

簡介說明



spring **SECURITY**

2022-02 段維瀚老師



Java 程式設計

JavaWeb
程式設計

資料庫設計

T-SQL / PL-SQL 語法
鍵值與索引的建立
資料表正規化
程式如何與資料庫系統互動

職前訓練

SpringFramework

SpringBoot

成為企業優質
Java開發軟體人才
必備技能

SpringSecurity

認證 (Authentication)
授權 (Authorization)
開放授權協議 (OAuth2)
JWT (Json Web Token)
SSO (Single-Sign-On)

SpringBatch

Job 與 Step
JobParameters
JobExecution
JobRepository
JobLauncher
chunk 處理流程
Item Reader
Item Writer
Item Processor

SpringCloud

服務發現 (Eureka)
斷路器 (Hystrix)
智能路由 (Zuul)
負載平衡 (Ribbon)
Spring Cloud Getway
Hashicorp Vault 平台來管理機密和保護敏感數據

在職進修

安全

Security

「驗證」與「授權」

安全框架

Security Framework

安全框架

- 解決系統安全問題的框架

- 框架包含

- 規範統一的系統運作流程
 - 一堆成品與「半」成品的實作

- 為何要使用框架

- 業務流程可以被規範、減少編寫大量重複程式碼的工作。

安全框架

- 常見的安全框架

- (前身 JSecurity) **Apache Shiro**
- 2003開始 (前身 Acegi) 2007正名 **Spring Security**

安全框架組合

- SSH / SSM
 - Apache Shiro
- Spring Boot / Spring Cloud
 - Spring Security

Spring Security 驗證與授權

- 使用者名稱 / 密碼登入
 - SessionId 與 cookie
 - remember-me
- Token
 - OAuth2 架構
 - JWT (Json Web Token) 格式
 - SSO (Single Sign-On) 單點登入應用

Spring Security 語法特色

- 透過聲明式語法
 - 實現安全認證
 - 例如：驗證/授權(角色/權限)

聲明式語法

```
// 表單提交
http.formLogin()
// loginpage.html 表單的 action 值
.loginProcessingUrl("/login")
// 自定義登入頁面
.loginPage("/loginpage")
// 登入成功後的頁面
.successForwardUrl("/")
// 登入失敗後的頁面
.failureForwardUrl("/fail");
```

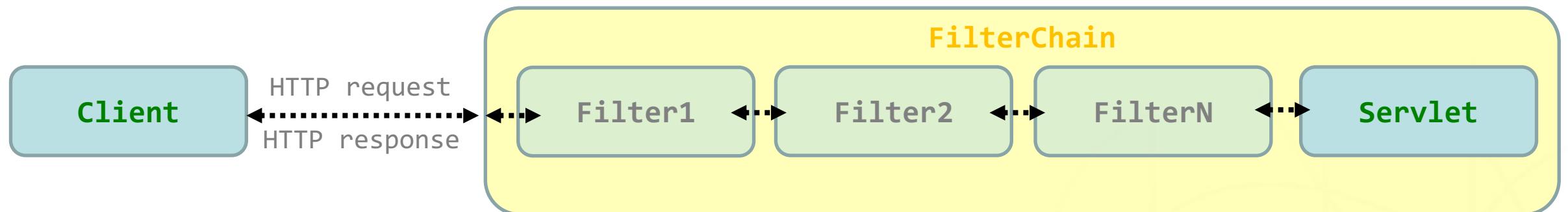
驗證/授權(角色/權限)

```
// 授權認證
http.authorizeHttpRequests()
// /loginpage 不需要被認證
.antMatchers("/loginpage").permitAll() // 使用 ant patterns
// 權限判斷
// 必須要有 admin 權限才能訪問 /adminpage
.antMatchers("/adminpage").hasAuthority("admin")
// 角色判斷
.antMatchers("/managerpage").hasRole("manager") // 也可使用 @Secured("ROLE_manager")
.antMatchers("/employeepage").hasAnyRole("manager", "employee")
// 其他所有請求都必須被認證
.anyRequest().authenticated();
```

