



# Spring Boot Security + JWT



# JWT nima?

1. JWT JSON Web Token degan ma'noni anglatadi. JSON Web Token (JWT) ochiq standart (RFC 7519) bo'lib, JSON ob'ekti sifatida tomonlar o'rtasida ma'lumotni xavfsiz uzatishning ihcham va mustaqil usulini belgilaydi.
2. Ushbu ma'lumot raqamli imzolanganligi sababli tekshirilishi va ishonchli bo'lishi mumkin. Mijoz faqat bir marta hisob ma'lumotlari yordamida server bilan autentifikatsiya qilishi kerak bo'ladi. Bu vaqt ichida server hisob ma'lumotlarini tekshiradi va mijozga JSON Web Token (JWT) ni qaytaradi.
3. Kelajakdagi barcha so'rovlar uchun mijoz ushbu JSON Web Token (JWT) yordamida serverga o'zini autentifikatsiya qilishi mumkin va shuning uchun foydalanuvchi nomi va parol kabi hisob ma'lumotlarini yuborishi shart emas.

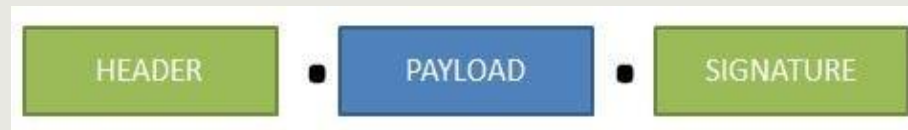
# Spring Security va JWT kutubxonalarini qo'shing

```
<dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-security</artifactId>  
</dependency>
```

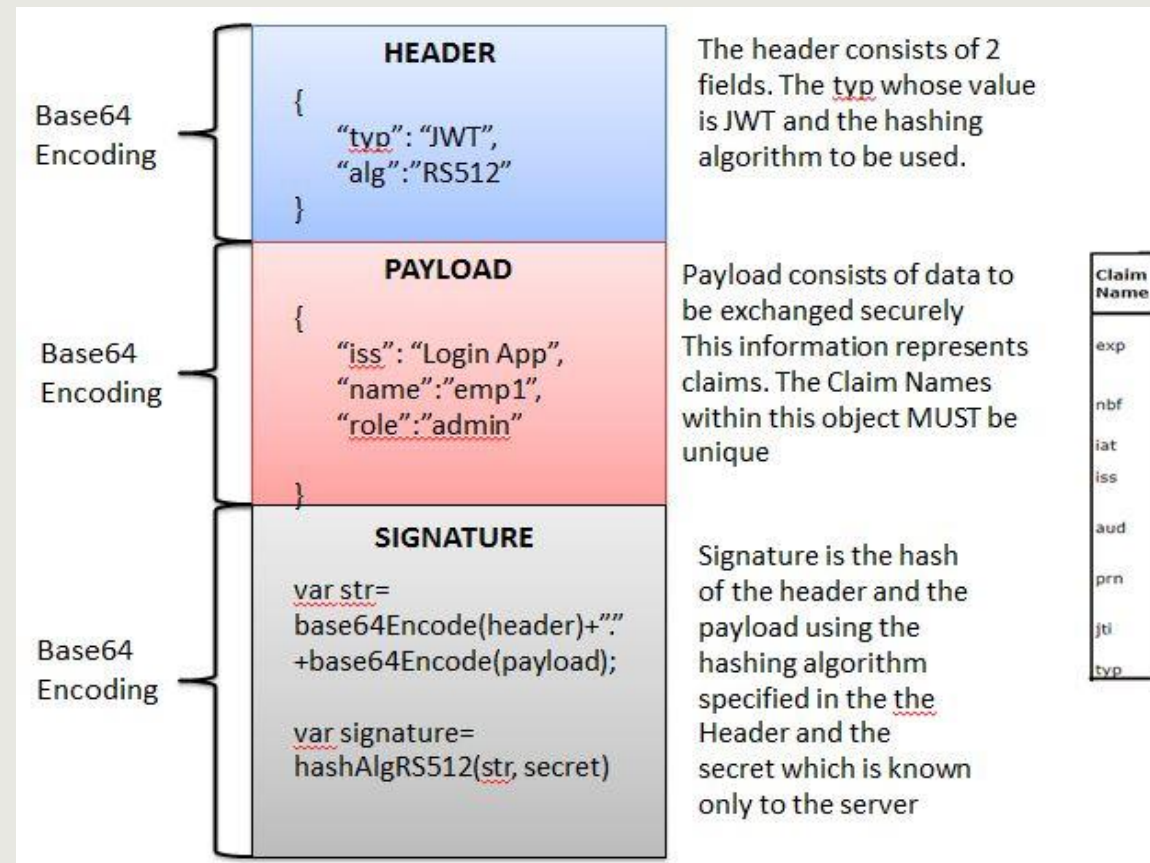
```
<dependency>  
    <groupId>io.jsonwebtoken</groupId>  
    <artifactId>jjwt</artifactId>  
    <version>0.9.1</version>  
</dependency>
```

