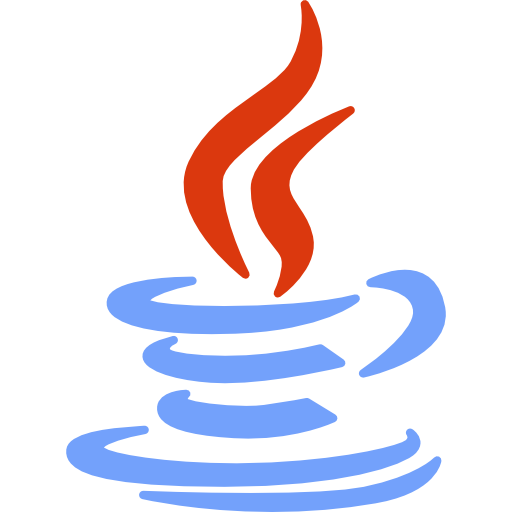
**ĐÀO TẠO JAVA SPRING BOOT**





**MỤC LỤC**

[I. TỔNG QUAN 7](#_Toc107241424)

[I.1. Mục đích 7](#_Toc107241425)

[I.2. Phạm vi 7](#_Toc107241426)

[I.3. Tài liệu liên quan 8](#_Toc107241427)

[I.4. Giải thích từ ngữ và các chữ viết tắt 8](#_Toc107241428)

[II. NỘI DUNG 9](#_Toc107241429)

[II.1. TỔNG QUAN HỆ THỐNG 9](#_Toc107241430)

[II.1.1. Phát biểu bài toán 9](#_Toc107241431)

[II.1.2. Mục tiêu và phạm vi hệ thống 9](#_Toc107241432)

[II.1.3. Mô hình hệ thống 9](#_Toc107241433)

[II.2. ĐẶC TẢ YÊU CẦU CHỨC NĂNG HỆ THỐNG 12](#_Toc107241434)

[II.2.1. Phân hệ Quản trị hệ thống 12](#_Toc107241435)

[II.2.1.1. Quản trị người dùng 12](#_Toc107241436)

[II.2.1.2. Quản trị quyền người dùng 14](#_Toc107241437)

[II.2.1.3. Đồng bộ và phân cấp 15](#_Toc107241438)

[II.2.1.4. Quản trị menu 16](#_Toc107241439)

[II.2.1.5. Quản trị nhóm quyền 17](#_Toc107241440)

[II.2.2. Phân hệ Trồng trọt và bảo vệ thực vật 18](#_Toc107241441)

[II.2.2.1. Cấu hình danh mục 19](#_Toc107241442)

[II.2.2.2. Tình hình sản xuất 26](#_Toc107241443)

[II.2.2.3. Tình hình sâu bệnh 27](#_Toc107241444)

[II.2.2.4. Cơ sở sản xuất kinh doanh cấp mới 29](#_Toc107241445)

[II.2.2.5. Xếp loại cơ sở trồng trọt 30](#_Toc107241446)

[II.2.2.6. Mã vùng trồng trọt 31](#_Toc107241447)

[II.2.2.7. Cấp chứng nhận 32](#_Toc107241448)

[II.2.2.8. Chính sách hỗ trợ 33](#_Toc107241449)

[II.2.2.9. Vườn cây đầu dòng 34](#_Toc107241450)

[II.2.2.10. Tiêu chuẩn, quy chuẩn kỹ thuật 35](#_Toc107241451)

[II.2.2.11. Liên kết sản xuất 36](#_Toc107241452)

[II.2.3. Phân hệ Chăn nuôi và thú y 37](#_Toc107241453)

[II.2.3.1. Cấu hình danh mục 37](#_Toc107241454)

[II.2.3.2. Quản lý cơ sở sản xuất kinh doanh con giống 44](#_Toc107241455)

[II.2.3.3. Quản lý danh sách cá nhân làm dịch vụ thụ tinh nhân tạo 46](#_Toc107241456)

[II.2.3.4. Quản lý nhà yến 47](#_Toc107241457)

[II.2.3.5. Quản lý trang trại chăn nuôi 48](#_Toc107241458)

[II.2.3.6. Quản lý Số lượng vật nuôi trên địa bàn 49](#_Toc107241459)

[II.2.3.7. Hổ trợ tiêu hủy gia súc / gia cầm 50](#_Toc107241460)

[II.2.3.8. Quản lý bệnh trên gia súc 51](#_Toc107241461)

