비밀번호 암호화

💜회원 등록 시 '비밀번호'는 사용자가 입력한 문자 그대로 DB 에 등록하면 안 됩니다.
'정보통신망법, 개인정보보호법' 에 의해 비밀번호 암호화(Encryption)가 의무입니다.

- 양방향 암호 알고리즘
  - 암호화: 평문 → **(암호화 알고리즘)** → 암호문
  - 복호화: 암호문 → (암호화 알고리즘) → 평문
- 단방향 암호 알고리즘
  - 암호화: 평문 → **(암호화 알고리즘)** → 암호문
  - 복호화: **불가** (

단방향 암호화(One-Way Encryption)

단방향 암호화란 한쪽 방향으로만 암호화를 한다는 의미이다. 즉 암호화만 가능하고 복호화는 할 수 없다.그렇기 때문에 비밀번호를 관리할 때 유용하게 사용된다.비밀번호를 단방향 암호화 방식으로 저장하는 경우에는 패스워드 DB가 노출되어도 안전하다.패스워드를 검증할 때에는 사용자로부터 입력받은 비밀번호를 똑같은 방식으로 암호화하여 암호화된 패스워드끼리 비교를 하면 된다.

유저가 비밀번호를 잊어버렸을 때는 찾기가 불가능하다.대신 비밀번호 변경 메일, SMS 인증을 통해 새로운 비밀번호를 입력하도록 하면 된다.

대표적으로 많이 사용하고 있는 알고리즘은 **SHA-256 암호화 알고리즘**이다.

❓그럼 **사용자**가 로그인할 때는 **암호화된 패스워드를 기억**해야 할까요?

- Password 확인절차

1. 사용자가 로그인을 위해 "**아이디, 패스워드 (평문)**" 입력 → 서버에 로그인 요청
    1. 서버에서 패스워드 (평문) 을 암호화
        1. 평문 → **(암호화 알고리즘)** → 암호문
2. DB 에 저장된 "**아이디, 패스워드 (암호문)**"와 일치 여부 확인

- Password Matching

```
// 비밀번호 확인
if(!passwordEncoder.matches("사용자가 입력한 비밀번호","저장된 비밀번호")){
  throw new IllegalAccessError("비밀번호가 일치하지 않습니다.");
  }
```

- boolean matches(CharSequence rawPassword, String encodedPassword);
  - rawPassword : 사용자가 입력한 비밀번호
  - encodedPassword : 암호화되어 DB 에 저장된 비밀번호

암호화 기능 추가하는 방법

WebSecurityConfig

```
@Configuration
@EnableWebSecurity// 스프링 시큐리티 지원하능하게 함
public class WebSecurityConfig{

   @Bean//비밀번호 암호화 기능 등록
   public PasswordEncoder passwordEncoder(){
   return new BcryptPasswordEncoder();
   }
   }
```