

스프링 시작하기

스프링 jaypt 암호화

참조

<https://github.com/dron512/mhkakaoJWT>

1.1 What Is jasypt?

Jasypt(Java Simplified Encryption)는 Java 언어로 개발된 간단한 암호화 라이브러리입니다. 이 라이브러리는 개발자가 암호화와 복호화를 쉽게 구현할 수 있도록 도와줍니다. Jasypt는 데이터베이스 비밀번호, 사용자 정보, 보안 토큰 등과 같은 중요한 정보를 안전하게 저장하고 전송하기 위해 사용될 수 있습니다.

```
<!-- https://mvnrepository.com/artifact/com.github.ulisesbocchio/jasypt-spring-  
boot-starter -->  
  <dependency>  
    <groupId>com.github.ulisesbocchio</groupId>  
    <artifactId>jasypt-spring-boot-starter</artifactId>  
    <version>3.0.5</version>  
  </dependency>
```

DB 정보를 암호화 해서 application.properties 입력하도록 하겠습니다.

1.2 실행방법 설정

Build and runModify options ▾ Alt+M

java 17 SDK of 'restapi03' module ▾

-Djasypt.encryptor.password=mh1234mh1234mh1234mh1234mh1234mh1234mh1234mh1234mh1234

\$ ↵ ↗

com.mh.restapi03.Restapi03Application

\$

Press Alt for field hints

Active profiles:

Comma separated list of profiles

On frame deactivation: Update classes and resources ▾ ×

Open run/debug tool window when started ×

Add dependencies with “provided” scope to classpath ×

modify options -> add vm options 입력한다

1.3 프로젝트 생성

```
jasypt.encryptor.password=${PASSWORD}
```

application.properties 에 Jasypt에서 사용할 비밀번호 추가 하도록 합니다.

```
@Configuration
```

```
@EnableEncryptableProperties
```

```
public class JasyptConfig {
```

```
    @Value("${jasypt.encryptor.password}")|
```

```
    private String password;
```

```
@Bean
```

```
public PooledPBEStrngEncryptor jasyptStringEncryptor() {
```

```
    PooledPBEStrngEncryptor encryptor = new PooledPBEStrngEncryptor();
```

```
    encryptor.setPoolSize(4);
```

```
    encryptor.setPassword(password);
```

```
    encryptor.setAlgorithm("PBEWithMD5AndTripleDES");
```

```
    return encryptor;
```

```
}
```

```
}
```

@EnableEncryptableProperties
jasypt 를 사용할수있도록 활성화 한다.

1.4

@Test

```
void contextLoads() {
    String password = "";
    PooledPBEStrategyEncryptor encryptor = new PooledPBEStrategyEncryptor();
    encryptor.setPoolSize(4);
    encryptor.setPassword(password);
    encryptor.setAlgorithm("PBEWithMD5AndTripleDES");
    String content = "";    // 암호화 할 내용
    String encryptedContent = encryptor.encrypt(content); // 암호화
    String decryptedContent = encryptor.decrypt(encryptedContent); // 복호화
    System.out.println("Enc : " + encryptedContent + ", Dec: " + decryptedContent);

    encryptor = new PooledPBEStrategyEncryptor();
    encryptor.setPoolSize(4);
    encryptor.setPassword(password);
    encryptor.setAlgorithm("PBEWithMD5AndTripleDES");
    content = "";    // 암호화 할 내용
    encryptedContent = encryptor.encrypt(content); // 암호화
    decryptedContent = encryptor.decrypt(encryptedContent); // 복호화
    System.out.println("Enc : " + encryptedContent + ", Dec: " + decryptedContent);
}
```



1.4

application.properties 에 암호화 문자열을 입력

```
id: ENC(RSzLlN+ATBeCdhklo8GTZJH/Dvue1zXwvEDrqGjev3LqJekBfwvBRmeGcufBc0I) #rest api ?  
secret: ENC(1jD3o6dRKTMB0AzoAXEZYjFKIC4LeCIy5El0HZIaXYS2263nlxTqSzipGy72rPkh) # ?????? ??? ?
```

고맙습니다.

