--------------------------------------------------------------------------------------------------------

MADARSA LITTLE FLOWAR JALALABAD  
Address-JALALABAD BIJNOR  
State-UTTAR PRADESH  
District-BIJNOR  
**Registered on NSP! - KYC status:- Approved**  
**Institute Id :**98889678062  
**AISHE/DISE Code-**09031001734  
**Registered Course Level**  
1. School I-X  
**Registered course** (PWD- 72DTU5BREN)

**Irctc - Iqbal1234**

C:\Users\768795\OneDrive - Cognizant\ **Attrib -h -s -r HIDDEN**

**EPFO UAN – 101467869224 Daha@1996**

101542869050

**DL application -682349821**

**GET JSON AS AN ARRAY--**

db.getCollection('collection').find({}).toArray()

**IMPORT JSON TO MONGODB BIN FOLDER using CMD---**

mongoimport --db eskaRvDB --collection TarrifOptions –file C:\nemo\_\sqlDump\mongoDb\TarrifOptions.json --jsonArray

**Command For Git Hub**

**git clone** [**git@github.com:ahadzafar08/MyProject4.git**](mailto:git@github.com:ahadzafar08/MyProject4.git)

**Web Client**–

1. Add WebClientFlux dependency in **POM**
2. Store accounts service url in **application.properties**
3. **Application-** 
   1. **@Value** (get the accounts service url)
   2. Create **@Bean** for WebClient

{*WebClient.builder().baseUrl(addressServiceUrl).build();* }

1. **Response –** create accountResponse and call it in CustomerResponse
2. **Service –** Autowire WebClient and create method to call **GET** accountService *{webClient.get().uri("/id/"+accountId).retrieve().bodyToMono(AccountResponse.class). block();}*

**OpenFeign**

Provides declarative method to make the REST call of the client replica we have created

It will provide requestMapping url , method names

1. Add spring cloud version and add ***cloud-openFeign*** to the **POM** file
2. Add annotation to ApplicationMain
   1. *@EnableFeignClients("com.infybuzz.cloud.feignclients")*
3. Create feignClient Interface
   1. *@FeignClient(url = "http://localhost:8080", value="account-feign-client" ,path="/api/account")*
   2. *public interface AddressFeignClient {*
4. Declare methods from accounts controller

**Eureka Server-**

Customer and Accounts service will register on EUREKA by service-name

*No need to mention the service url ,ONLY NAME.*

1. Create Eureka Server project
   1. In main class ***@EnableEurekaServer*** *and add eureka properties*
2. In **Student,Address** Pom add EurekaClient dependency
   1. In main class ***@EnableEurekaClient***
   2. And add eureka.url in properties
3. Add eureka service name to **FEIGNCLIENT** url.

**SpringCloudLoadBalancer –**