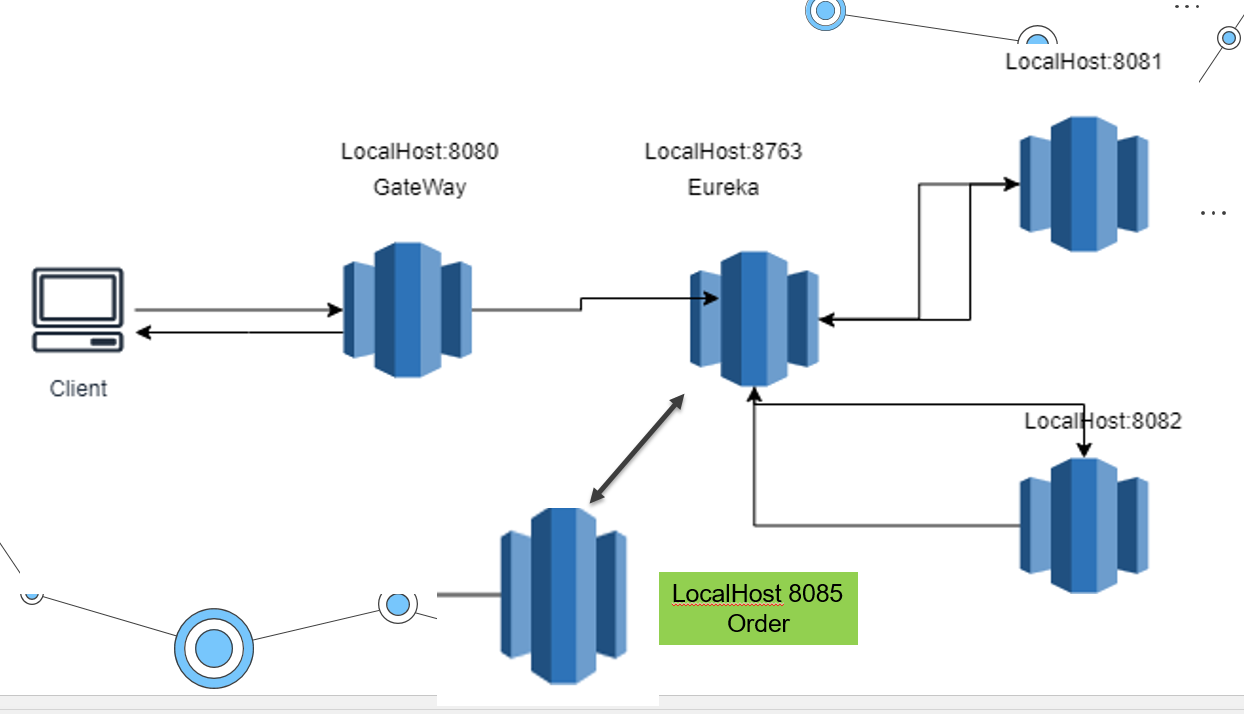
https://medium.com/design-microservices-architecture-with-patterns/microservices-communications-f319f8d76b71

1-рт өчигдрийн жишээ програм дээр 8085-р серверийг асааж Order буюу номнуудыг захиалгыг боловсруул.