[II.2.3.9. Quản lý bệnh trên gia cầm 52](#_Toc107241462)

[II.2.3.10. Cơ sở an toàn dịch bệnh 53](#_Toc107241463)

[II.2.3.11. Quản lý cơ sở ấp trứng gia cầm 54](#_Toc107241464)

[II.2.3.12. Quản lý cơ sở kho lạnh bảo quản thực phẩm gia súc, gia cầm 55](#_Toc107241465)

[II.2.3.13. Quản lý cơ sở sản xuất thức ăn chăn nuôi thương mại 57](#_Toc107241466)

[II.2.3.14. Quản lý cơ sở sản xuất nguyên liệu thức ăn chăn nuôi / thức ăn truyền thống 58](#_Toc107241467)

[II.2.3.15. Quản lý cá nhân được cấp chứng chỉ hành nghề thú y 60](#_Toc107241468)

[II.2.3.16. Quản lý cơ sở kinh doanh thuốc thú y 61](#_Toc107241469)

[II.2.4. Phân hệ Thủy sản 64](#_Toc107241470)

[II.2.4.1. Cấu hình danh mục 64](#_Toc107241471)

[II.2.4.2. Quản lý đăng ký tàu cá 76](#_Toc107241472)

[II.2.4.3. Quản lý đồng sở hữu 80](#_Toc107241473)

[II.2.4.4. Quản lý Giấy phép khai thác 81](#_Toc107241474)

[II.2.4.5. Quản lý vi phạm hành chính 83](#_Toc107241475)

[II.2.4.6. Quản lý sang tên trong tỉnh 84](#_Toc107241476)

[II.2.4.7. Kiểm tra an toàn thực phẩm trên tàu 87](#_Toc107241477)

[II.2.4.8. Cơ sở sản xuất, ương dưỡng giống thủy sản 89](#_Toc107241478)

[II.2.4.9. Nuôi bè 91](#_Toc107241479)

[II.2.4.10. Nuôi tôm 92](#_Toc107241480)

[II.2.4.11. Nuôi cá tra 93](#_Toc107241481)

[II.2.4.12. Nuôi khác 95](#_Toc107241482)

[II.2.4.13. Cơ sở sản xuất thức ăn, sản phẩm xử lý môi trường nuôi trồng thủy sản 96](#_Toc107241483)

[II.2.4.14. Cơ sở kinh doanh thức ăn, sản phẩm xử lý môi trường nuôi trồng thủy sản 97](#_Toc107241484)

[II.2.4.15. Theo dõi tình hình nuôi các đối tượng chính 99](#_Toc107241485)

[II.2.5. Phân hệ Phát triển nông thôn 101](#_Toc107241486)

[II.2.5.1. Quản trị dự án giảm nghèo 101](#_Toc107241487)

[II.2.5.2. Quản trị loại hình hoạt động 103](#_Toc107241488)

[II.2.6. Phân hệ Quản lý chất lượng Nông lâm sản và Thủy sản 132](#_Toc107241489)

[II.2.6.1. Doanh nghiệp 132](#_Toc107241490)

[II.2.6.2. Cơ sở 133](#_Toc107241491)

[II.2.6.3. Chứng nhận an toàn thực phẩm 135](#_Toc107241492)

[II.2.6.4. Đánh giá phân loại doanh nghiệp 137](#_Toc107241493)

[II.2.6.5. Kết quả lấy mẫu giám sát an toàn thực phẩm sản phẩm nông lâm thủy sản 139](#_Toc107241494)

[II.2.6.6. Nội dung chuỗi cung cấp thực phẩm nông lâm thủy sản an toàn 141](#_Toc107241495)

[II.2.6.7. Mô hình trình diễn kỹ thuật, đề tài, dự án 142](#_Toc107241496)

[II.2.6.8. Danh sách cơ sở được chứng nhận VietGAP 143](#_Toc107241497)

[Giấy chứng nhận ATTP 144](#_Toc107241498)

[II.2.7. Phân hệ Thủy lợi 146](#_Toc107241499)

[II.2.7.1. Quản lý danh mục 146](#_Toc107241500)

[II.2.7.2. Công trình cống 147](#_Toc107241501)

[II.2.7.3. Công trình kênh 148](#_Toc107241502)

[II.2.7.4. Công trình trạm bơm 149](#_Toc107241503)

[II.2.7.5. Công trình tuyến đê 150](#_Toc107241504)

[II.2.7.6. Quản lý nước sinh hoạt 151](#_Toc107241505)