JavaWeb Servlet 運作流程



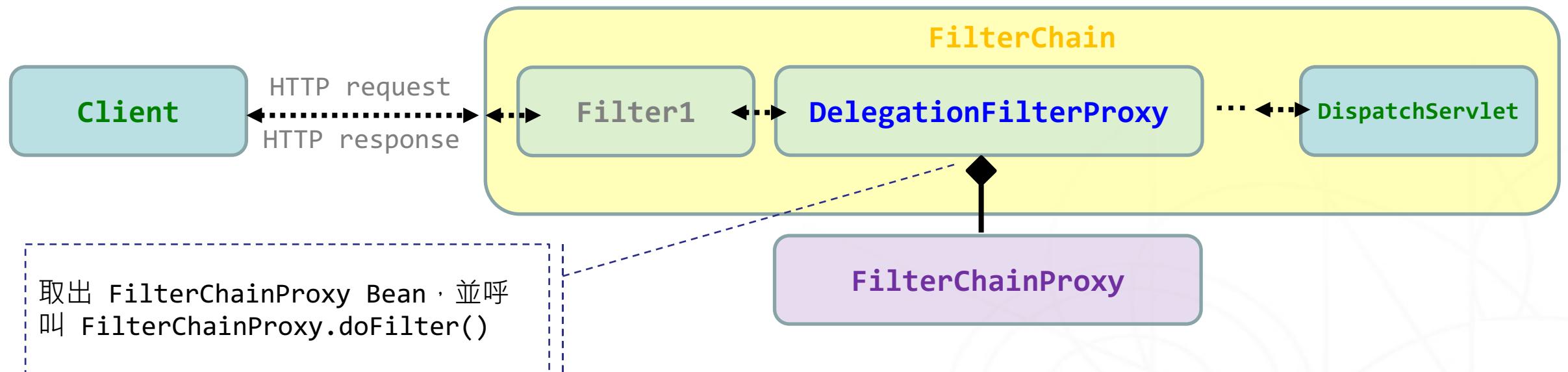
JavaWeb Servlet 運作流程



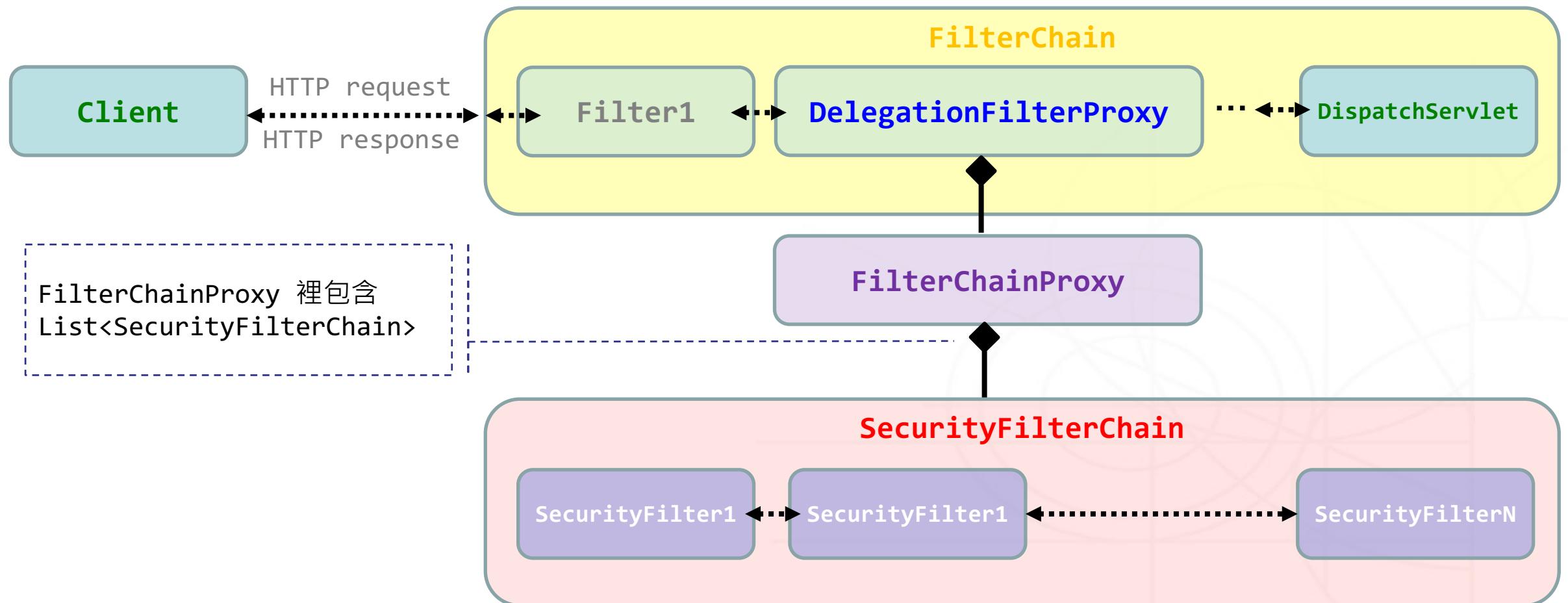
Spring Web 運作流程



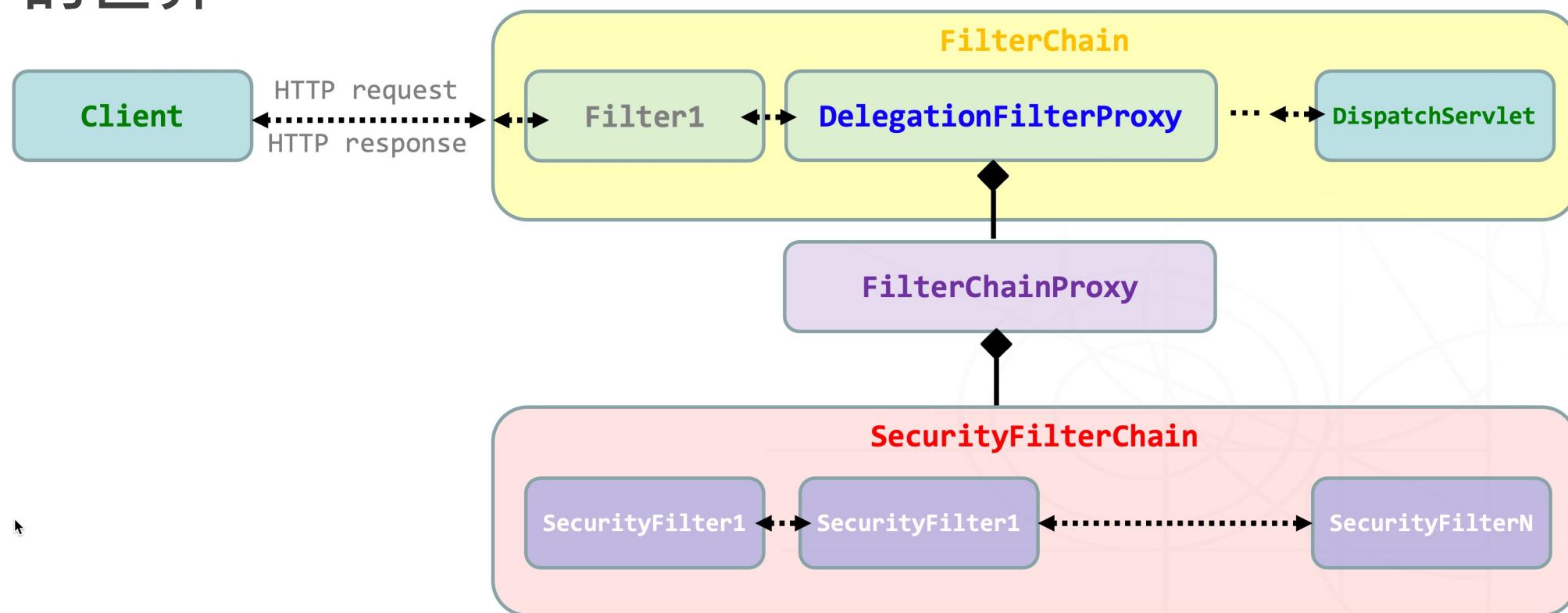
Spring Web Security 運作流程



Spring Web Security 運作流程



- **DelegatingFilterProxy** 可以說是，從 Servlet 的世界正式進入 Spring 的世界。而，**FilterChainProxy** 可以說是，從 Spring 的世界進入 Spring Security 的世界。



Spring Web Security 運作流程

- Spring Security > Architecture
 - <https://docs.spring.io/spring-security/reference/servlet/architecture.html>

實作練習



spring
SECURITY

2022-02 段維瀚老師

本次課程實作

- 使用者名稱 / 密碼登入
 - remember-me (session失效後仍可保持登入)
- 授權/角色

實作前...

