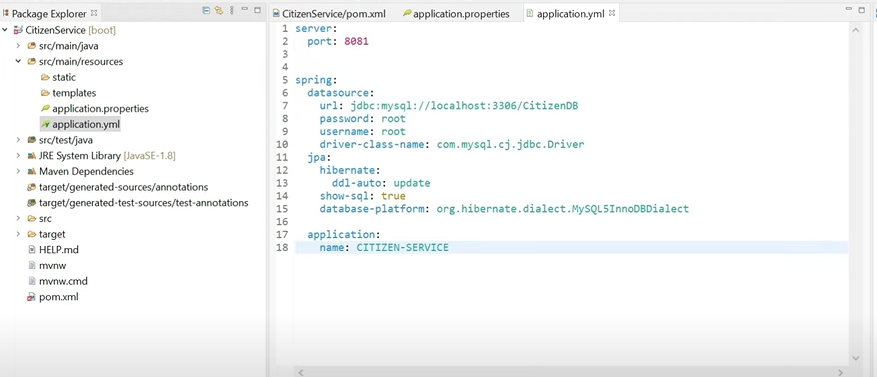


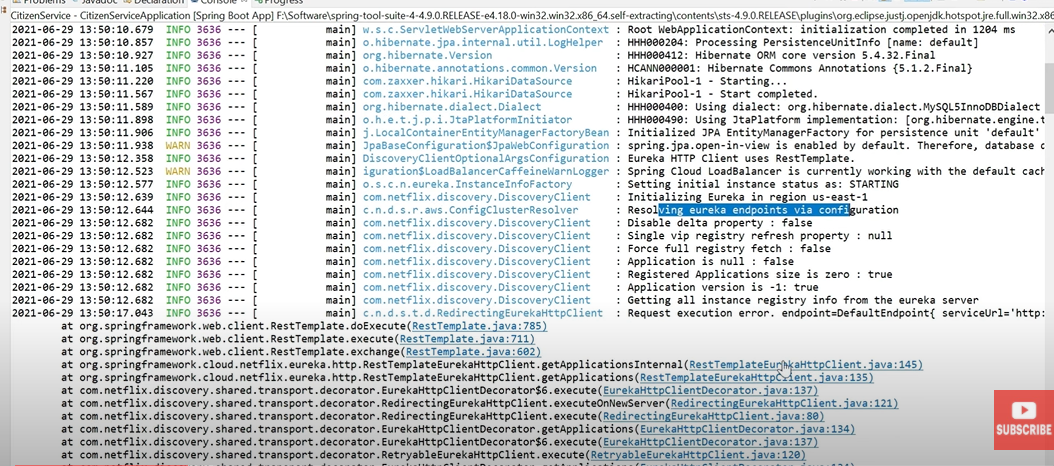


* Eureka Discovery Client starter dependency is required because it will help Eureka server to discover all the required services
* A REST based service for locating services for the purpose of load balancing and failover of middle-tier servers.

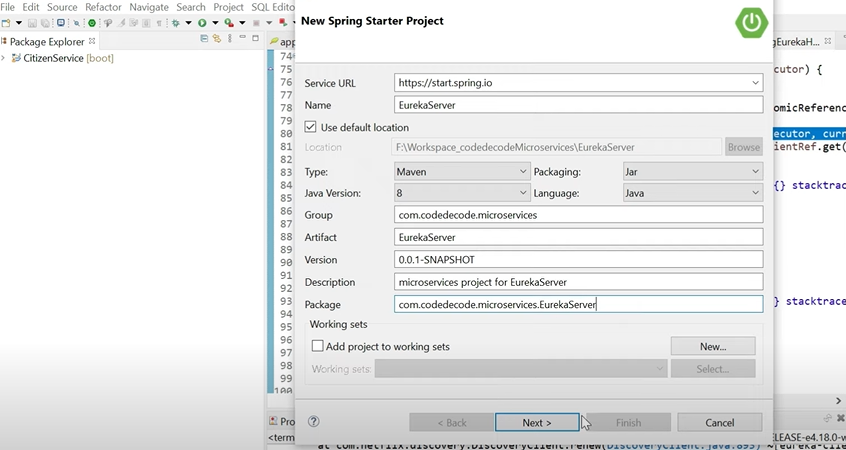


* We will create Yml file instead of properties file here.
* We are going to create one database for each micro-services and that will be the design pattern for our project i.e. One database for one micro-service and not shared database for all micro-services.