[I.1.1.2. Quản lý thiên tai 153](#_Toc107241506)

[I.1.1.3. Quản lý vi phạm 155](#_Toc107241507)

[I.1.1.4. Quản lý nước mưa, mặn, lũ 156](#_Toc107241508)

[II.2.8. Phân hệ Lâm nghiệp 160](#_Toc107241509)

[II.2.8.1. Cấu hình danh mục 161](#_Toc107241510)

[II.2.8.2. Cây phân tán 164](#_Toc107241511)

[II.2.8.3. Khai thác gỗ và lâm sản 165](#_Toc107241512)

[II.2.8.4. Xử lý vi phạm 166](#_Toc107241513)

[II.2.8.5. Cơ sở chăn nuôi động vật hoang dã 168](#_Toc107241514)

[II.2.8.6. Cơ sở chế biến kinh doanh gỗ 169](#_Toc107241515)

[II.2.8.7. Theo dõi diễn biến rừng 170](#_Toc107241516)

[II.2.9. Phân hệ Trung tâm khuyến nông và giống cây trồng 171](#_Toc107241517)

[II.2.9.1. Cấu hình danh mục 171](#_Toc107241518)

[II.2.9.2. Tập huấn – chuyển giao 172](#_Toc107241519)

[II.2.9.3. Đề tài dự án 172](#_Toc107241520)

[II.2.9.4. Hỗ trợ liên kết 174](#_Toc107241521)

[II.2.9.5. Kế hoạch hỗ trợ liên kết 175](#_Toc107241522)

[II.2.10. Phân hệ Cảng cá 177](#_Toc107241523)

[II.2.10.1. Cấu hình danh mục 177](#_Toc107241524)

[II.2.10.2. Đăng ký tàu cá cập cảng 181](#_Toc107241525)

[II.2.10.3. Giám sát tàu cá 182](#_Toc107241526)

[II.2.10.4. Kiểm tra tàu cá rời cảng 184](#_Toc107241527)

[II.2.10.5. Thu phí tàu cập cảng 185](#_Toc107241528)

[II.2.10.6. Cấp biên nhận thủy sản bốc vỡ qua cảng 186](#_Toc107241529)

[II.2.10.7. Xác nhận khối lượng thủy sản còn lại 188](#_Toc107241530)

[II.2.10.8. Thu phí Ô cầu tàu và mặt bằng khác 189](#_Toc107241531)

[II.3. YÊU CẦU PHI CHỨC NĂNG 190](#_Toc107241532)

[II.3.1. Yêu cầu bảo mật, phân quyền người sử dụng 191](#_Toc107241533)

[II.3.2. Yêu cầu ghi log hệ thống 191](#_Toc107241534)

[II.3.3. Yêu cầu sao lưu và phục hồi dữ liệu 191](#_Toc107241535)

[II.3.4. Yêu cầu về tính khả dụng (Usability) 192](#_Toc107241536)

[II.3.5. Mã yêu cầu: yêu cầu về tính ổn định (Reliability) 192](#_Toc107241537)

[II.3.6. Yêu cầu về hiệu năng (Performance) 192](#_Toc107241538)

[II.3.7. Yêu cầu về tính hỗ trợ (Supportability) 192](#_Toc107241539)

[II.3.8. Yêu cầu các ràng buộc thiết kế (Design contraints) 192](#_Toc107241540)

[II.3.9. Yêu cầu về giao tiếp (Interfaces) 193](#_Toc107241541)

[II.3.10. Các yêu cầu về tài liệu người dùng và hỗ trợ trực tuyến 195](#_Toc107241542)

[II.3.11. Mã yêu cầu: các thành phần mua ngoài 195](#_Toc107241543)

[II.3.12. Các yêu cầu pháp lý, bản quyền và ghi chú khác 195](#_Toc107241544)

[II.3.13. Các tiêu chuẩn áp dụng 195](#_Toc107241545)

[II.4. ĐIỀU KIỆN NGHIỆM THU HỆ THỐNG 196](#_Toc107241546)

[PHỤ LỤC 196](#_Toc107241547)

# Tạo dự án Spring Boot đầu tiên

## Chuẩn bị

Đối với Spring Boot một trong hai IDE là **Eclipse** (miễn phí) và ***IntelliJ IDEA*** Ultimate (bản Community không có hỗ trợ Spring). Tải xuống tại đây:

A screen shot of a phone

AI-generated content may be incorrect. IntelliJ IDEA: <https://www.jetbrains.com/idea/download/>

## Spring initializr

Spring Initializr có thể truy cập trên web tại <http://start.spring.io/>, hoặc với IntelliJ thì có tích hợp luôn vào khi tạo project luôn.