- **UserDetialsService**

- 使用者服務（驗證）介面 / 實現登入邏輯
 - 需要自行實作

- **UserDetials**

- 使用者資訊介面

- 實作：

- 預設 User 實作了 UserDetials 介面以儲存使用者資訊，也可自行實作。

- **PasswordEncoder**

- 官方推薦實作：**BCryptPasswordEncoder()**

User 資料表資訊

username	password	authority
A01	\$2a\$10\$fSqGtmR2YEgyNtw6eZS0yu1N1.SipyoZnDrkOFh4qQGkkpaqIuwMu	admin,normal, ROLE_manager
A02	\$2a\$10\$aCRmt7w925GBTX.s0YAh2e.TyBzeFvUD6x1JxCLwPvXzABz7ev3Zm	normal, ROLE_employee

1234

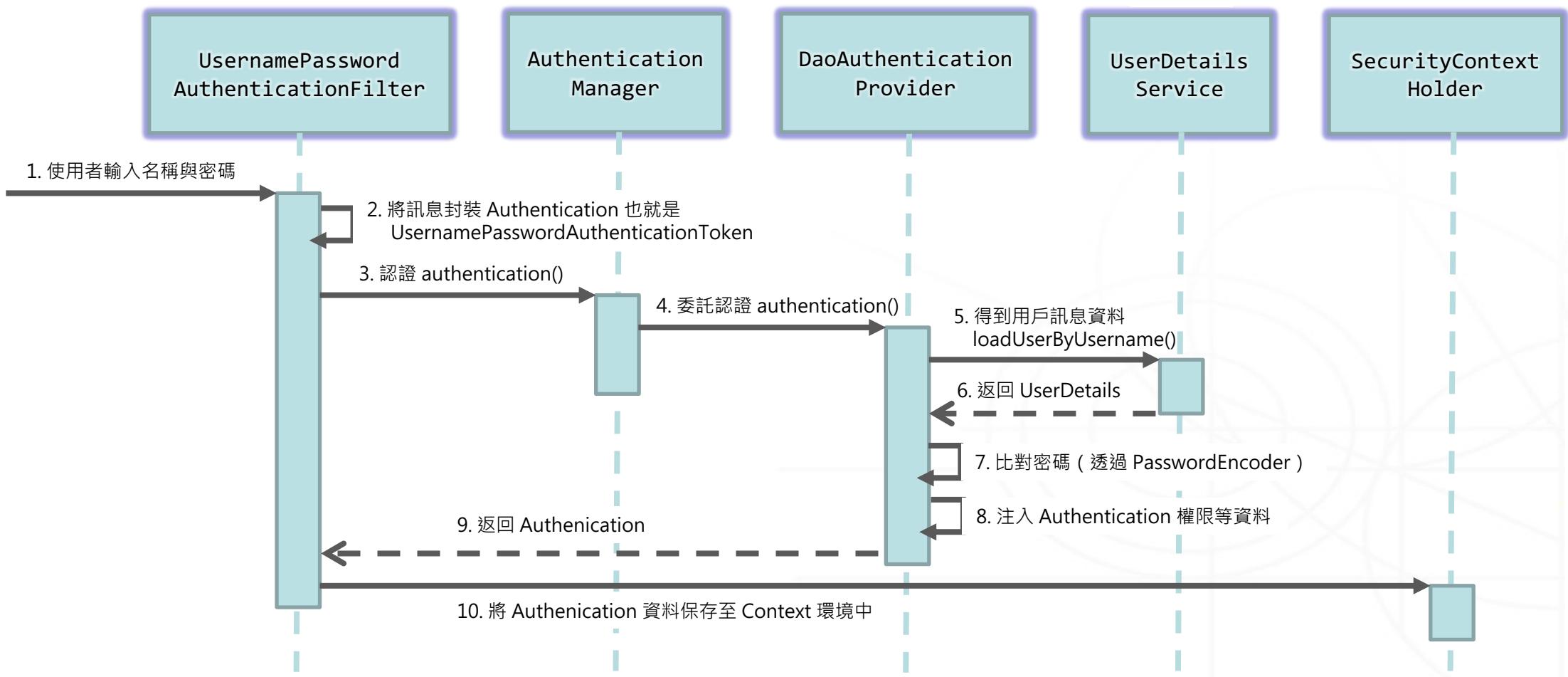
ROLE_ 固定要加入

5678

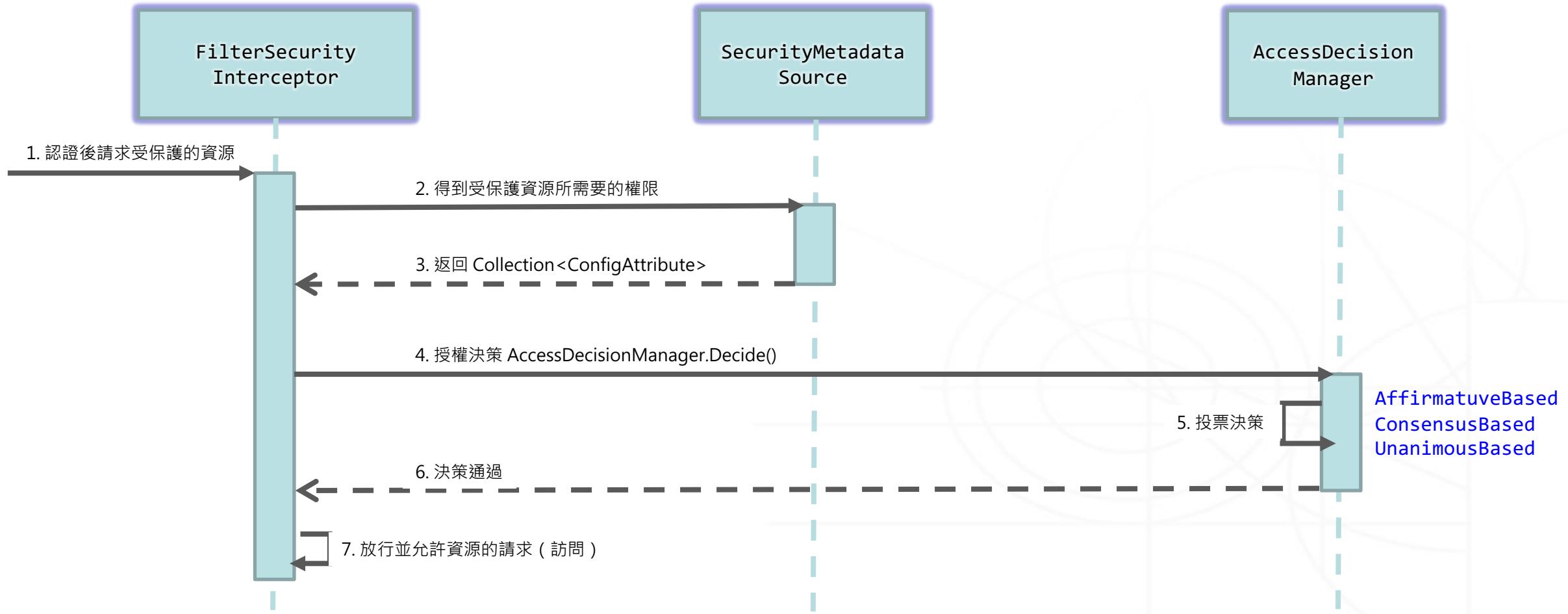
實作練習

- 請建立一個 **SpringBoot** 專案
 - 增加依賴 - **Web**
 - 增加依賴 - **Thymeleaf**
 - 增加依賴 - **SpringSecurity**

Spring Security 認證流程



Spring Security 授權流程



接下來你還要繼續學習的是 . . .

- 聲明式語法
- 安全認證
 - 驗證/授權(角色/權限)
 - 第三方登入 OAuth2
- SpringSecurityOAuth2 架構
 - 自行搭建：授權服務器/發送Token/資源服務器配置
- SpringSecurityOAuth2 整合 JWT (JSON Web Tokens)
- SpringSecurityOAuth2 整合 SSO (Single Sign-On)