Localhost 8083

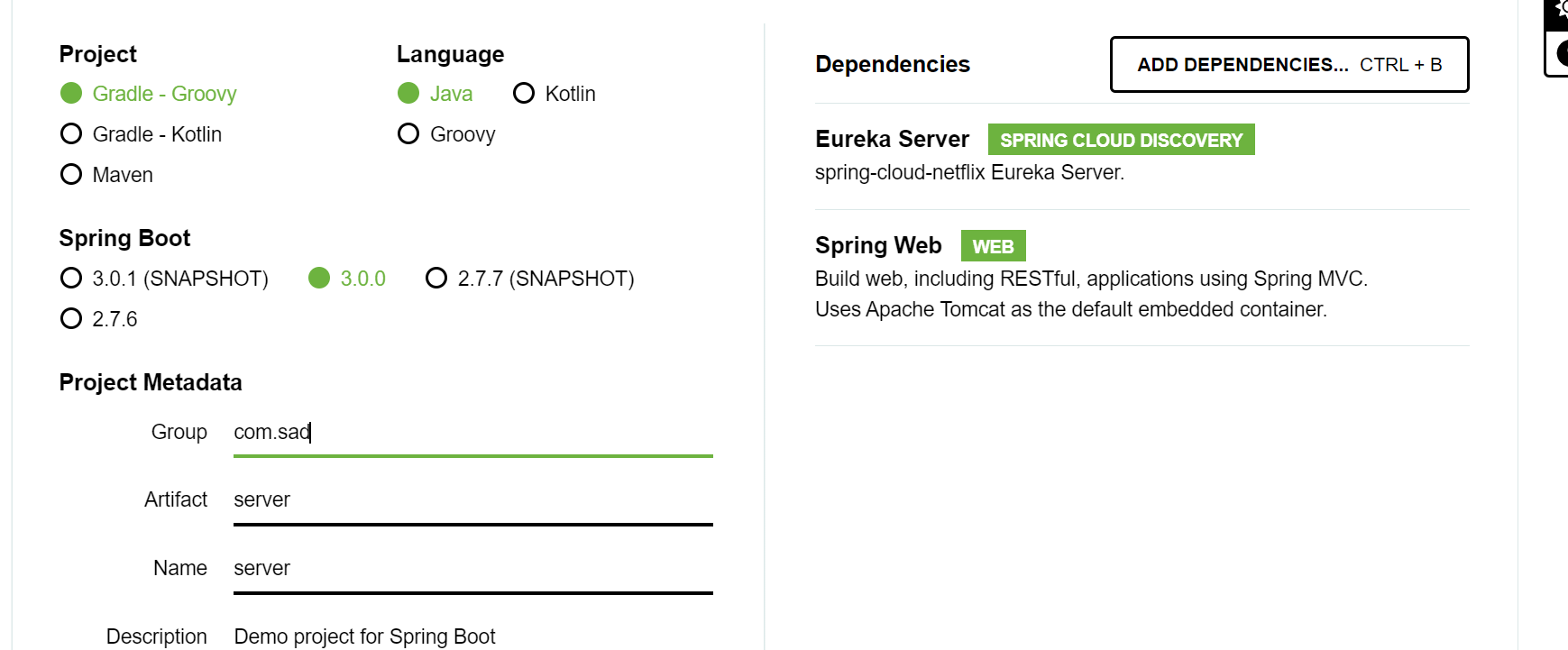
Localhost 8084

**Microservice architecture үүсгэх**

**1-р алхам**

Eureka project үүсгэх

<https://start.spring.io/> дээр maven project-г eureka server болон spring web –н dependence-уудтай гаар server нэртэй project үүсгэнэ.



File-Import командаар maven project-г оруулж ирнэ.

Application.properties файл дээр

spring.application.name="eureka-discovery-service"

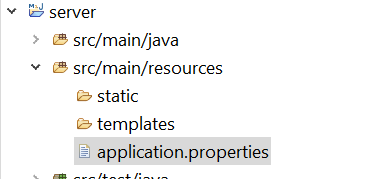
server.port = 8763

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

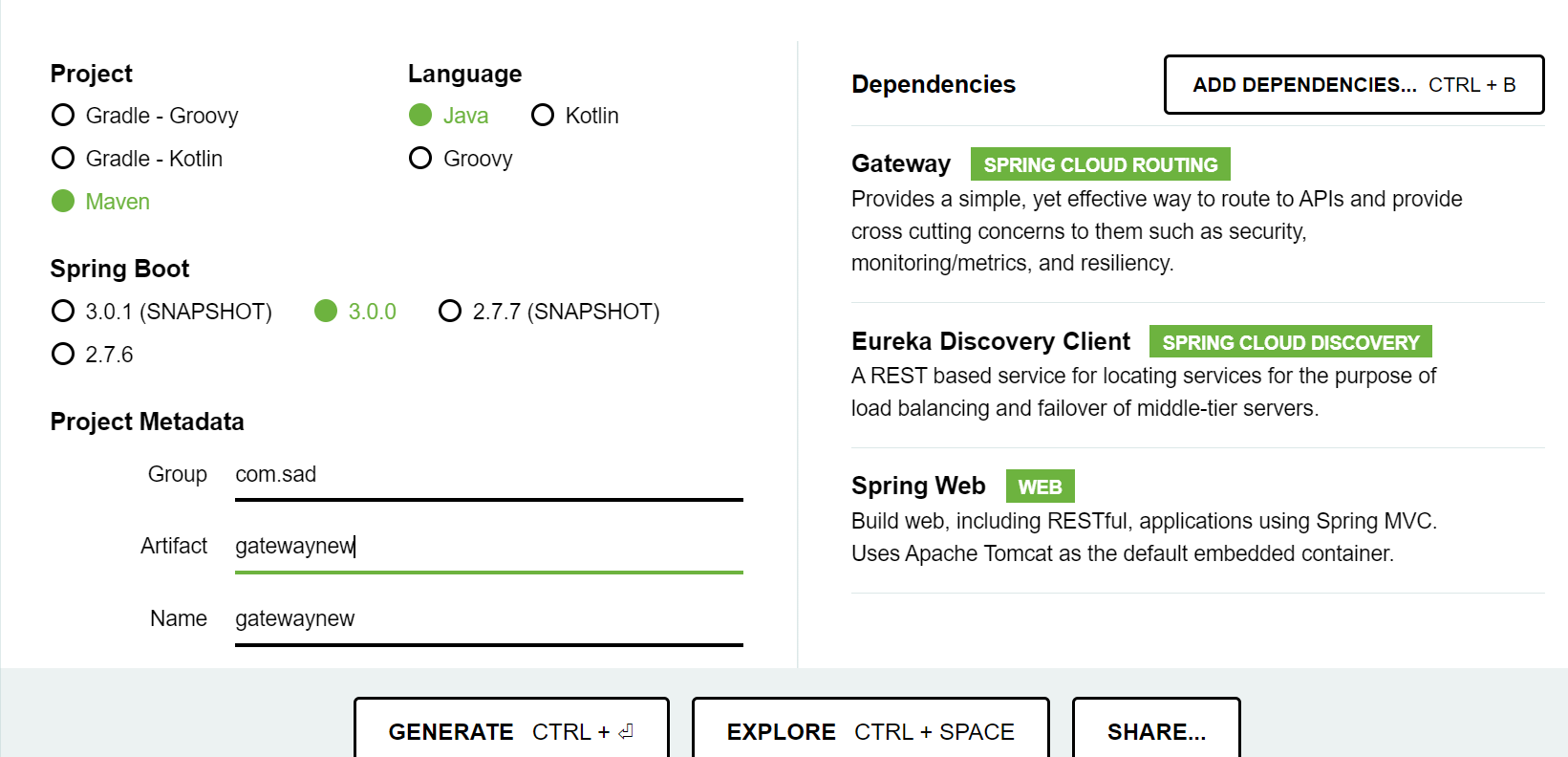
eureka.client.service-url.defaultZone=http://localhost:8763/eureka