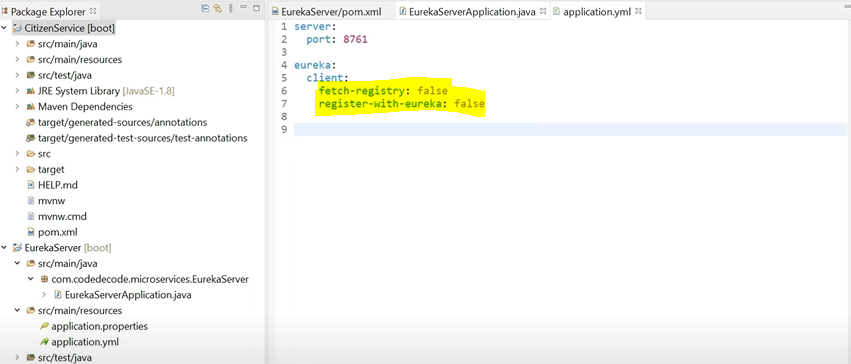


* We are getting errors as we have not yet configured the Eureka server.
* We will now create the Eureka Service
* Let’s create the project: **Eureka Service:**

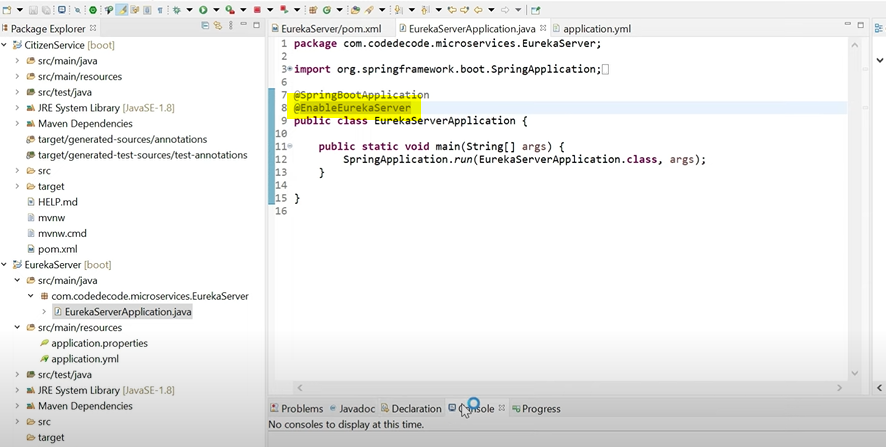




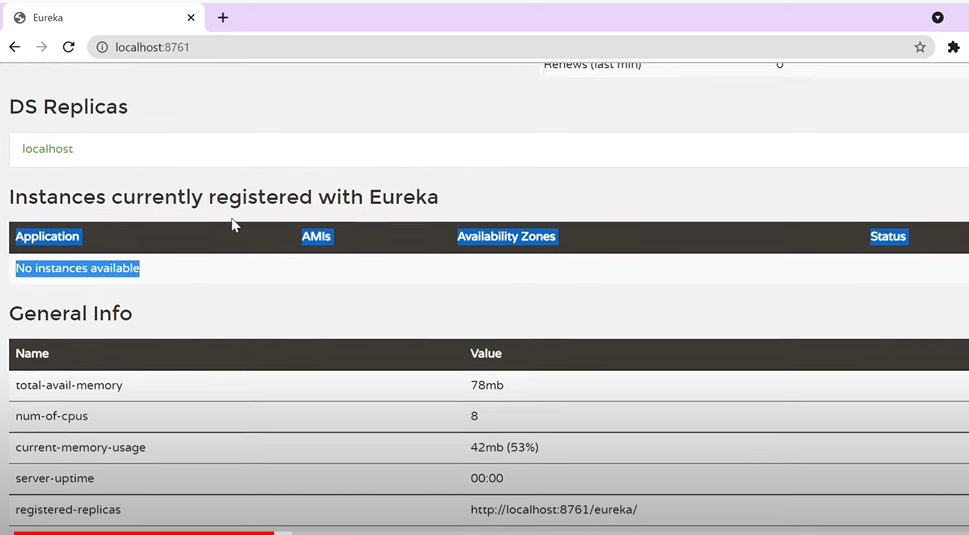
* We only need one dependency i.e. Eureka server which is from Netflix and nothing else to create the Eureka service for the service discoveries.
* We also need to create the Yml file for the port and other configurations.