A screenshot of a computer

AI-generated content may be incorrect.

## Hoàn tất

Sau khi xong, các bạn nhấn nút Generate là xong. Một file zip chứa source ban đầu sẽ được tải về, chỉ cần giải nén và bắt đầu code.

Cấu trúc project được khởi tạo sẵn như sau.

A screenshot of a computer program

AI-generated content may be incorrect.

# JAVA CORE

## Dependency injection áp dụng vào Spring Boot

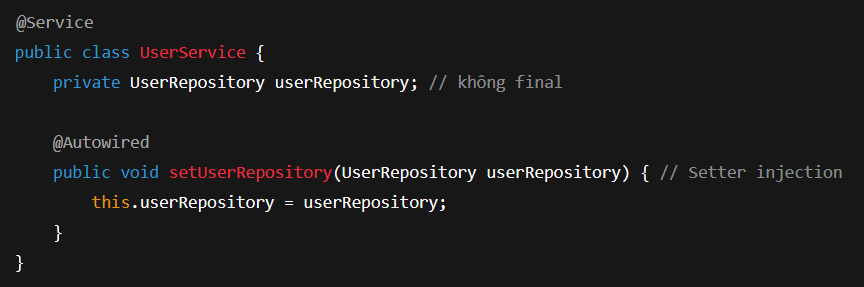
Có 3 loại chính:

* **Constructor-based injection:** Dùng inject các module bắt buộc. Các module được inject nằm trong constructor, và được gán lần lượt vào các field.
* **Setter-based injection:** Dùng inject các module tùy chọn. Mỗi module sẽ được inject thông qua setter, nằm ở tham số và cũng gán cho field nào đó.
* **@Autowired**: Sử dụng annotation @Autowired

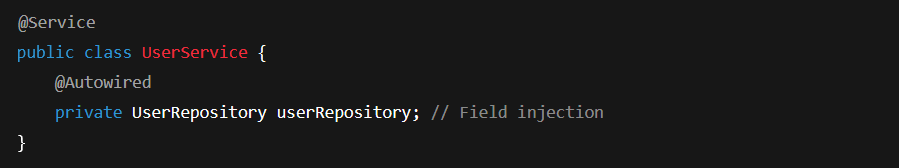
1. Constructor-based injection (tiêm qua constructor)



1. Setter-based injection (tiêm qua setter)



3. Field injection @Autowired



## Spring Bean

**Spring Bean** là một **đối tượng Java** do **Spring Container** quản lý vòng đời và các dependency của nó.

🡺 Những object mà được tạo và quản lý bởi **Spring Container** thì được gọi là Bean

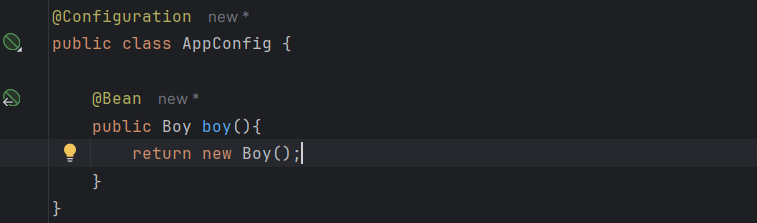
### Cách Tạo Bean

1. Dùng @Component, @Repository, @Service, @Controller
2. Dùng @Bean trong class có annotation @Configuration
   * **@Configuration**:
     + Là một Annotation đánh dấu trên một class, cho biết rằng lớp đó chứa các thông tin cấu hình cho ứng dụng.
     + Spring Boot sẽ tìm và quét các class được đánh dấu @Configuration để tạo và quản lý các beans.
   * **@Bean**:
     + Là một Annotation đánh dấu trên một method trong class được đánh dấu @Configuration.
     + Nó cho Spring Boot biết rằng method đó trả về một bean, và Spring Boot nên quản lý bean đó trong ứng dụng.