Тохиргоог хуулж өгнө.



**2-р алхам Gateway project үүсгэх (миний жишээн дээр gatewaynew app)**

<https://start.spring.io/> дээр maven project-г eureka cloud discovery болон spring web –н dependence-уудтай гаар gateway үүрэг бүхий gateway нэртэй project үүсгэнэ.



File-import командаар maven project-г оруулж ирнэ.

src/main/resources каталог дотор application.yml файл үүсгэн дараах тохиргоог хийж өгнө. Үүнд server-н порт 8080 eureka server-н порт 8763 мөн bookModule, publisherModule, userModule, readerModule гэсэн 4 ширхэг микросервис арр байна.

server:

port: 8080

**Client хандах**

eureka:

**Дээрх eureka server**

client:

register-with-eureka: true

fetch-registry: true

serviceUrl:

defaultZone: http://localhost:8763/eureka/

spring:

application:

**bookModule**

name: gateway-service

cloud:

gateway:

**bookModule руу client-ээс хандах зам**

routes:

- id: bookModule

uri: http://localhost:8081/v1/book/

**book-н url-ууд**

predicates:

**publisherModule**

- Path=/v1/book/\*\*

**publisherModule руу client-ээс хандах зам**

- id: publisherModule

uri: http://localhost:8082/v1/publisher/

predicates:

- Path=/v1/publisher/\*\*

- id: userModule

**publisher-н url-ууд**

uri: http://localhost:8083/v1/user/

predicates:

- Path=/v1/user/\*\*

- id: readerModule

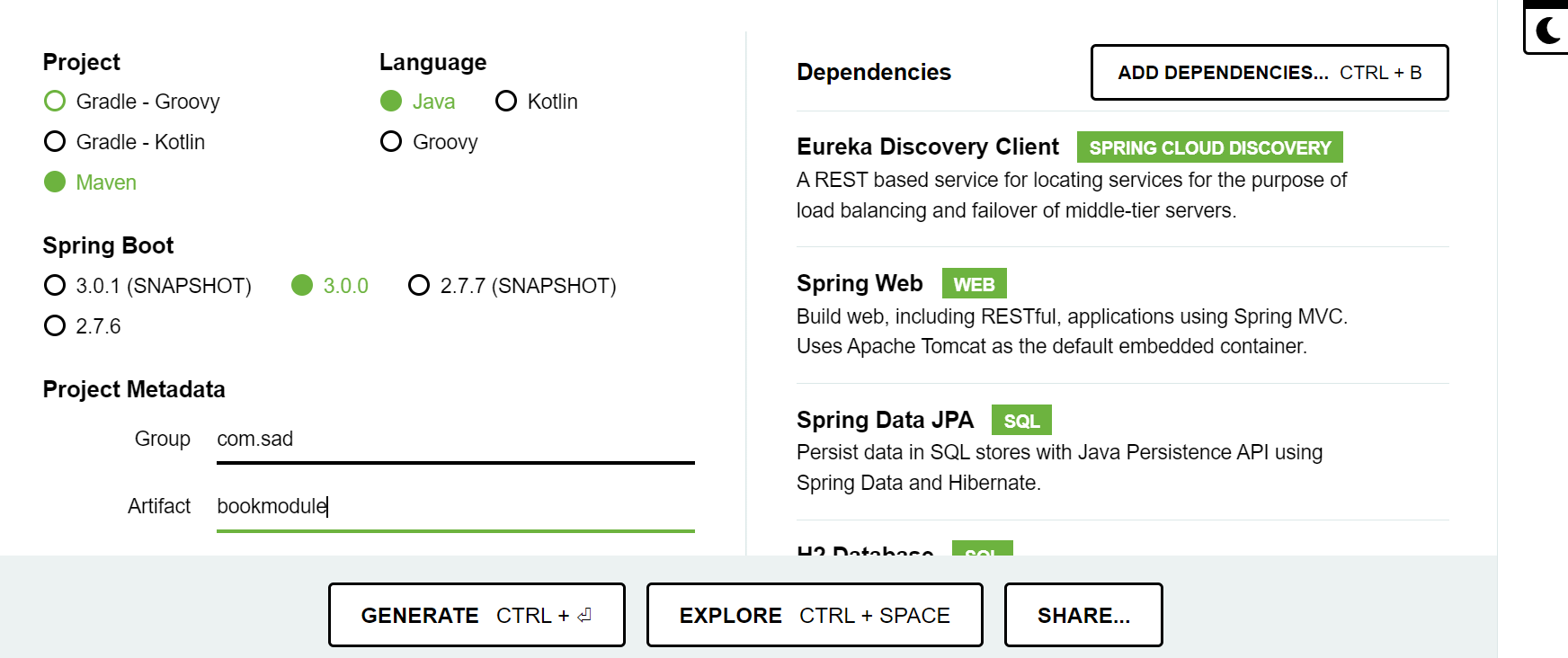
uri: http://localhost:8084/v1/reader/

predicates:

- Path=/v1/reader/\*\*

**3-р алхам bookModule микросервис үүсгэх**

<https://start.spring.io/> дээр maven project-г eureka cloud discovery болон spring web, jpa, h2 .. гэх мэт шаардлагатай dependence-уудтай гаар bookmodule нэртэй project үүсгэнэ.



1.Scr.main.java каталог дотор model пакеж үүсгээд Book.java класс ууүсгэнэ.

**public** **class** Book {

**private** **int** id;

**private** String bookName;

**private** **int** price;

**public** Book() {

}

**public** Book(**int** id, String bookName, **int** price) {

**this**.id = id;

**this**.bookName = bookName;

**this**.price = price;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getBookName() {

**return** bookName;

}

**public** **void** setBookName(String bookName) {

**this**.bookName = bookName;

}

**public** **int** getPrice() {

**return** price;

}

**public** **void** setPrice(**int** price) {

**this**.price = price;

}

}

2. Scr.main.java каталог дотор controller пакеж үүсгээд BookController.java класс ууүсгэнэ.

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.ResponseBody;

**import** org.springframework.web.bind.annotation.RestController;

**import** com.sad.bookmodule.model.Book;

Gateway app-д бүртгүүлсэн url

@RestController

@RequestMapping("v1/book")

**public** **class** BookController {

List<Book> books = **new** ArrayList<>();

@GetMapping("/books")

**public** List<Book> getBook()

{

books = Arrays.*asList*(**new** Book(1, "Уран зохиол", 10000),

**new** Book(2, "Шилдэг зохиолууд", 12000),

**new** Book(3, "Монголын нууц товчоо", 24000),

**new** Book(4, "Нарны өнгө", 6000),

**new** Book(5, "Гэмт хэргийн дууль", 23000));

**return** books;

}

@GetMapping("/book/{id}")

@ResponseBody

**public** Book getByidBook(@PathVariable String id)

{

books = getBook();

Book book = books.stream()

.filter(f-> f.getId() == Integer.*parseInt*(id))

.findAny()

.orElse(**null**);

**return** book;

}

}3. application.properties дээр доорх тохиргоог хийж өгнө.

spring.application.name="bookmodule"