# JWT tuzilishi:

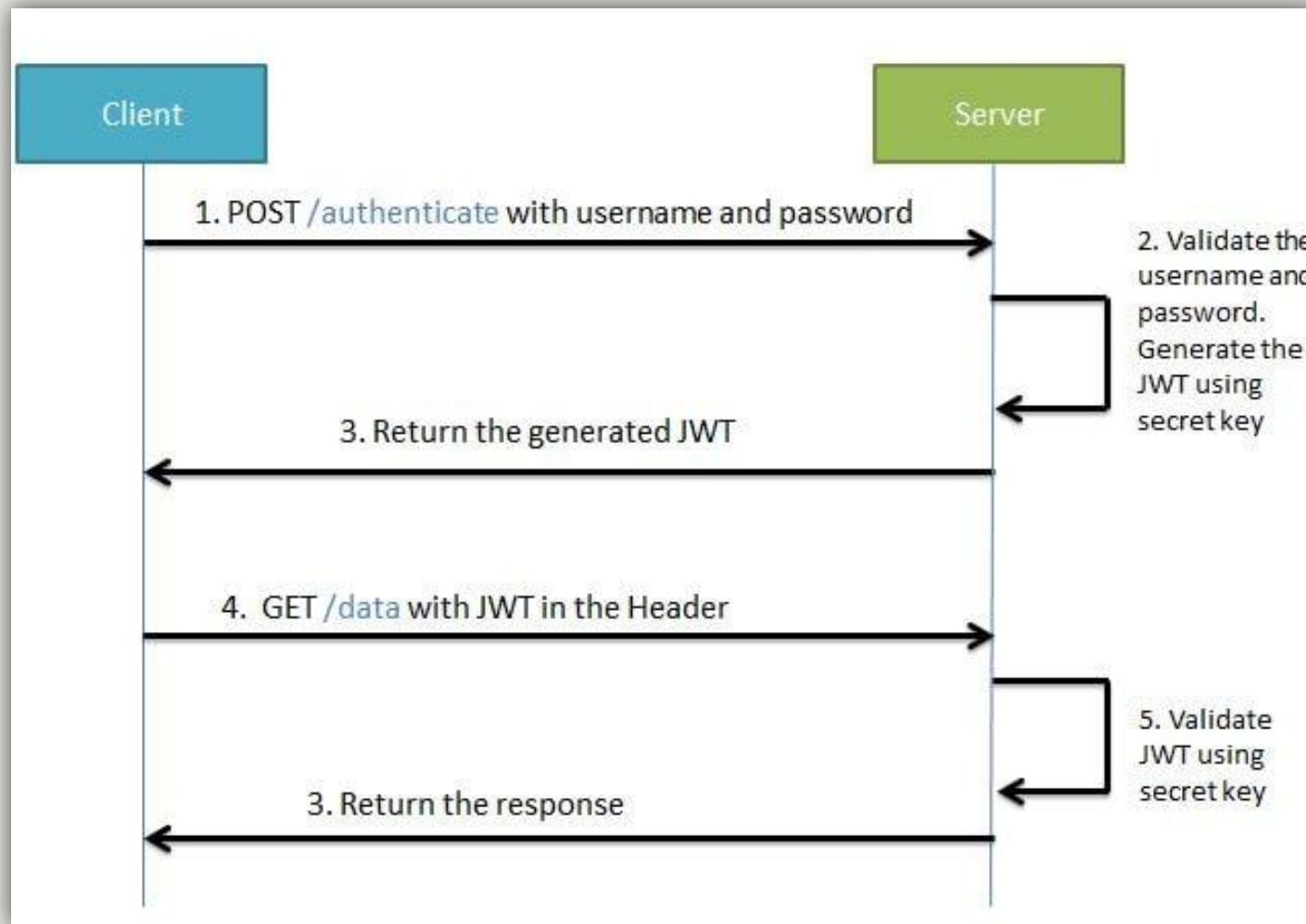
JWT quyidagi formatga  
ega `header.payload.signature` JWT:



tuzilishi :



# JWT qanday ishlatilishining ish jarayoni

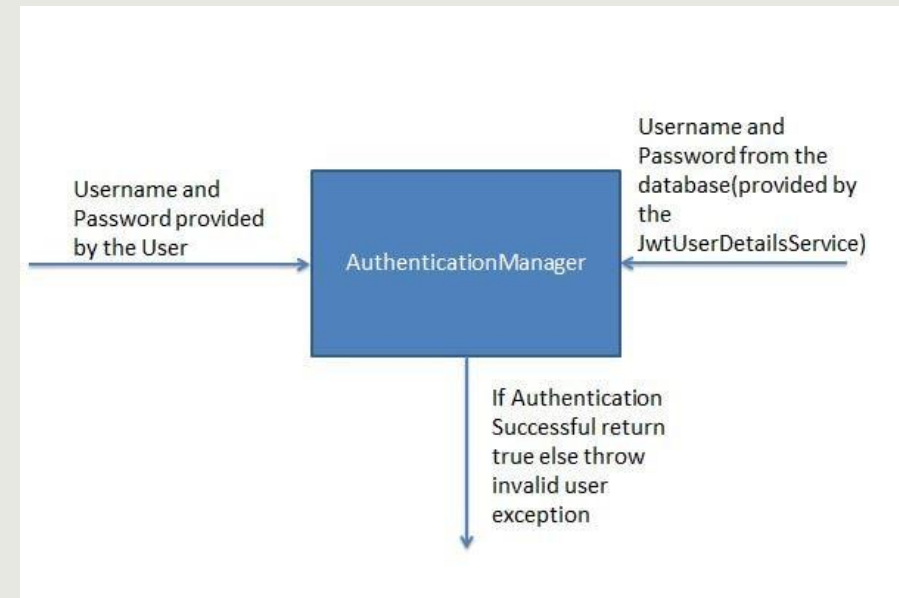
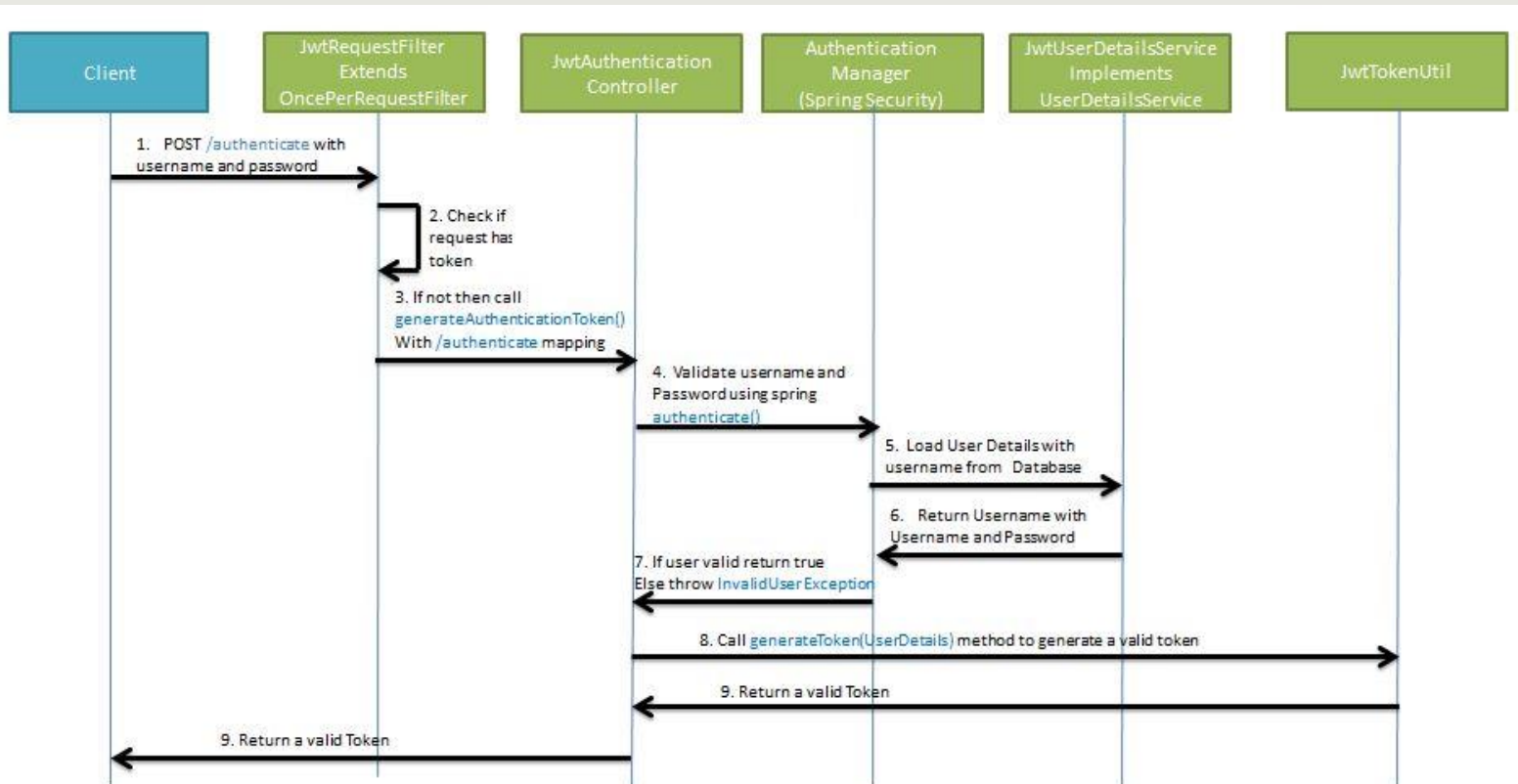