Ví dụ 1: @Component



* + **@Bean**:



***Ghi chú****: tại sao đã có cách 1 @Component mà lại xuất hiện cách 2 @Bean, vì 1 số trường hợp không phải lúc nào cũng dùng được @Component ví dụ ta muốn sử dụng 1 só thư viện bên ngoài ta không thể thực hiện @ Component đư*ợc , lúc đó dùng *@Bean*

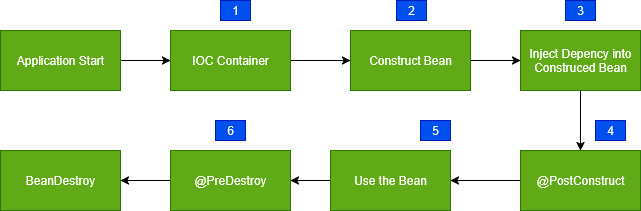
### Component Scan:

Khi ứng dụng Spring Boot bắt đầu chạy, thì nó sẽ tìm hết các class đánh dấu là bean trong chương trình và tạo bean. Quá trình tìm kiếm các bean này gọi là component scan.

@SpringBootApplication  
@ComponentScan(basePackages = {"com.example.demo", "com.example.demo.Orders"})  
public class DemoApplication {  
 public static void main(String[] args) {

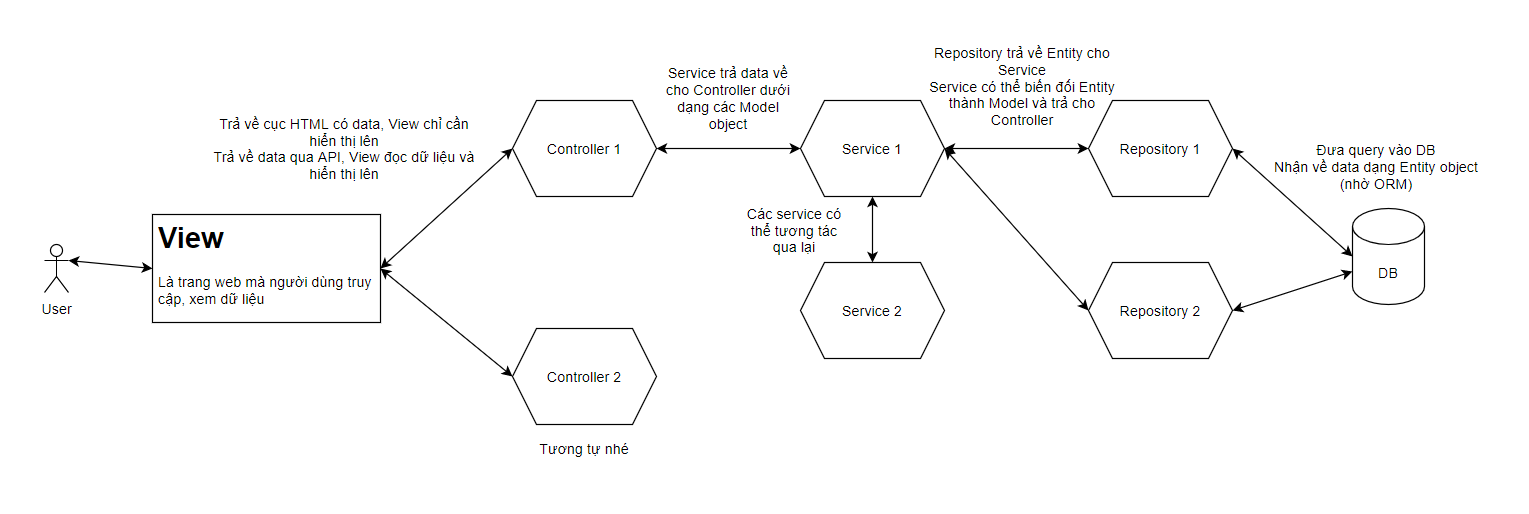
}

### Lifecycle của Bean



# Cơ chế Lazy: **Bean sẽ chỉ được khởi tạo khi nó thực sự được sử dụng**, thay vì tạo ra ngay khi container Spring start.

## luồng chạy của ứng dụng



# BÀI CODE MẪU

## JAVA SPRING BOOT + MYSQL + CRUD

### CẤU HÌNH

1. **Framework chính: Spring Boot 3.5.6**
2. **Ngôn ngữ: Java 23**
3. **Build tool: Gradle Kotlin DSL**
4. **Server Port: 8085**