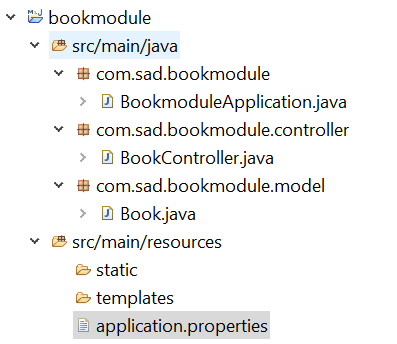
server.port=8081

eureka.client.register-with-eureka=true

eureka.client.fetch-registry=true

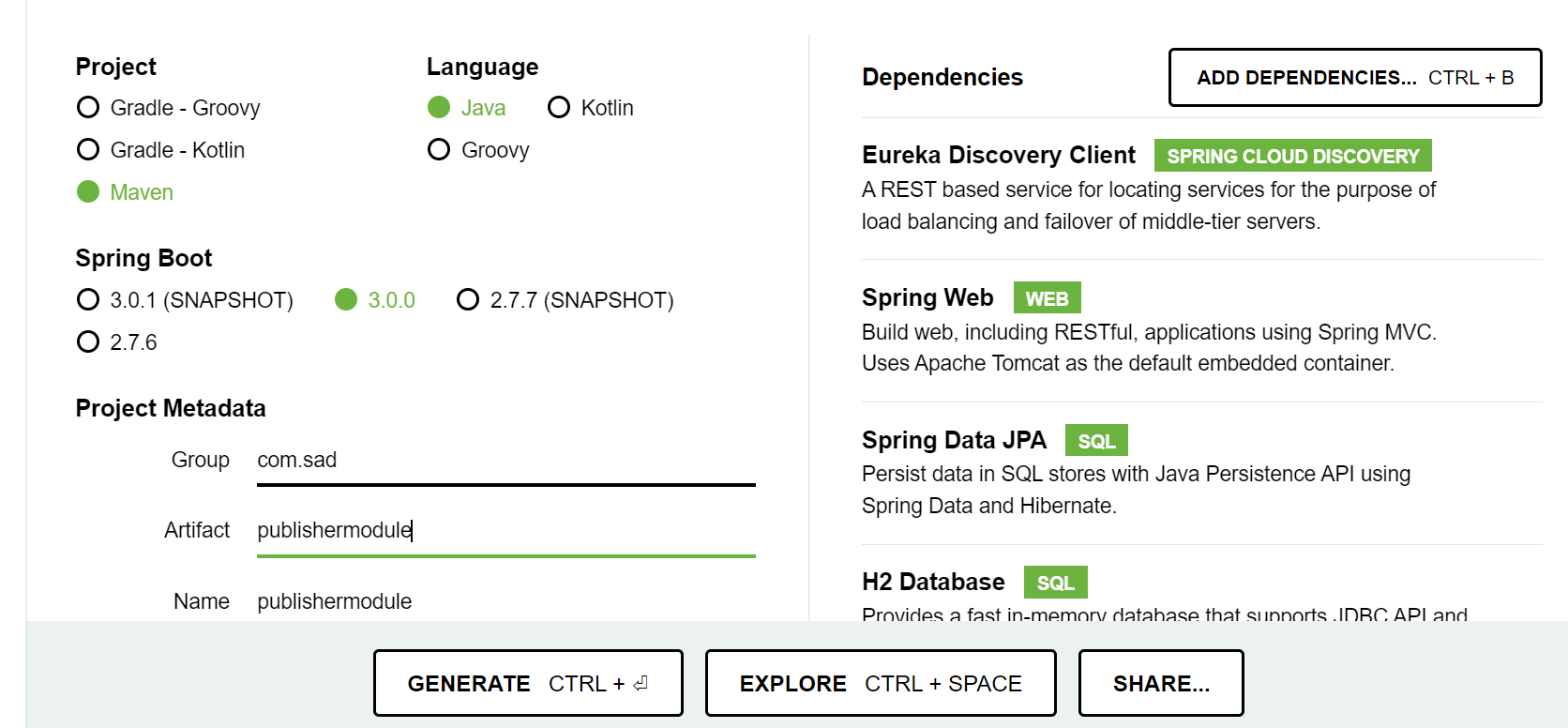
eureka.client.serviceUrl.defaultZone=http://localhost:8763/eureka/

instance.hostname=localhost



**4-р алхам publisherModule микросервис үүсгэх**

<https://start.spring.io/> дээр maven project-г eureka cloud discovery болон spring web, jpa, h2 .. гэх мэт шаардлагатай dependence-уудтай гаар publishermodule нэртэй project үүсгэнэ.



**1.Scr.main.java каталог дотор model пакеж үүсгээд Publisher.java класс ууүсгэнэ.**

**public** **class** Publisher {

**private** **long** id;

**private** String publisherName;

**private** String address;

**private** String phoneNumber;

**public** Publisher() {

}

**public** Publisher(**long** id, String publisherName, String address, String phoneNumber)

{

**this**.id = id;

**this**.publisherName = publisherName;

**this**.address = address;

**this**.phoneNumber = phoneNumber;

}

**public** **long** getId() {

**return** id;

}

**public** **void** setId(**long** id) {

**this**.id = id;

}

**public** String getPublisherName() {

**return** publisherName;

}

**public** **void** setPublisherName(String publisherName) {

**this**.publisherName = publisherName;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** String getPhoneNumber() {

**return** phoneNumber;

}

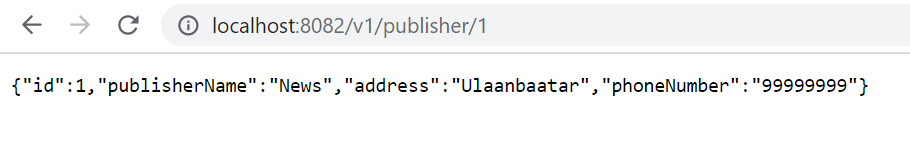
**public** **void** setPhoneNumber(String phoneNumber) {

**this**.phoneNumber = phoneNumber;

}

}

**2. Scr.main.java каталог дотор controller пакеж үүсгээд PublisherController.java класс ууүсгэнэ.**



@RestController

Gateway app-д бүртгүүлсэн url

@RequestMapping("v1/publisher")

**public** **class** PublisherController {

@GetMapping("/{publisherId}")

**public** Publisher getPublisher()

{

**return** **new** Publisher(1,"News", "Ulaanbaatar", "99999999");

}

}

**3. application.properties дээр доорх тохиргоог хийж өгнө.**

spring.application.name="publisherModule"

