

# Migrating a Non-Reactive Spring Boot Application to WebFlux

## ## 1. Introduction

This guide outlines the necessary changes to migrate a traditional blocking Spring Boot application to a reactive WebFlux-based architecture using R2DBC for database interactions.

We will cover migration steps for:

- **H2 (In-Memory Database) in Reactive Mode**
- **MySQL/PostgreSQL in Reactive Mode**

---

## ## 2. Key Changes in Migration

### ### 2.1 Replace Spring MVC with WebFlux

- Update the dependencies to replace `spring-boot-starter-web` with `spring-boot-starter-webflux`.

#### #### **Changes in `pom.xml` (Maven) or `build.gradle` (Gradle)**

```
```xml
```

```
<!-- Remove Spring MVC dependency -->
```

```
<dependency>
```

```
  <groupId>org.springframework.boot</groupId>
```

```
  <artifactId>spring-boot-starter-web</artifactId>
```

```
  <scope>provided</scope>
```

```
</dependency>
```

```
<!-- Add Spring WebFlux -->
```

```
<dependency>
```

```
  <groupId>org.springframework.boot</groupId>
```

```
  <artifactId>spring-boot-starter-webflux</artifactId>
```

```
</dependency>
```

```
```
```

### ### 2.2 Replace JPA with R2DBC

- Remove `spring-boot-starter-data-jpa` and `hibernate`.
- Add `spring-boot-starter-data-r2dbc` and appropriate R2DBC driver.

#### #### **H2 R2DBC Dependency**

```
```xml
```

```
<dependency>
```

```
  <groupId>org.springframework.boot</groupId>
```

```

    <artifactId>spring-boot-starter-data-r2dbc</artifactId>
</dependency>
<dependency>
    <groupId>io.r2dbc</groupId>
    <artifactId>r2dbc-h2</artifactId>
    <scope>runtime</scope>
</dependency>

```

#### \*\*MySQL/PostgreSQL R2DBC Dependency\*\*

```

```xml
<!-- R2DBC dependency -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-r2dbc</artifactId>
</dependency>

```

```

<!-- R2DBC Driver for MySQL -->
<dependency>
    <groupId>dev.miku</groupId>
    <artifactId>r2dbc-mysql</artifactId>
</dependency>

```

```

<!-- R2DBC Driver for PostgreSQL -->
<dependency>
    <groupId>io.r2dbc</groupId>
    <artifactId>r2dbc-postgresql</artifactId>
</dependency>

```

### ### 2.3 Update Database Configuration

#### \*\*H2 Configuration (application.properties)\*\*

```

spring.r2dbc.url=r2dbc:h2:mem:///testdb
spring.r2dbc.username=sa
spring.r2dbc.password=
spring.sql.init.mode=always

```

#### \*\*MySQL Configuration (application.properties)\*\*

```

spring.r2dbc.url=r2dbc:mysql://localhost:3306/mydb
spring.r2dbc.username=root
spring.r2dbc.password=root

```

```
spring.sql.init.mode=always
```
```

```
#### **PostgreSQL Configuration (application.properties)**
```
```

```
spring.r2dbc.url=r2dbc:postgresql://localhost:5432/mydb
spring.r2dbc.username=postgres
spring.r2dbc.password=secret
spring.sql.init.mode=always
```
```

---

### ## 3. Summary of Changes

| Component<br>(R2DBC)                                                                                                             | Traditional Spring Boot (JPA)               | WebFlux |
|----------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|---------|
| ----- ----- -----                                                                                                                |                                             |         |
| <b>**Spring Dependency**</b>                                                                                                     | <code>`spring-boot-starter-web`</code>      |         |
| <code>`spring-boot-starter-webflux`</code>                                                                                       |                                             |         |
| <b>**Database Library**</b>                                                                                                      | <code>`spring-boot-starter-data-jpa`</code> |         |
| <code>`spring-boot-starter-data-r2dbc`</code>                                                                                    |                                             |         |
| <b>**Entity**</b>   <code>`@Entity`, `@Id`</code> (JPA)   <code>`@Table`, `@Id`</code> (R2DBC)                                   |                                             |         |
| <b>**Repository**</b>   <code>`JpaRepository`</code>   <code>`ReactiveCrudRepository`</code>                                     |                                             |         |
| <b>**Service Methods**</b>   <code>`List&lt;T&gt;`</code> (Blocking)   <code>`Flux&lt;T&gt;` / `Mono&lt;T&gt;`</code> (Reactive) |                                             |         |
| <b>**Controller Return Type**</b>   Direct objects   <code>`Mono&lt;T&gt;` / `Flux&lt;T&gt;`</code>                              |                                             |         |

---

### ## 4. Conclusion

By following the steps outlined above, you can successfully migrate your **\*\*Spring Boot JPA-based\*\*** application to a **\*\*Spring WebFlux + R2DBC-based\*\*** reactive application using **\*\*H2, MySQL, or PostgreSQL\*\***.