🐳 **Thông tin Docker Services**

## JAVA SPRING BOOT + KAFKA

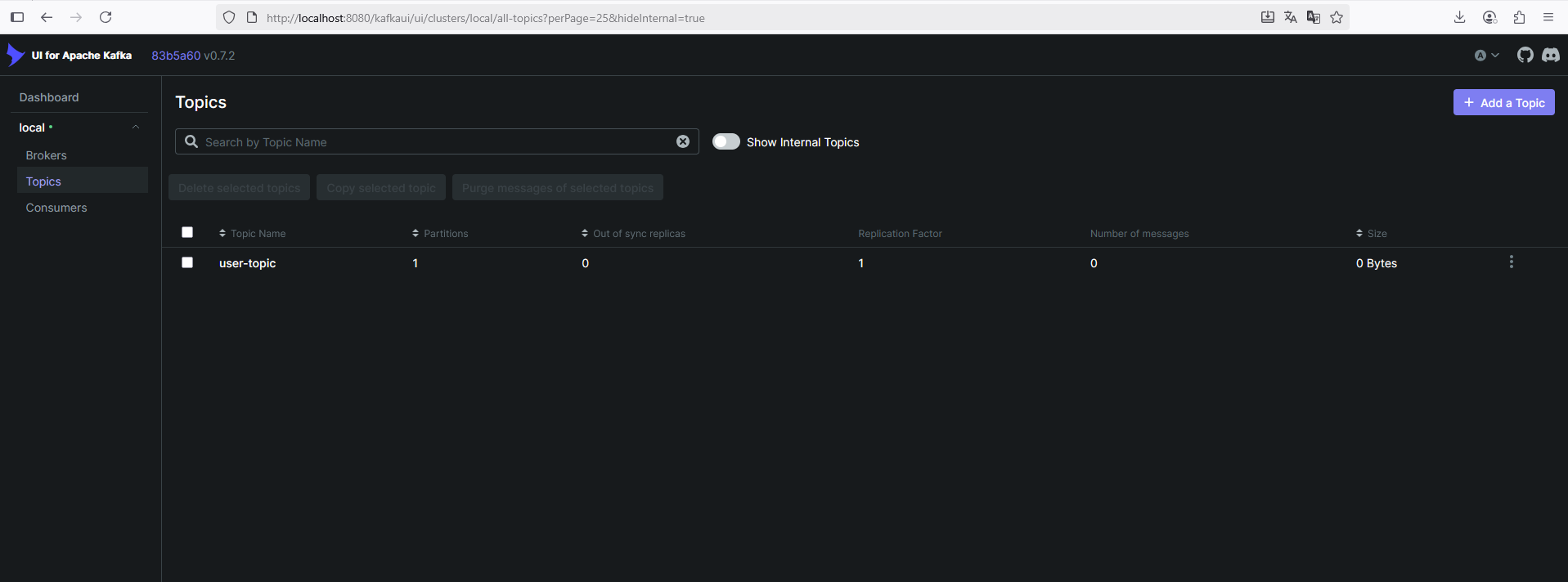
### CẤU HÌNH

* **Framework chính: Spring Boot 3.5.6**
* **Ngôn ngữ: Java 23**
* **Build tool: Gradle Kotlin DSL**
* **Server Port: 8085**

🐳 **Thông tin Docker Services**

📌 Kết nối cho Dev

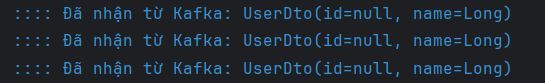
| **Component** | **Host** | **Port** | **Connection String** |
| --- | --- | --- | --- |
| Zookeeper | localhost | 2181 | localhost:2181 |
| Kafka Broker | localhost | 9092 | PLAINTEXT://localhost:9092 |
| Kafka UI (Console) | localhost | 8080 | http://localhost:8080/kafkaui |



**Postman request:**

| **Method** | **URL** | **Mô tả** |
| --- | --- | --- |
| POST | http://localhost:8085/api/users | Tạo User, send data lên kafka |

**Kết quả:**



## JAVA SPRING BOOT + MINIO

### CẤU HÌNH

* **Framework chính**: Spring Boot 3.5.6
* **Ngôn ngữ**: Java 23
* **Build tool**: Gradle Kotlin DSL
* **Server Port**: 8087