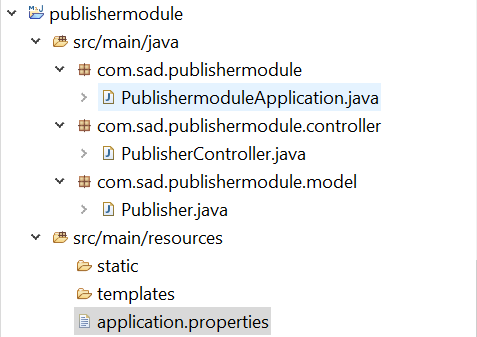
server.port=8082

eureka.client.register-with-eureka=true

eureka.client.fetch-registry=true

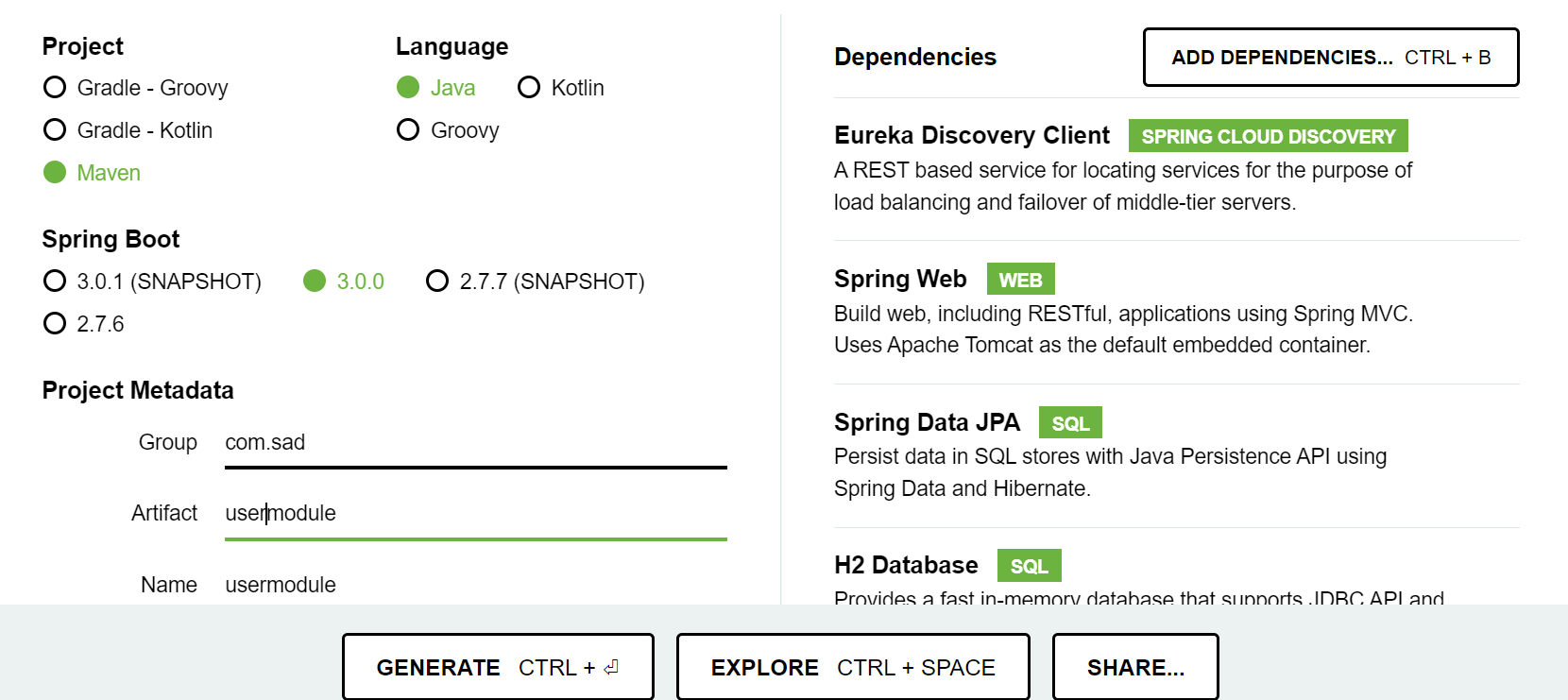
eureka.client.serviceUrl.defaultZone=http://localhost:8763/eureka/

instance.hostname=localhost



**5-р алхам userModule микросервис үүсгэх**

<https://start.spring.io/> дээр maven project-г eureka cloud discovery болон spring web, jpa, h2 .. гэх мэт шаардлагатай dependence-уудтай гаар usermodule нэртэй project үүсгэнэ.



**1.Scr.main.java каталог дотор model пакеж үүсгээд User.java класс ууүсгэнэ.**

**public** **class** User {

**private** **int** id;

**private** String UserName;

**public** User() {

}

**public** User(**int** id, String userName) {

**this**.id = id;

UserName = userName;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getUserName() {

**return** UserName;

}

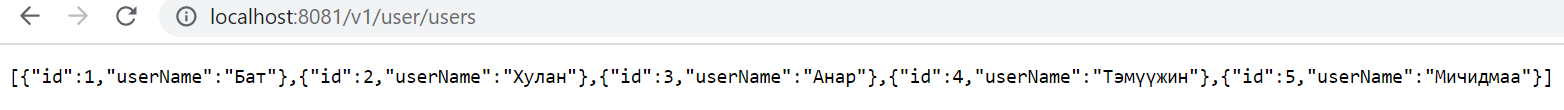
**public** **void** setUserName(String userName) {

UserName = userName;

}

}

**2. Scr.main.java каталог дотор controller пакеж үүсгээд UserController.java класс ууүсгэнэ.**



Gateway app-д бүртгүүлсэн url

@RestController

@RequestMapping("v1/user")

