

20. 선택적 도전 과제-3

1. 엔티티 수정

- 로그인 시도 횟수는 영구적으로 저장되어야 할듯
- user 엔티티에 로그인 시도 횟수를 추가하자
- User

```
@Column({ default: 0 })  
loginAttempts: number;
```

2. signIn 메소드 수정

- auth.service.ts

```
async signIn(username: string, pass: string): Promise<{ access_token: string,  
refresh_token: string }> {  
  const user = await this.userService.findByUsername(username);  
  if (!user) {  
    throw new UnauthorizedException();  
  }  
  
  if (user.loginAttempts >= 5) {  
    throw new UnauthorizedException('최대 로그인 시도 횟수를 초과하였습니다.');  }  
  
  if (!(await this.comparePasswords(pass, user.password))) {  
    user.loginAttempts += 1;  
    await this.userRepository.save(user);  
    throw new UnauthorizedException();  
  }  
  
  // 로그인 성공 시, 로그인 시도 횟수 초기화  
  user.loginAttempts = 0;  
  await this.userRepository.save(user);  
}
```

```

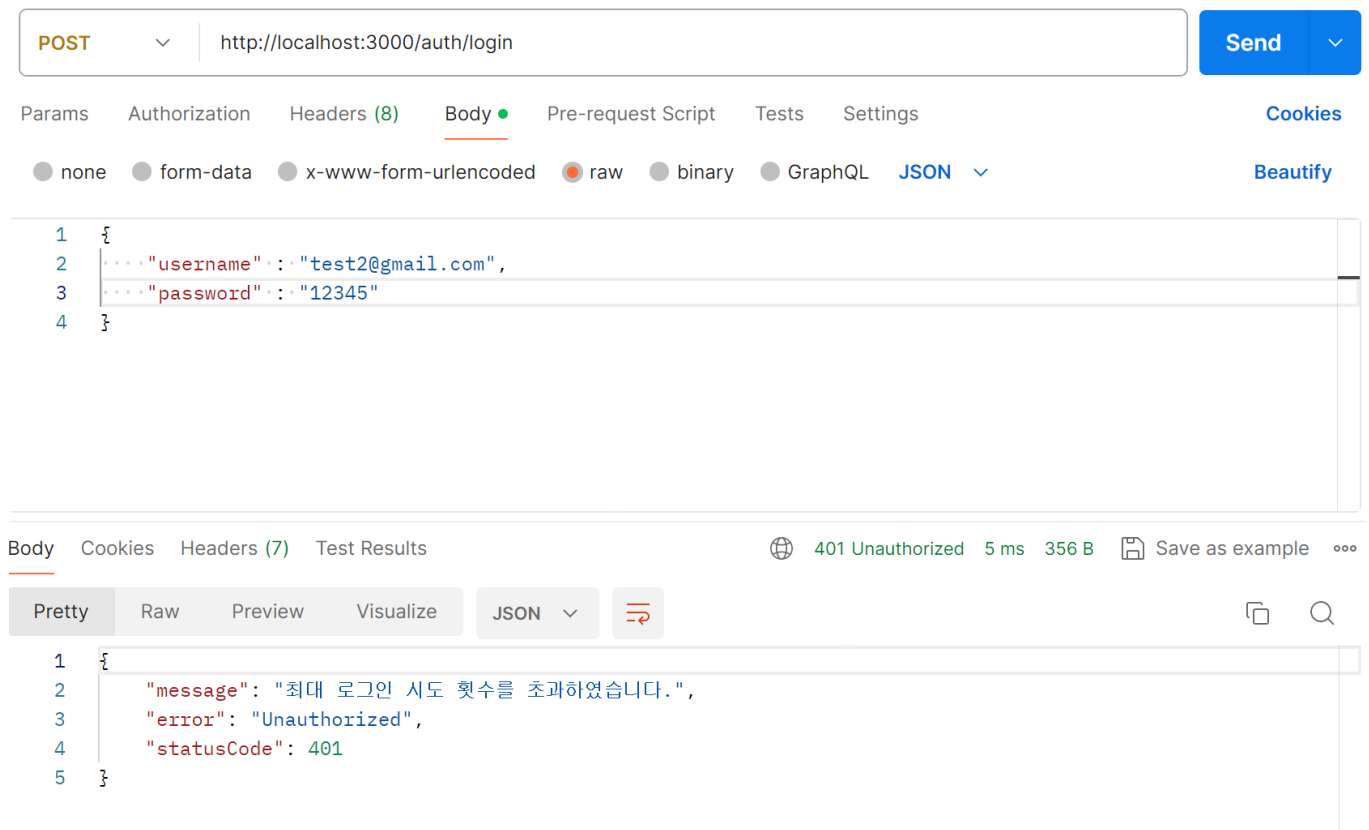
const payload = { username: user.username, sub: user.id, role: user.role };
const refresh_token = await this.jwtService.signAsync(payload, {
  expiresIn: '7d' // Refresh 토큰 유효기간 설정
});

// refresh token 업데이트
user.refreshToken = refresh_token;
await this.userRepository.save(user);

return {
  access_token: await this.jwtService.signAsync(payload, {
    expiresIn: '60s'
  }),
  refresh_token,
};
}

```

3. 구현 확인



POST http://localhost:3000/auth/login

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings Cookies Beautify

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▾

```

1 {
2   "username": "test2@gmail.com",
3   "password": "12345"
4 }

```

Body Cookies Headers (7) Test Results 401 Unauthorized 5 ms 356 B Save as example

Pretty Raw Preview Visualize JSON ▾

```

1 {
2   "message": "최대 로그인 시도 횟수를 초과하였습니다.",
3   "error": "Unauthorized",
4   "statusCode": 401
5 }

```