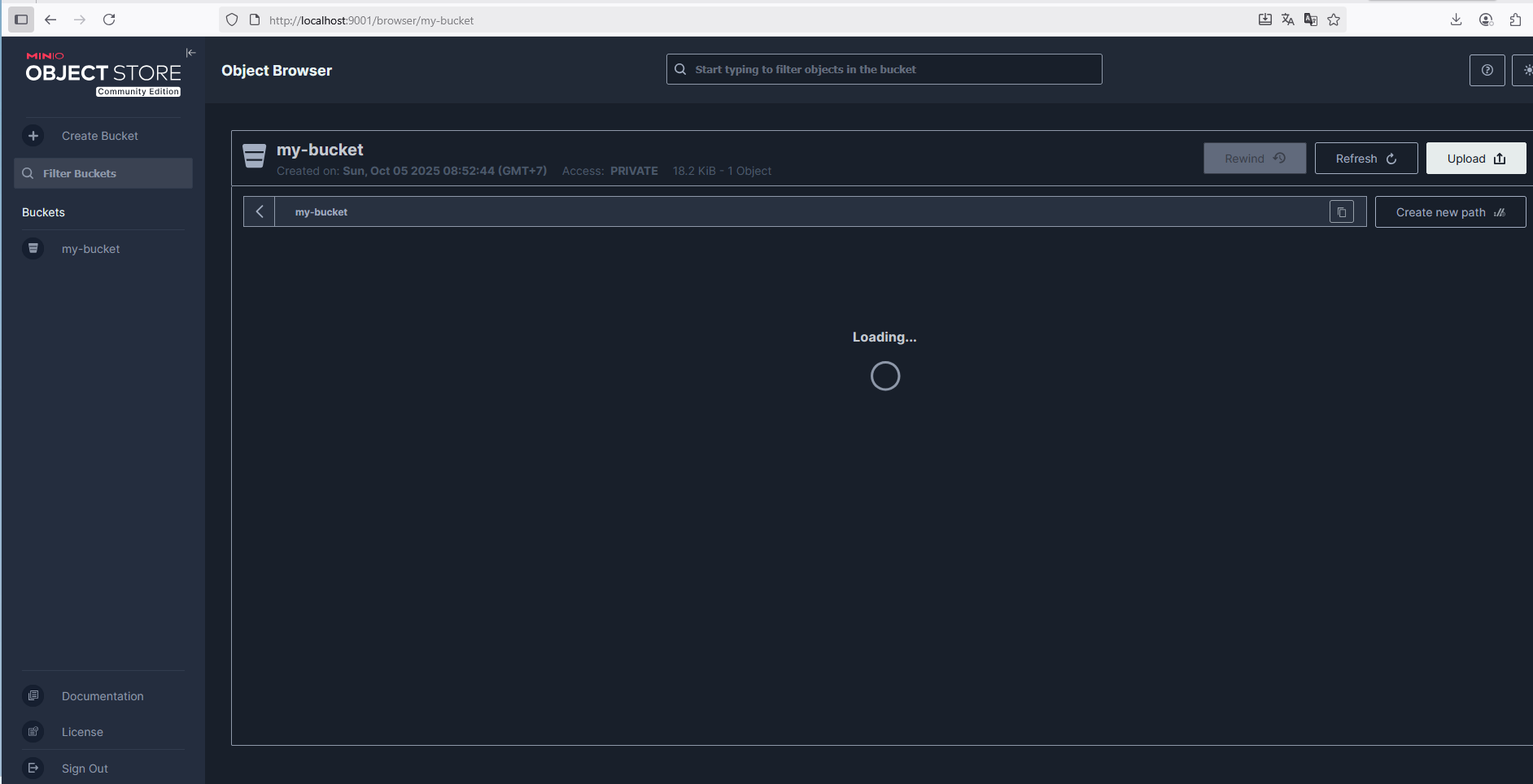
🐳 **Thông tin Docker Services**

### 2. ****MinIO Console (UI Web)****

* URL: <http://localhost:9001>

Login bằng:

* **Username**: minioadmin
* **Password**: minioadmin



**Postman request:**

| **Chức năng** | **Method** | **URL** | **Body (Postman)** | **Kết quả** |
| --- | --- | --- | --- | --- |
| Upload file | POST | http://localhost:8087/files/upload | - Body: **form-data**Key: file (type = File)Value: chọn file từ máy tính | File uploaded: <tên\_file> |
| Download file | GET | http://localhost:8087/files/download/{fileName} | Không có body | Tải về file (Postman → Save Response) |