**public** **class** UserController {

List<User> users =**new** ArrayList<>();

@GetMapping("/users")

**public** List<User> getUsers()

{

users = Arrays.*asList*(

**new** User(1, "Бат"),

**new** User(2, "Хулан"),

**new** User(3, "Анар"),

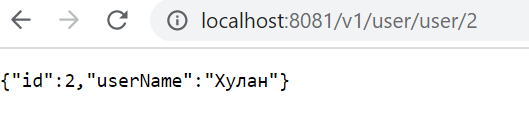
**new** User(4, "Тэмүүжин"),

**new** User(5, "Мичидмаа")

);

**return** users;

}



@GetMapping("/user/{id}")

@ResponseBody

**public** User getUser(@PathVariable String id)

{ users = getUsers();

User user = users.stream()

.filter(f-> f.getId() == Integer.*parseInt*(id))

.findAny()

.orElse(**null**);

**return** user;

}

}

**3. application.properties дээр доорх тохиргоог хийж өгнө**.

spring.application.name="userModule"

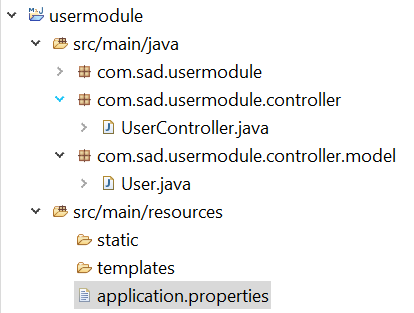
server.port=8083

eureka.client.register-with-eureka=true

eureka.client.fetch-registry=true

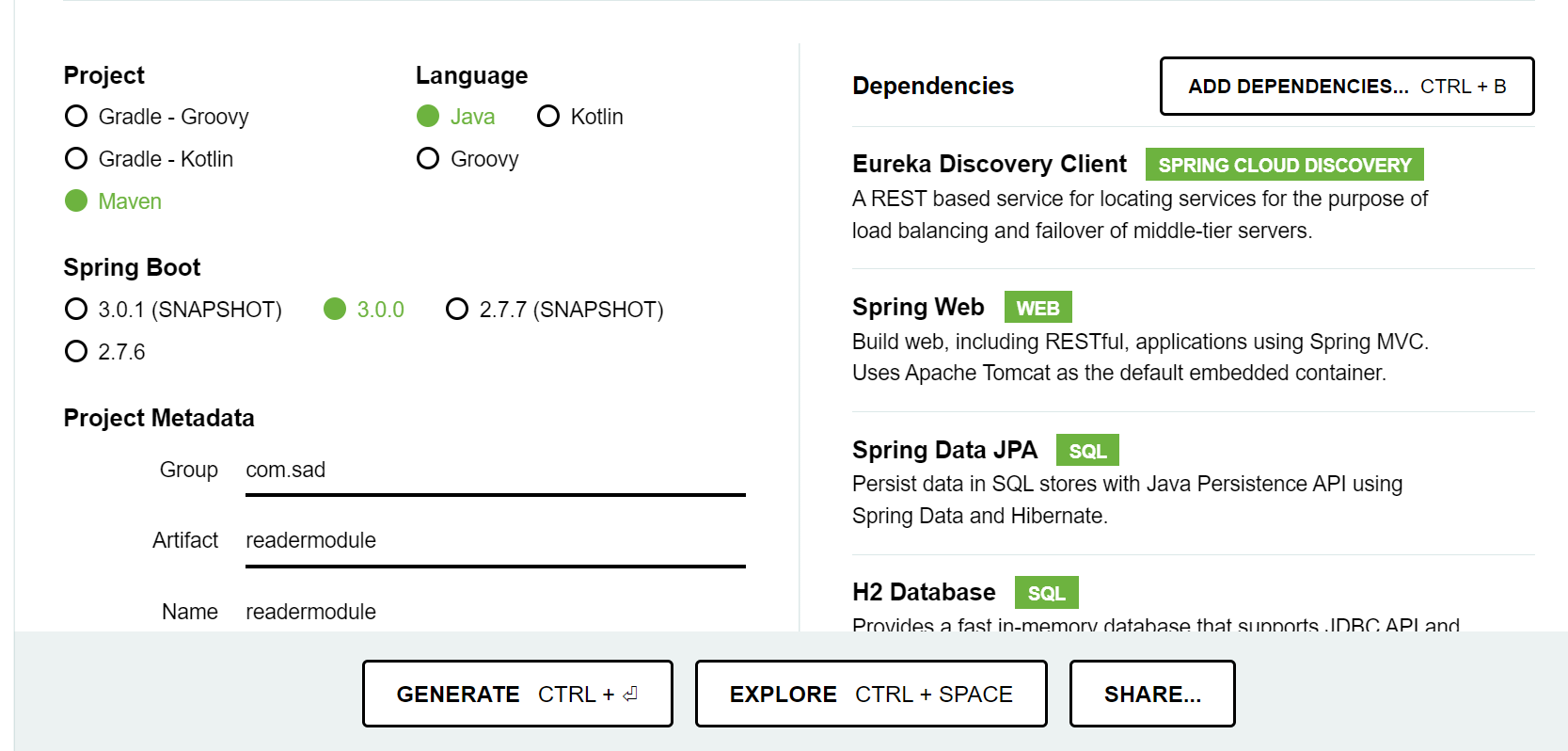
eureka.client.serviceUrl.defaultZone=http://localhost:8763/eureka/

instance.hostname=localhost



**6-р алхам readerModule микросервис үүсгэх**

<https://start.spring.io/> дээр maven project-г eureka cloud discovery болон spring web, jpa, h2 .. гэх мэт шаардлагатай dependence-уудтай гаар readermodule нэртэй project үүсгэнэ.