# Spring boot va JWT konfiguratsiyasi

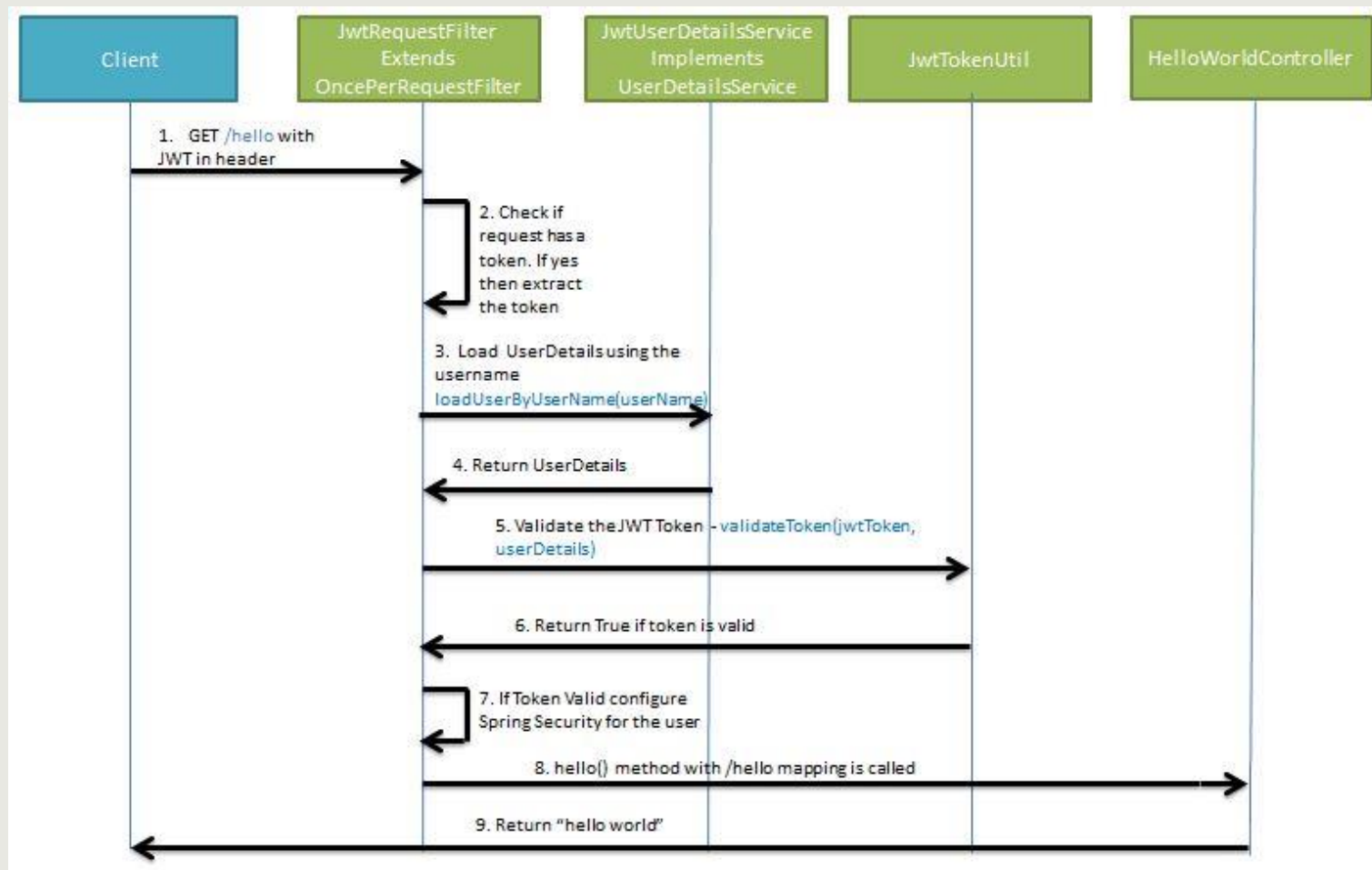


- Biz Spring Security va JWT ni ikkita operatsiyani bajarish uchun sozlaymiz.
- 1. **JWT** yaratish - POST API-ni xaritalash **/autentifikatsiya qilish** bilan ochish  
Foydalanuvchi nomi va parolni kiritishda u JSON Web Token (JWT) ni yaratadi.
- 2. **JWTni tekshirish - Agar foydalanuvchi /hello** bilan GET API ga kirishga harakat qilsa . Agar so'rovda yaroqli JSON Web Token (JWT) bo'lsa, u kirishga ruxsat beradi.

# JWT yaratish operatsiyalari uchun bosqichma bosqich oqimi quyidagicha:



# JWT tasdiqlanishi:





Token dan har qanday ma'lumotni olish uchun bizga maxfiy kalit kerak bo'ladi

```
private Claims getAllClaimsFromToken(String token) {  
  
    return Jwts.parser().setSigningKey(secret).parseClaimsJws(token).  
getBody();  
}
```

# Ma`lumotlardan foydalangan holda token generatsiya qilish metodi:

```
private String doGenerateToken(Map<String, Object> claims, String subject) {  
  
    return Jwts.builder()  
        .setClaims(claims)  
        .setSubject(subject)  
        .setIssuedAt(new Date(System.currentTimeMillis()))  
        .setExpiration(new Date(System.currentTimeMillis() +  
JWT_TOKEN_VALIDITY * 1000))  
        .signWith(SignatureAlgorithm.HS512, secret).compact();  
}
```



# Etiboringiz uchun raxmat

---

Tuzuvchi: Sherjonov Jahongir