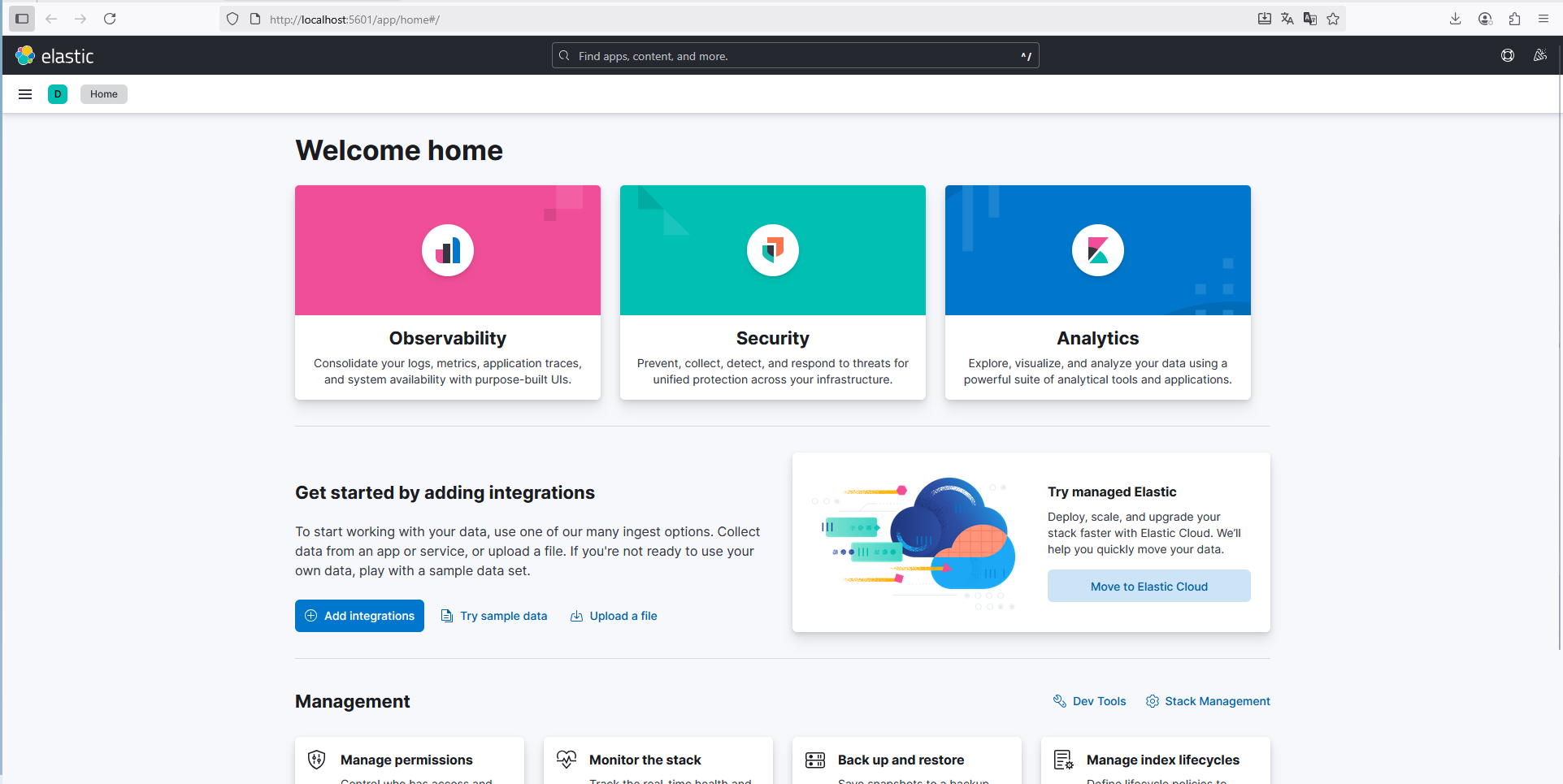
## JAVA SPRING BOOT + ELASTIC

### CẤU HÌNH

* **Framework chính**: Spring Boot 3.5.6
* **Ngôn ngữ**: Java 23
* **Build tool**: Gradle Kotlin DSL
* **Server Port**: 8086

🐳 **Thông tin Docker Services**

* *docker-compose up –build*
* *Giao diện web Kibana:* 
  + [*http://localhost:5601*](http://localhost:5601)
  + *http://localhost:5601/app/management*



**Postman request:**

| **Method** | **URL** | **Mô tả** |
| --- | --- | --- |
| POST | /api/users | Tạo User |
| GET | /api/users | Lấy tất cả Users |
| GET | /api/users/{id} | Lấy User theo ID |
| PUT | /api/users/{id} | Cập nhật User |
| DELETE | /api/users/{id} | Xóa User |
| GET | /api/users/search?email=abc@xyz.com | Tìm User theo Email |