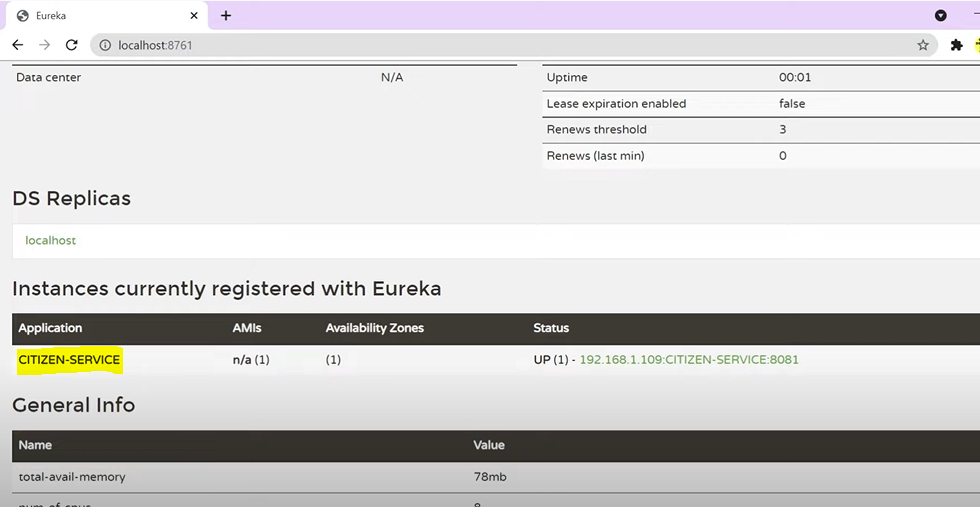
* The default server port for Eureka server is **8761**
* If we want to change it then we need to explicitly mention the updated port no. in all the corresponding services yml file.
* The Eureka server itself behaves as a client and it tries to fetch and register other Eureka service to stop this we need to put the above configuration.



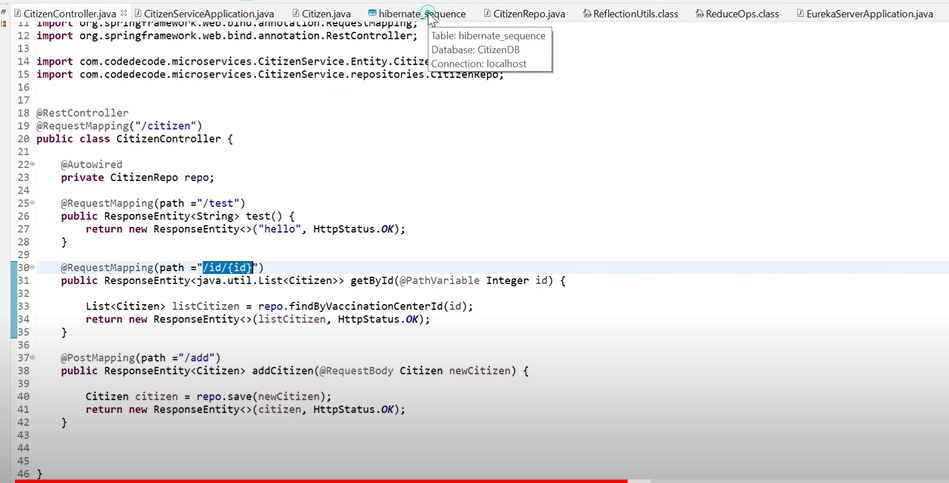
* We need to enable the Eureka server and we are good to go