1.Scr.main.java каталог дотор model пакеж үүсгээд Reader.java болон Book.java класс-уудыг ууүсгэнэ.

**public** **class** Reader {

**private** **int** id;

**private** String name;

**private** List<Book> booksList = **new** ArrayList<>();

**public** Reader(){

}

**public** Reader(**int** id, String name){

**this**.id = id;

**this**.name = name;

}

**public** Reader(**int** id, String name, String booksRead){

**this**.id = id;

**this**.name = name;

}

**public** **void** addBook(Book book){

booksList.add(book);

}

**public** List<Book> getBooksList() {

**return** booksList;

}

**public** **void** setBooksList(List<Book> booksList) {

**this**.booksList = booksList;

}

**public** **int** getId() {

**return** id;

}

**public** String getName() {

**return** name;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

**public** **class** Book {

**private** **int** id;

**private** String bookName;

**public** Book(**int** id, String bookName){

**this**.bookName = bookName;

**this**.id = id;

}

**public** Book(){

}

**public** **int** getId() {

**return** id;

}

**public** String getBookName() {

**return** bookName;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

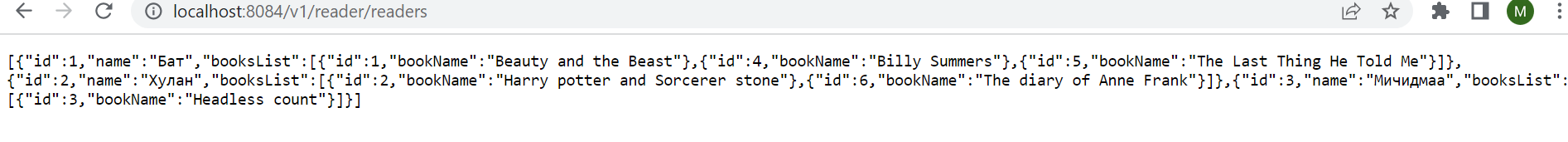
**public** **void** setBookName(String bookName) {

**this**.bookName = bookName;

}

}

2. Scr.main.java каталог дотор controller пакеж үүсгээд ReaderController.java класс ууүсгэнэ.



Gateway app-д бүртгүүлсэн url

@RestController

@RequestMapping("v1/reader")

**public** **class** ReaderController {

@GetMapping("/readers")

**public** List<Reader> getReaders()

{

List<Reader> readers = **new** ArrayList<>();

Reader reader1 = **new** Reader(1, "Бат");

reader1.addBook(**new** Book(1, "Beauty and the Beast"));

reader1.addBook(**new** Book(4, "Billy Summers"));

reader1.addBook(**new** Book(5, "The Last Thing He Told Me"));

Reader reader2 = **new** Reader(2, "Хулан");

reader2.addBook(**new** Book(2, "Harry potter and Sorcerer stone"));

reader2.addBook(**new** Book(6, "The diary of Anne Frank"));

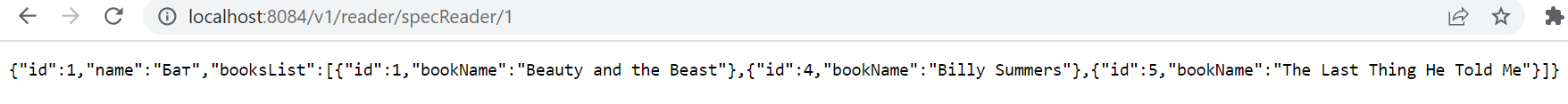
Reader reader3 = **new** Reader(3, "Мичидмаа");

reader3.addBook(**new** Book(3, "Headless count"));

readers = Arrays.*asList*(reader1, reader2, reader3);

**return** readers;

}



@GetMapping("/specReader/{id}")

@ResponseBody

**public** Reader getUser(@PathVariable String id)

{ List<Reader> readers = **new** ArrayList<>();

readers =getReaders();

Reader reader = readers.stream()

.filter(f-> f.getId() == Integer.*parseInt*(id))

.findAny()

.orElse(**null**);

**return** reader;

}

}

3. application.properties дээр доорх тохиргоог хийж өгнө.

spring.application.name="readerModule"

server.port=8084

eureka.client.register-with-eureka=true

eureka.client.fetch-registry=true

eureka.client.serviceUrl.defaultZone=http://localhost:8763/eureka/

instance.hostname=localhost

