



Seoul  
Software  
ACademy

# 웹 개발자 부트캠프 과정

SeSAC x CODINGOn

With. 팀 뽀빠

# Spring Security

# 인증(Authentication) vs. 인가(Authorization)

- 인증(Authentication)


- 사용자의 신원 입증
- ex. 특정 사이트에 로그인할 때 누구인지 확인하는 절차를 거침

- 인가(Authorization)

- 인증된 사용자가 요청된 자원에 접근 가능한지 권한 확인
- ex. 일반 유저는 관리자 페이지에 접속할 수 없음

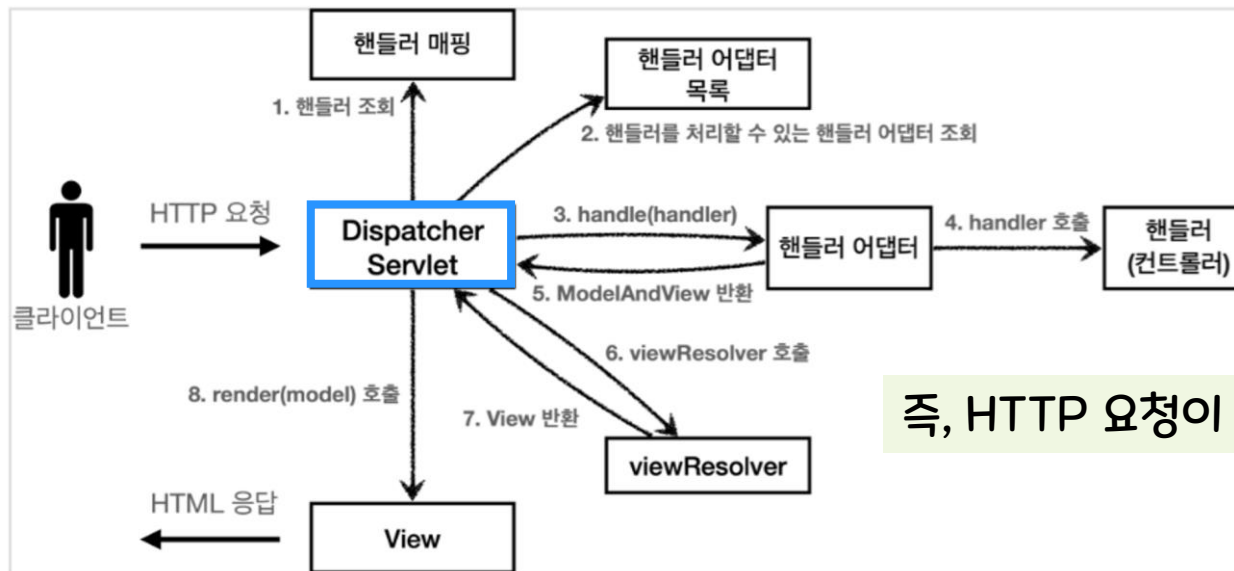
“Spring Security 를 사용하면 인증, 인가를 쉽게 구현할 수 있다!”

# Spring Security

- 스프링 기반의 애플리케이션 보안(인증, 인가)을 담당하는 스프링 하위 프레임워크
- 프레임워크의 일종으로 반드시 필요한 것은 아님
  - 하지만 쓰는 것이 편하다.. 그렇지 않으면 개발자가 보안 로직을 직접 개발해야 한다.. 
- 다양한 보안 관련 옵션 제공
- 특징
  1. **Filter**를 기반으로 동작  
(Filter는 Dispatcher Servlet으로 가기 전에 적용되므로 가장 먼저 요청을 받음! 다음 장에 이어서..)
  2. **Bean**으로 설정 가능

# 복습. Spring MVC

- Spring 에서 제공하는 웹 모듈로 웹 애플리케이션을 빌드하기 위한 프레임워크
- Model, View, Controller 의 3가지 구성 요소
- MVC 패턴을 지원하고 **DispatcherServlet** 이라는 특수 서블릿을 통해 요청 처리
  - 이 서블릿은 모든 요청을 받아 적절한 Controller 에게 전달하고 그 결과를 다시 사용자에게 반환



즉, HTTP 요청이 **Dispatcher Servlet**을 거쳐 Controller로 전달된다!