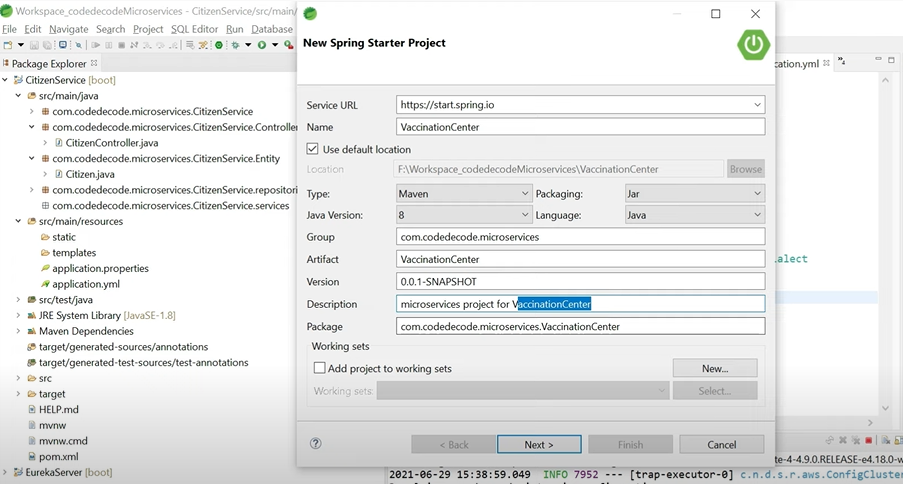
* We don’t need to put **@EnableEurekaServer** annotation on the other services as it is deprecated.

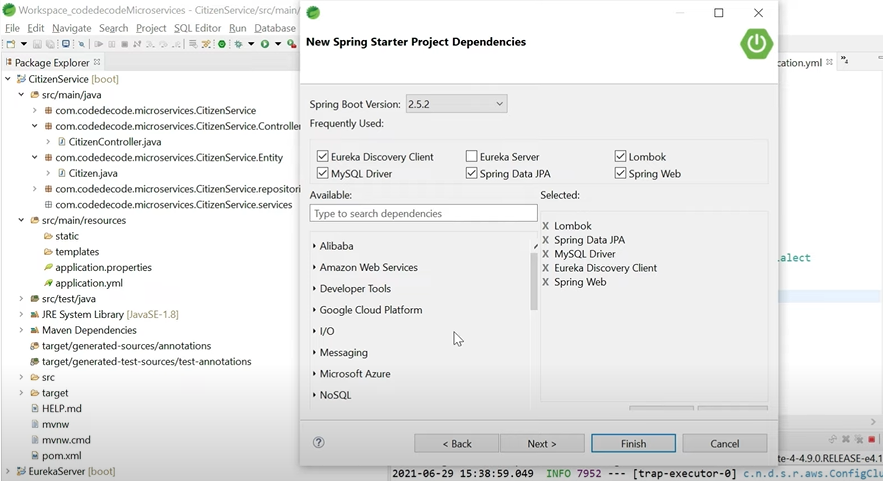


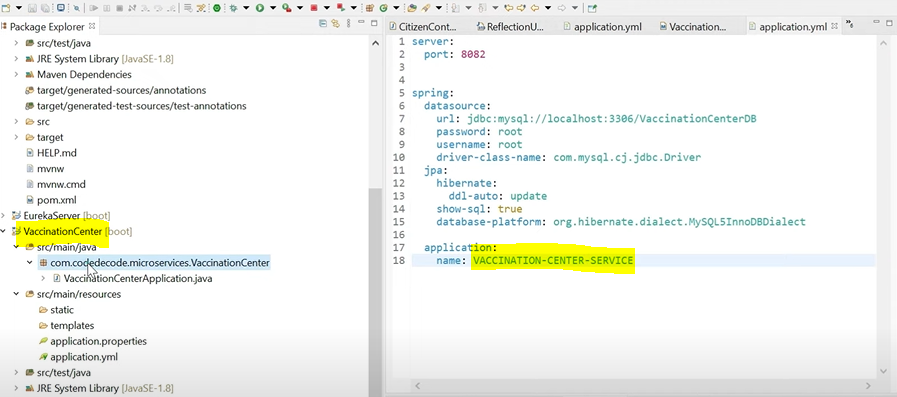
* In this way our one service is up and registered with Eureka server
* Now we will create the rest of the classes and interfaces for the services.

