



LẬP TRÌNH JAVA SPRING BOOT

**THỰC HÀNH DỰ ÁN: Thực Hiện Chức Năng Đơn
Giản Với Đề Tài Dịch Vụ Chuyển Đổi Tiền Tệ**

- ❑ Thực hành dự án với kiến trúc Microservice (MSA), Spring Cloud
 - ❖ Các kỹ thuật sử dụng annotations, JPA, Spring Cloud và database(cơ sở dữ liệu quan hệ). Có thể tùy chọn database
 - ❖ Kết hợp thêm với kỹ thuật Service Registry (Eureka Server), Load Balance Requests
 - ❖ Mô tả chức năng dự án, gồm có 2 dịch vụ chính
 - Dịch vụ thứ nhất(sv1) để lấy tỉ giá giữa hai loại tiền tệ từ database. Ví dụ usd và vnd tỉ giá là 23,000.00
 - Dịch vụ thứ hai(sv2) thực hiện tính số lượng tiền và tỉ giá bằng phép nhân và trả về kết quả
 - Các dịch vụ còn lại là thành phần trong Spring Cloud
-

- 1) Tạo một **Config Server** sử dụng git remote hoặc có thể lưu ở local
 - I. Gồm 3 file sau
 - `currency-exchange-service.properties`
 - `currency-exchange-service-dev.properties`
 - `currency-exchange-service-qa.properties`
 - II. Port của Config Server là **8888**
 - III. Nội dung 2 file **`currency-exchange-service-dev.properties`** và **`currency-exchange-service-qa.properties`** là cấu hình database với hai database khác nhau
 - 2) Tạo **Eureka Server** để làm máy chủ quản lý các service, Port là **8761**
-

3) Tạo **Currency Exchange Service** dùng để lấy tỉ giá tiền tệ, Port là **8000**

- Tạo một entity để lưu tỉ giá như sau
 - Các thuộc tính bao gồm: Long **id**, String **from**(column name 'currency_from'), String **to** (column name 'currency_to'), BigDecimal **conversionMultiple**, int **port**
- I. Tạo file **data.sql** để thực hiện init data với nội dung như sau
 - `insert into exchange_value(id,currency_from,currency_to,conversion_multiple,port) values(10001,'USD','VND',23300,0);`
 - `insert into exchange_value(id,currency_from,currency_to,conversion_multiple,port) values(10002,'EUR','VND',23800,0);`
 - II. Đăng ký đến **Config Server** với file **currency-exchange-service-dev.properties**
 - III. Tạo controller xử lý lấy tỉ giá tiền tệ từ database
 - IV. Định nghĩa tên của module là **currency-exchange-service**
 - V. Tạo một **Instance** với port là **8001** và đăng ký đến **Config Server** với file **currency-exchange-service-qa.properties**
 - VI. Đăng ký cả 2 instance với **Eureka Server**
-

4) Tạo **Currency Calculation Service** thực hiện tính toán, Port là **8100**

- I. Cài đặt **Feign Client** cho module
- II. Tạo class **CalculatedAmount**

```
@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
public class CalculatedAmount {
    private Long id;
    private String from;
    private String to;
    private BigDecimal conversionMultiple;
    private BigDecimal quantity;
    private BigDecimal TotalCalculatedAmount;
    private int port;
}
```

- III. Tạo một bean **RestTemplate** để load balanced

```
@Configuration
public class CurrencyCalculationConfig {
    @Bean
    @LoadBalanced // annotation này sẽ cân bằng tải cho
    mỗi requests
    public RestTemplate restTemplate() {
        return new RestTemplate();
    }
}
```

- IV. Viết controller xử lí tính toán chuyển đổi tiền tệ theo số lượng tiền kết hợp kỹ thuật **Feign Client** để gọi đến service lấy tỉ giá
- V. Viết controller tương tự như (IV) nhưng với kỹ thuật **RestTemplate** để gọi đến service lấy tỉ giá
- VI. Lưu ý không hardcode url khi gọi đến service phải dung tên được đăng ký đến **Eureka Server**
- VII. Tạo một **Instance** với port là **8101**
- VIII. Đăng ký cả 2 instance với **Eureka Server**

5) Tạo **Gateway** với Port là **8765**

- I. Đăng ký đến **Eureka Server**
 - II. Cài đặt **Currency Calculation Service** vào gateway. Lưu ý không hardcode cổng trong uri
 - III. Thay thế ví dụ như: lb://CURRENCY-CALCULATION-SERVICE/
-

- ❑ Cấu trúc project, tên biến, hàm... đều phải có ý nghĩa, sắp xếp đẹp
 - ❑ Vận dụng các annotations trong spring boot không chỉ giới hạn trong slide bài học
 - ❑ Sử dụng Github để push code lên, Gửi link Github sau khi hoàn thành bài làm, lưu ý để public
 - ❑ Team đào tạo sẽ tìm ra bài tốt nhất và sớm nhất để làm mẫu cho những bạn chưa xong, lưu ý là tự làm trước khi tham khảo bài người khác
 - ❑ Team đào tạo sẽ hỗ trợ các bạn xem bài và góp ý nếu có vấn đề hoặc để hoàn thiện hơn
-

Cảm Ơn Bạn Đã Chăm Chỉ!

