**For swaggerImplementation**

**SwaggerConfiguration.java**

@Configuration

@EnableSwagger2

**public** **class** SwaggerConfiguration {

@Bean

**public** Docket swaggerApiConfig() {

**return** **new** Docket(DocumentationType.***SWAGGER\_2***)

.select()

.paths(PathSelectors.*any*())

.apis(RequestHandlerSelectors.*basePackage*("com.example.execute.api.controller"))

.build();

}

}

OR

Note:put @EnableSwagger annotation in spring boot main call(nut it was not working in my case)

**Add Dependency**

<!-- https://mvnrepository.com/artifact/io.springfox/springfox-swagger2 -->

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-boot-starter</artifactId>

<version>3.0.0</version>

</dependency>

**InBrowser Goto this link**

<http://localhost:9091/swagger-ui/#/student-controller>

**MongoDb Details**

**Application.properties file**

server.port=9091

spring.data.mongodb.host=localhost

spring.data.mongodb.port=27017

spring.data.mongodb.database=studentdetails

**Student.java**

**@Document //@Entity for MySQL**

**public** **class** Student {

**@Id**

**private** String id;

**private** String name;

**private** Integer age;

**StudentRepository.java**

@Repository //JpaRepository for MySQL

**public** **interface** StudentRepository **extends** **MongoRepository**<Student, String>{

}

**Process**

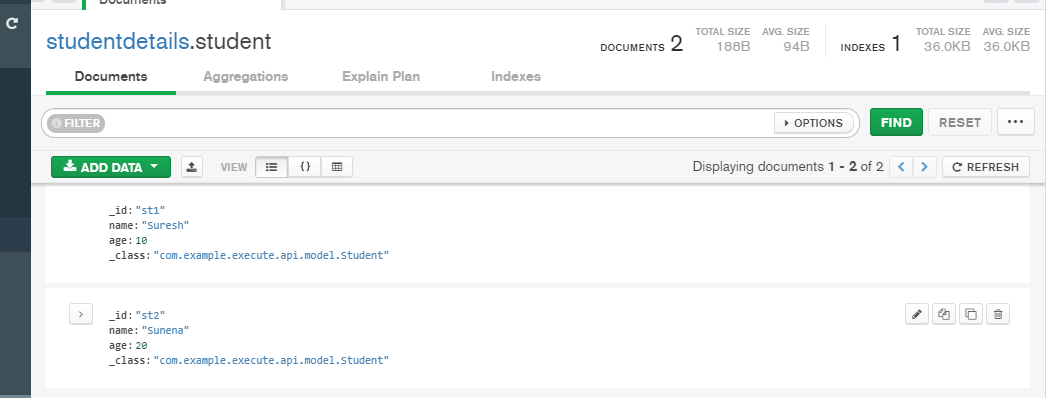
1. Goto Command terminal cd pathto bin of mongodb

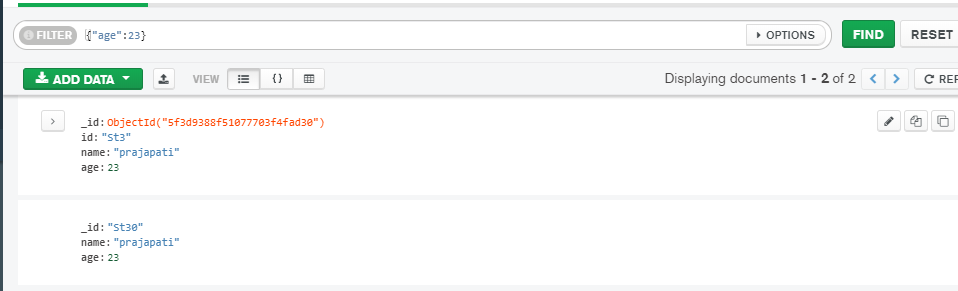
cd C:\Program Files\MongoDB\Server\4.2\bin

2. Start MongoDaemon by entering

Mongod hit enter

3. Then goto MongoDB Compass Community GUI in Desktop tool bar and connect

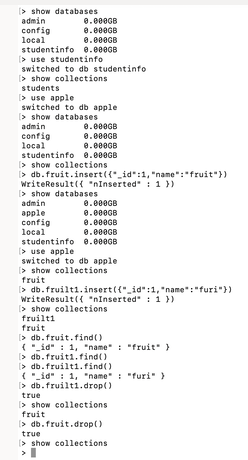




**MongoDb Command from Terminal**

**Show databases**

**Use databasename //create database if doesnot exits or select database if exists**



**Application.properties for MySQL**

server.port=9092

spring.application.name=ADDRESS-SERVICE

spring.datasource.url = jdbc:mysql://localhost:3306/address?**createDatabaseIfNotExist=true**

spring.datasource.username = root

spring.datasource.password = root

spring.datasource.platform=mysql

spring.jpa.hibernate.ddl-auto = create

spring.jpa.show-sql = true

**Primary key**

If Id is string use @Id, @Length(max=7)(it depends on spring version, may be not needed)

If Id is Integer @Id ,@GeneratedValue(strategry=GenerationType.AUTO)

**Harcoded OpemFeign**

**If Not using Eureka service Discovery**

@FeignClient(name="address-service", url = "http://localhost:9092/")

**public** **interface** MyFeignClient {

@GetMapping(value = "getAddress")

String getAddress();

}

**If using Eureka service Discovery**

**If Using Eureka Service Discovery**

@FeignClient(value = "B-SERVICE")

**public** **interface** MyFeignClientA {

@GetMapping(value = "getB")

String getB();

}