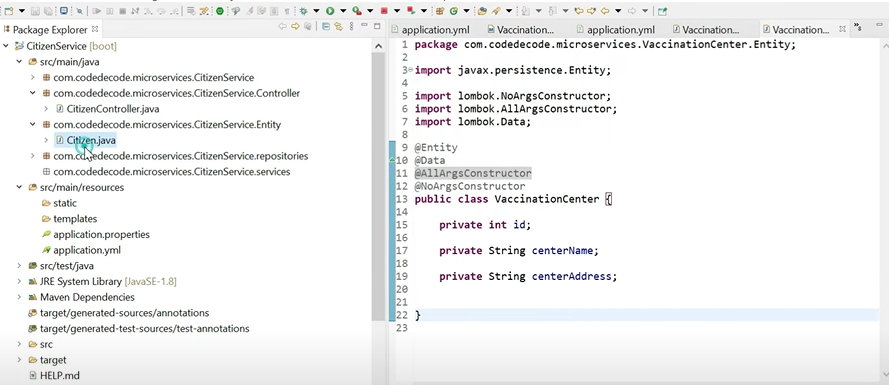


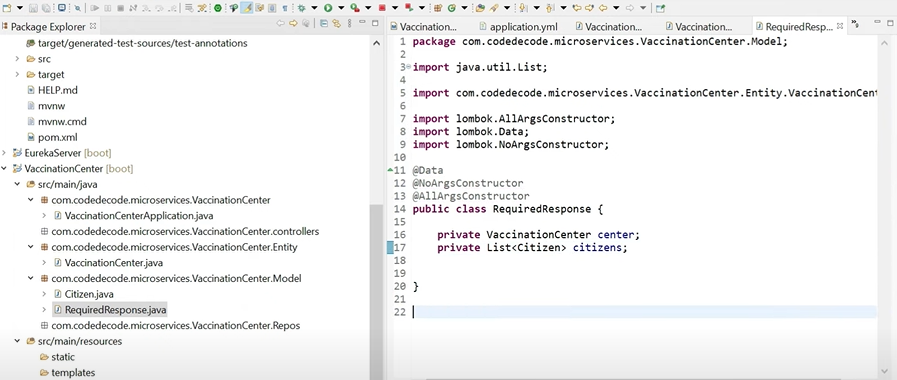
* Let’s create the project: **Vaccination Center Service:**