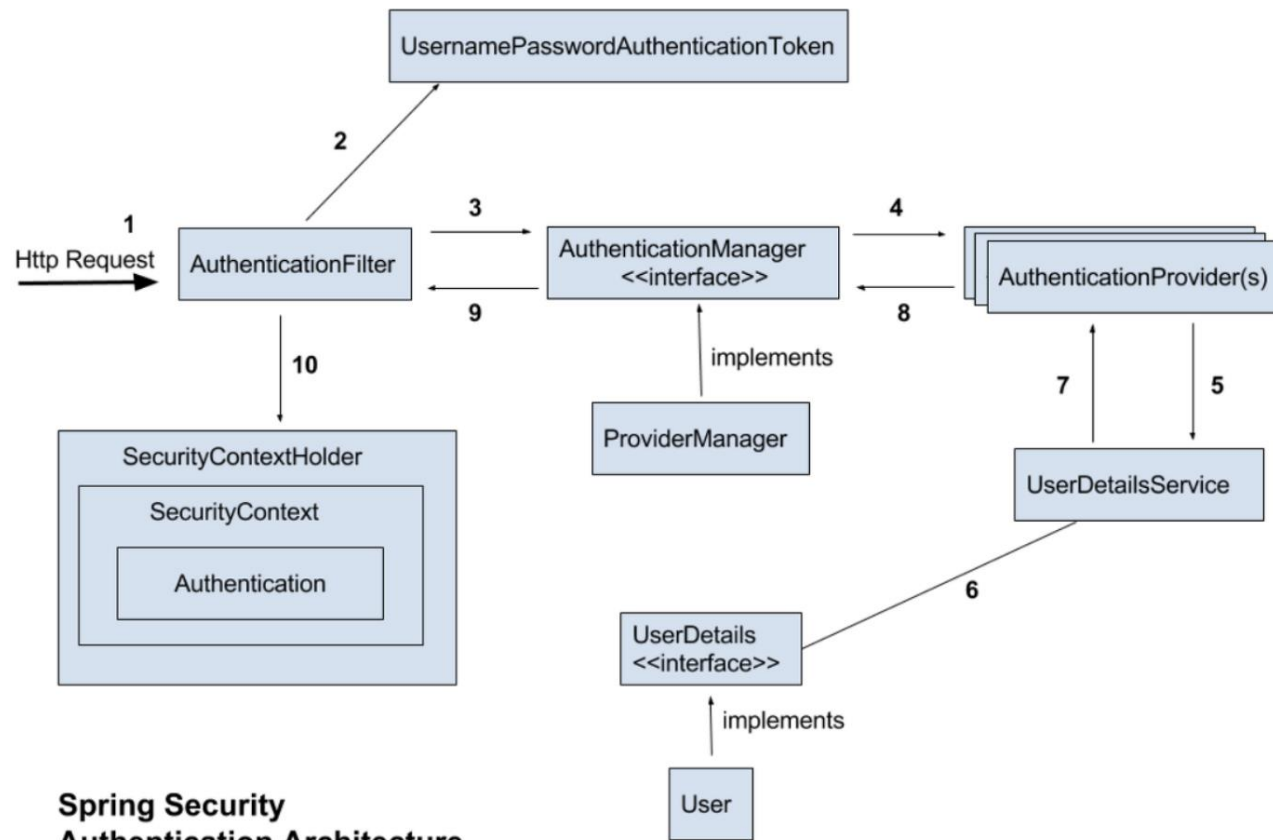
# Spring Security 동작 과정

- Spring Security 는 요청이 **Dispatcher Servlet** 으로 가기 전에 **Filter** 단에서 **인증/인가 처리**를 진행



