

# 2025 캡스톤 디자인 디버깅 핸드북



## Spring Boot 개발자를 위한 디버깅 핸드북

Capstone Design 2025-1

저작권: [develop6013@gmail.com](mailto:develop6013@gmail.com) -- dev-jm-1024

---



### 기술 스택



#### Frontend

- HTML5, CSS3, JavaScript
- Vue.js, Thymeleaf
- Tailwind CSS
- GSAP, Lottie Files



#### Backend

- Java, Kotlin
  - Spring Boot, Spring Data JPA, Spring Security
  - REST API, JWT
  - MySQL, MariaDB, MongoDB
- 



### 프로젝트 링크



#### Frontend Repository

Vue.js + JavaScript + Thymeleaf

```
https://github.com/dev-jm-1024/capstone-templates.git
```



#### Backend Repository

Spring Boot

```
https://github.com/dev-jm-1024/game-hub.git
```



Contact: [develop6013@gmail.com](mailto:develop6013@gmail.com) || [jon1024@naver.com](mailto:jon1024@naver.com)

---

## 주요 오류 및 해결 방법

### 1. Gradle 경고 메시지

#### ◆ 문제: Deprecated Gradle Features

```
Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.
```

#### 발생 원인:

- Gradle 8.x에서는 동작하지만, Gradle 9.0 이상에서는 삭제될 기능 사용
- `build.gradle` 파일의 문법 문제
- 사용 중인 플러그인 또는 라이브러리 버전 충돌

#### 해결 방법:

- ✓ "Shorten the command line and rerun" 버튼 클릭으로 해결 완료

### 2. Command Line Too Long 오류

#### ◆ 문제: Windows 명령어 길이 제한

```
Error running 'GameHubApplication'
Command line is too long.
Shorten the command line and rerun.
```

#### 발생 원인:

- Windows 환경에서 Java 실행 시 명령어 인자가 32,768자 제한 초과

#### 해결 방법:

- ✓ "Shorten the command line and rerun" 버튼 클릭

### 3. DataSource 구성 실패

#### ◆ 문제: DataSource URL 미지정

```
APPLICATION FAILED TO START
```

```
Description:
```

```
Failed to configure a DataSource: 'url' attribute is not specified and no
```

embedded datasource could be configured.

Reason:

Failed to determine a suitable driver class

#### 발생 원인:

- `spring.datasource.url` 미설정
- DB 드라이버 의존성 누락
- `application.properties` 오타
- JPA 설정만 하고 DataSource 설정 누락

#### 해결 방법

### MySQL/MariaDB 사용 시

```
# application.properties
spring.datasource.url=jdbc:mariadb://localhost:3306/your_db
spring.datasource.username=your_user
spring.datasource.password=your_pass
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MariaDBDialect
```

### H2 Database 사용 시

```
# application.properties
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driver-class-name=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=

spring.h2.console.enabled=true
```

### Gradle 의존성 제거

```
// 이 부분들을 주석 처리 또는 삭제
// implementation("org.springframework.boot:spring-boot-starter-data-jpa")
// runtimeOnly("com.mysql:mysql-connector-j")
// runtimeOnly("org.mariadb.jdbc:mariadb-java-client")
```

---

## 4. Spring Security 로그인 페이지 비활성화

## 해결 방법

### Gradle 의존성 제거

```
// 이 부분들을 주석 처리 또는 삭제
// implementation("org.springframework.boot:spring-boot-starter-security")
// implementation("org.thymeleaf.extras:thymeleaf-extras-springsecurity6")
// testImplementation("org.springframework.security:spring-security-test")
```

## 5. Vaadin 메모리 누수 경고

### ◆ 경고 메시지

```
WARN 14924 --- The web application [ROOT] appears to have started a thread named
[ForkJoinPool.commonPool-worker-...]
but has failed to stop it. This is very likely to create a memory leak.
```

#### 발생 원인:

- Vaadin 개발 모드에서 ForkJoinPool을 사용한 비동기 작업
- 애플리케이션 종료 시 스레드가 명확하게 종료되지 않음

#### 해결 방법:

- **개발 환경:** 그냥 무시해도 됨 (경고일 뿐, 에러 아님)
- **운영 배포 시:** application.properties 에 다음 설정 추가

```
# application.properties
vaadin.productionMode=true
```

## 6. EntityManagerFactory 생성 실패

### ◆ 문제: Bean 생성 실패

```
ERROR [org.springframework.boot.SpringApplication] : Application run failed
Caused by: org.springframework.beans.factory.BeanCreationException:
Error creating bean with name 'entityManagerFactory'
```