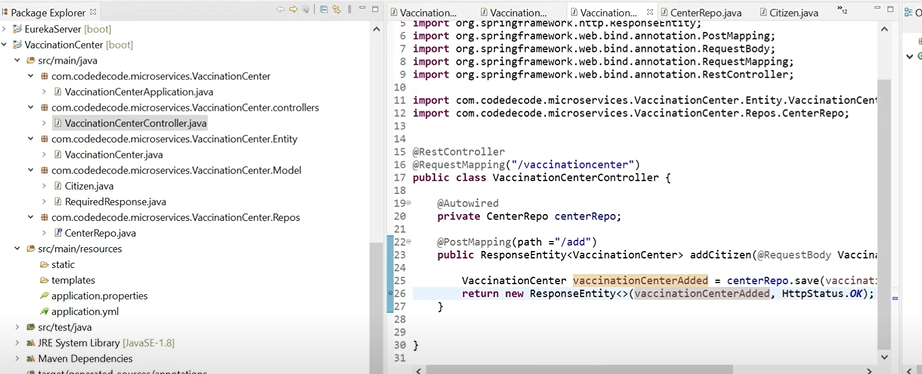


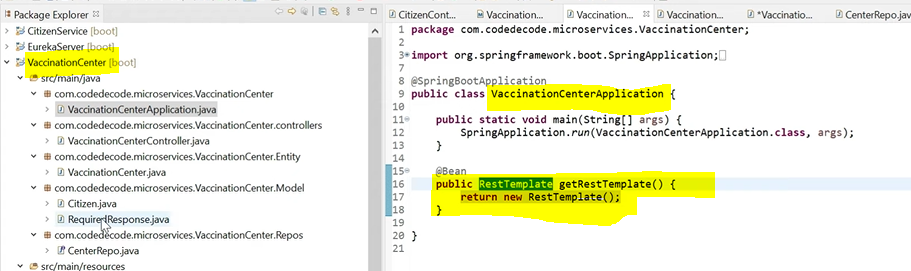


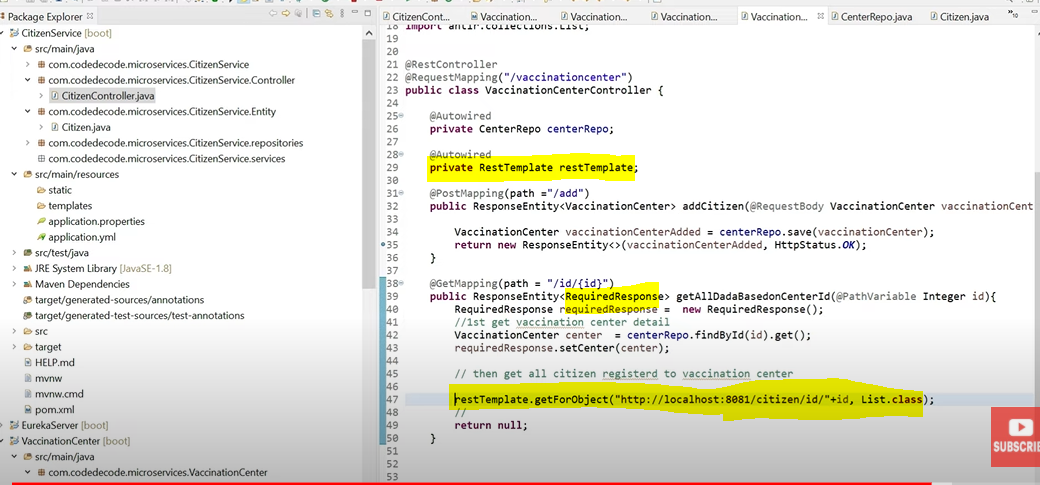




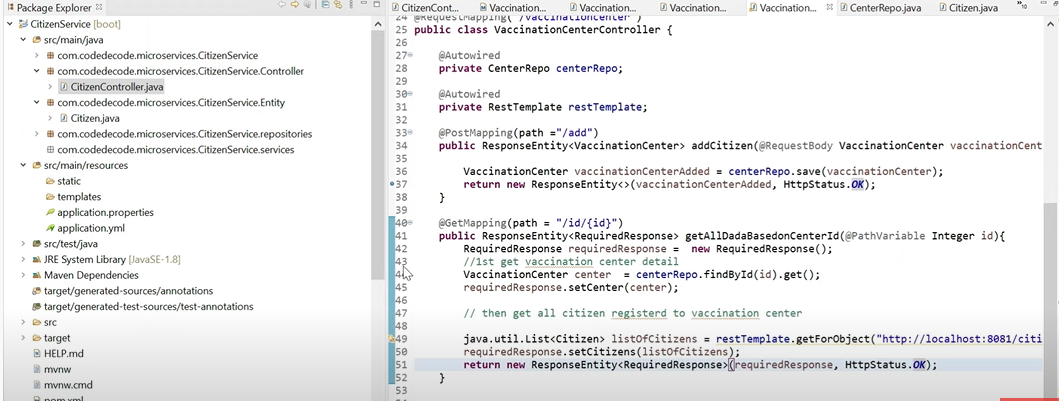








* We need to create RestTemplate to communicate with other service to get the details.



* In Place of the URL if we want Eureka to provide the required URL
* We need to provide the application name in place of local host and use @LoadBalanced annotation on the Rest Template as shown below.