# Spring Security Architecture

Q. 아이디, 비밀번호 기반 폼 로그인 시도 시 Spring Security 는 어떻게 인증 처리를 할까?

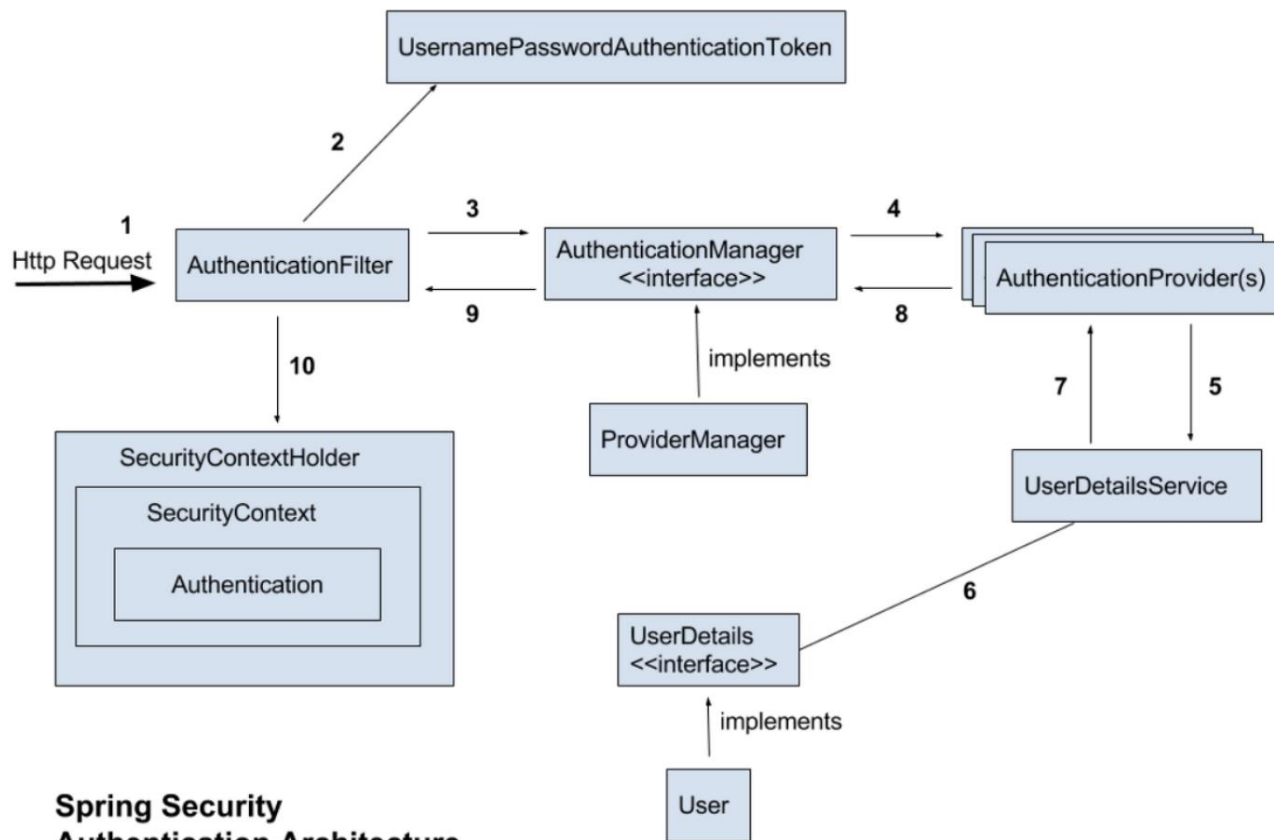


**Spring Security  
Authentication Architecture**

Chathuranga Tennakoon  
[www.springbootdev.com](http://www.springbootdev.com)

# Spring Security Architecture

Q. 아이디, 패스워드 기반 폼 로그인 시도 시 Spring Security 는 어떻게 인증 처리를 할까?



## 1. HTTP Request 수신

- 사용자의 로그인 정보(id, pw) 입력하면, HttpServletRequest 에 id, pw 가 전달됨
- AuthenticationFilter 가 넘어온 아이디와 비밀번호 유효성 검사 진행