#### 발생 원인:

- JPA 설정 또는 DB 연결 문제
- application.properties 에 DB 정보 누락/오타

- DB 드라이버 누락 또는 버전 미일치
- 포트 번호 오류 (예: MariaDB 3306, MySQL 3307)

## 해결 방법

### MariaDB 설정

```
spring.datasource.url=jdbc:mariadb://localhost:3306/your_db
spring.datasource.username=root
spring.datasource.password=your_pw

spring.jpa.hibernate.ddl-auto=none
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MariaDBDialect
```

### MySQL 설정

```
spring.datasource.url=jdbc:mysql://localhost:3307/your_db?
useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=Asia/Seoul
spring.datasource.username=your_name
spring.datasource.password=your_pw
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

### Gradle 의존성

```
// MariaDB
implementation("org.springframework.boot:spring-boot-starter-data-jpa")
implementation("org.mariadb.jdbc:mariadb-java-client")

// MySQL (jar 파일 사용 시)
implementation(files("libs/mysql-connector-j-8.0.33.jar"))
```

### Entity 스캔 설정

```
@SpringBootApplication(scanBasePackages = "kr.plusb3b.games.gamehub")
@EnableJpaRepositories(basePackages = "kr.plusb3b.games.gamehub.repository")
@EntityScan(basePackages = "kr.plusb3b.games.gamehub.api.dto")
```

## 7. JPA Repository 도메인 타입 해결 실패

### ◆ 문제: 제네릭 타입 누락

```
java.lang.IllegalArgumentException:
Could not resolve domain type of interface
```

```
kr.plusb3b.games.gamehub.repository.userrepo.RequestUserJoinInfoRepository
```

#### 발생 원인:

- JPA Repository 인터페이스에서 제네릭 타입 미지정
- `@Entity` 어노테이션 누락
- `@Id` 필드 누락

#### 해결 방법

#### 잘못된 예:

```
public interface RequestUserJoinInfoRepository extends JpaRepository {  
    // 제네릭 누락  
}
```

#### 올바른 예:

```
public interface RequestUserJoinInfoRepository extends JpaRepository<Entity,  
Long> {  
    // 제네릭 타입 명시  
}
```

---

## 8. 의존성 주입 실패

### ◆ 문제: `UnsatisfiedDependencyException`

```
UnsatisfiedDependencyException:  
Error creating bean with name 'boardController'
```

#### 발생 원인:

- Controller가 의존하는 Service나 Repository가 Bean으로 등록되지 않음
- `@Service`, `@Repository`, `@Component` 어노테이션 누락
- `@ComponentScan` 범위에 포함되지 않음

#### 해결 방법:

- 해당 클래스에 적절한 어노테이션 추가
- 사용하지 않는 파일은 삭제

---

## 9. Spring Security 패턴 파싱 오류

## ◆ 문제: HTTP Status 500

```
.requestMatchers("/", "/my/**", "/game-hub/*", "/games/**", "/boards/**/**")
```

### 발생 원인:

- `/boards/**/**` 는 `PatternParseException`을 발생시키는 잘못된 패턴
- Spring Web 6에서 `**` 뒤에 추가 경로 요소가 오면 예외 발생

### 해결 방법:

```
.requestMatchers("/", "/my/**", "/game-hub/*", "/games/**", "/boards/**")
```

## 🚫 10. 브라우저 렌더링 실패

### ◆ 문제: 빨간 금지 마크 표시

### 발생 원인:

- `login.html` 템플릿 없음
- Spring Security에서 `/login` 경로 접근 차단
- `SecurityConfig` 에서 `/login` 경로에 `permitAll()` 설정 누락

### 해결 방법:

- 필요한 HTML 템플릿 생성
- Security 설정에서 로그인 페이지 경로 허용

## 🔒 11. JWT 로그인과 Form 로그인 충돌

### ◆ 현재 상황

```
// Spring Security form 로그인
.formLogin(formLogin -> formLogin
    .loginPage("/login/login-form")
    .defaultSuccessUrl("/index", true)
    .permitAll()
)
```

```
<!-- JWT API 호출 -->
<form method="post" th:action="@{'/game-hub/api' + ${apiVersion} + '/login'}">
```

### 문제점:

- `formLogin()` 설정: Spring Security가 `/login` 경로를 폼 로그인으로 처리
- HTML form: `/game-hub/api/v1/login` 으로 JWT API 호출
- Spring Security와 JWT 로그인이 충돌

### 해결 방법

#### JWT 로그인만 사용하는 경우:

```
http
.csrf(csrf -> csrf.disable()) // JWT는 일반적으로 CSRF 불필요
.authorizeHttpRequests(auth -> auth
    .requestMatchers("/login/**", "/game-hub/api/v1/login").permitAll()
    .anyRequest().authenticated()
)
.addFilterBefore(new JwtAuthenticationFilter(jwtProvider),
UsernamePasswordAuthenticationFilter.class)
.formLogin(form -> form.disable());
```

## 12. JWT 서명 키 보안 문제

### ◆ 문제: WeakKeyException

```
io.jsonwebtoken.security.WeakKeyException: The signing key's size is 88 bits
which is not secure enough for the HS256 algorithm.
```

#### 발생 원인:

- HS256 알고리즘은 최소 256비트(32바이트)의 키 길이 요구
- 현재 사용 중인 키가 88비트(11바이트)로 너무 짧음
- `String secretKey = "mysecret";` 같은 짧은 키 사용

#### 해결 방법:

```
# application.properties
jwt.secret=보다_길게_작성할_것_최소_32바이트_이상_필요합니다
```

## 13. @RequestBody vs @ModelAttribute 구분

### 비교표



구분	@RequestBody	@ModelAttribute
Content-Type	application/json	application/x-www-form-urlencoded
데이터 소스	Request Body	Request Parameter
HTTP 방식	주로 REST API (fetch, axios)	주로 HTML form 전송
바인딩 방식	HttpMessageConverter 사용	WebDataBinder 사용
사용 포맷	JSON, XML 등	form-data, query string

## 코드 예제

### @RequestBody 사용

```
@PostMapping
public ResponseEntity<?> method(@RequestBody Dto dto) {
    // ...
}
```

```
fetch("/api", {
  method: "POST",
  headers: {
    "Content-Type": "application/json"
  },
  body: JSON.stringify({ data1: "123", data2: "456"})
});
```

### @ModelAttribute 사용

```
@PostMapping
public ResponseEntity<?> method(@ModelAttribute Dto dto) {
    // ...
}
```

```
<form action="" method="post">
  <input type="text" name="data1">
  <input type="text" name="data2">
</form>
```

## 14. JPA 업데이트/삭제 쿼리 트랜잭션 오류

- ◆ 문제: TransactionRequiredException

Caused by: jakarta.persistence.TransactionRequiredException: Executing an update/delete query

#### 발생 원인:

- UPDATE 또는 DELETE 쿼리 실행 시 트랜잭션으로 감싸지지 않음
- `@Modifying` 어노테이션 누락
- `@Transactional` 어노테이션 누락

#### 해결 방법

```
@Modifying
@Transactional
@Query("UPDATE UserAuth u SET u.authLastLogin = :time WHERE u.authUserId = :userId")
void updateLastLoginByUserId(@Param("userId") String userId, @Param("time")
LocalDateTime time);
```

#### 주요 사항:

- `@Modifying`: UPDATE/DELETE 쿼리에 필수
- `@Transactional`: 트랜잭션 활성화
- `clearAutomatically = true`: 영속성 컨텍스트 동기화 (권장)

## 핵심 정리

### 자주 발생하는 문제 체크리스트

문제 유형	해결 방법
Gradle 경고	"Shorten command line" 클릭
DataSource 오류	DB 설정 확인, 의존성 추가
EntityManagerFactory 실패	DB 연결 정보, 포트 번호 확인
Repository 오류	제네릭 타입 명시, @Entity 확인
Security 패턴 오류	<code>/**/**</code> 패턴 수정
JWT 키 오류	32바이트 이상 시크릿 키 사용
JPA 트랜잭션 오류	<code>@Modifying</code> + <code>@Transactional</code>

## 개발 팁

1. **DB 연결 문제**: 포트 번호와 드라이버 클래스를 항상 확인
2. **Security 설정**: JWT 사용 시 `formLogin()` 비활성화

3. **JPA 설정:** Entity 스캔 범위와 Repository 패키지 확인
  4. **의존성 관리:** 사용하지 않는 의존성은 주석 처리
  5. **프로덕션 배포:** Vaadin `productionMode=true` 설정
- 

## 지원 및 연락처

### GitHub:

- Frontend: <https://github.com/dev-jm-1024/capstone-templates.git>
- Backend: <https://github.com/dev-jm-1024/game-hub.git>

**Email:** [develop6013@gmail.com](mailto:develop6013@gmail.com) || [jon1024@naver.com](mailto:jon1024@naver.com)

---

 **Happy Debugging!** 이 핸드북이 Spring Boot 개발 여정에 도움이 되기를 바랍니다!

```
import org.springframework  
import springframework.steretype;
```

[캡스톤 디자인 25.05 2주차 - 진행현황](#)