## 2. 유효성 검사가 끝나면 인증용 객체인

UsernamePasswordAuthenticationToken를 생성

## 3. UsernamePasswordAuthenticationToken 을 AuthenticationManager 에게 보냄

## 4. UsernamePasswordAuthenticationToken 을 AuthenticationProvider 에 보냄

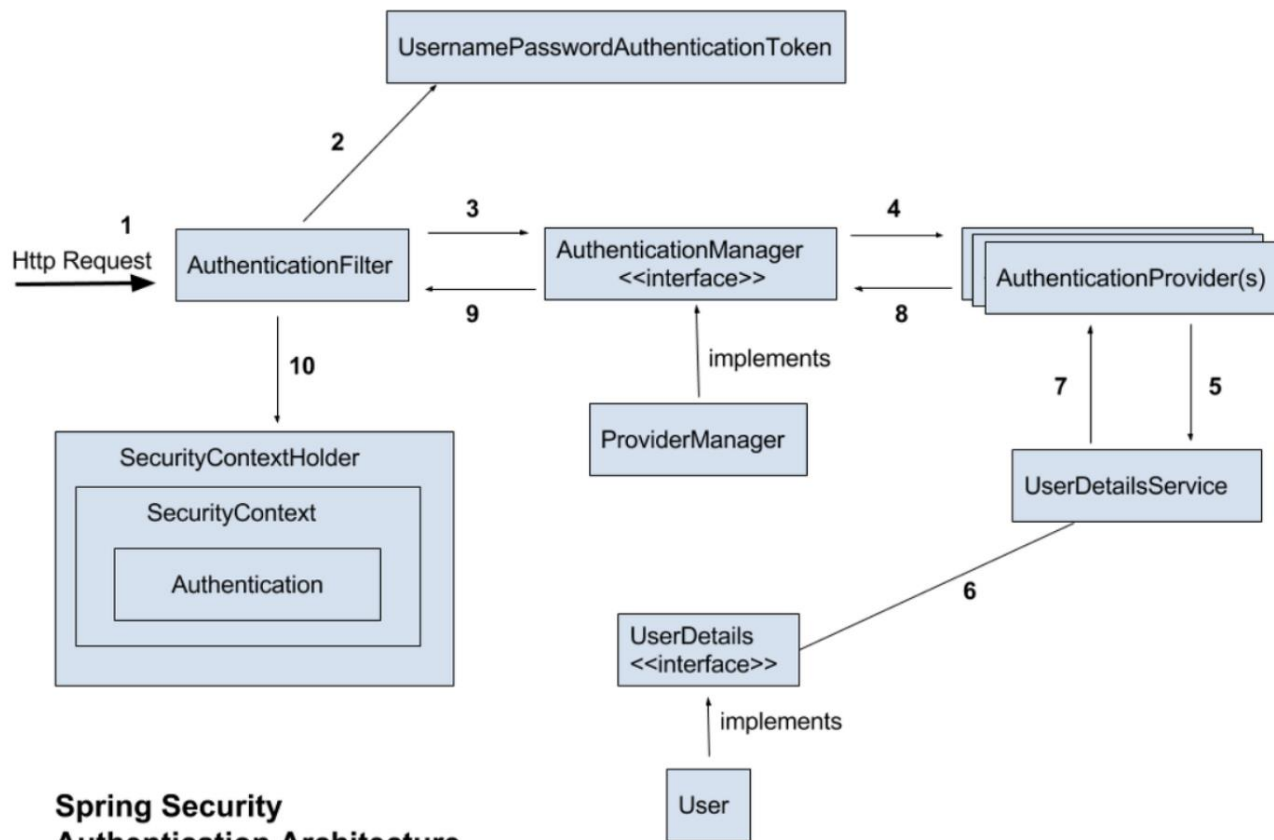
**Spring Security  
Authentication Architecture**

Chathuranga Tennakoon  
[www.springbootdev.com](http://www.springbootdev.com)



# Spring Security Architecture

Q. 아이디, 패스워드 기반 폼 로그인 시도 시 Spring Security 는 어떻게 인증 처리를 할까?



5. 사용자 아이디를 UserDetailsService 에 보냄
6. DB 에 있는 사용자 정보를 가져와 찾아낸 사용자 정보를 UserDetails 객체로 만들
7. 입력된 사용자 정보와 UserDetails 정보를 비교 해 실제 인증 처리
8. 인증 완료 여부
  - 인증 완료: 사용자 정보를 담은 Authentication 객체 반환
  - 인증 실패: AuthenticationException 발생
9. Authentication 객체가 SecurityContextHolder 에 저장됨

**Spring Security  
Authentication Architecture**

Chathuranga Tennakoon  
[www.springbootdev.com](http://www.springbootdev.com)

# Spring Security 사용하기

- build.gradle 에 dependencies 추가

```
implementation 'org.springframework.boot:spring-boot-starter-security'
```

- 참고. Thymeleaf 에서 Spring Security 를 사용한다면?

```
implementation 'org.thymeleaf.extras:thymeleaf-extras-springsecurity6'
```

# Spring Security 사용하기

1. config/WebSecurityConfig 파일 생성
  - security 세부 설정을 진행함
2. Custom Filter 생성
  - 인증 방식에 따라 상황에 맞는 로직 생성 (session, jwt token 등)
3. 컨트롤러에서 로그인 된 유저 정보 가져오기?