15-1 驗證與授權應用

第十五單元大綱

15-1　驗證與授權應用

15-2　Spring Filter應用

15-3　REST CORS設計應用

15-4　單元測驗

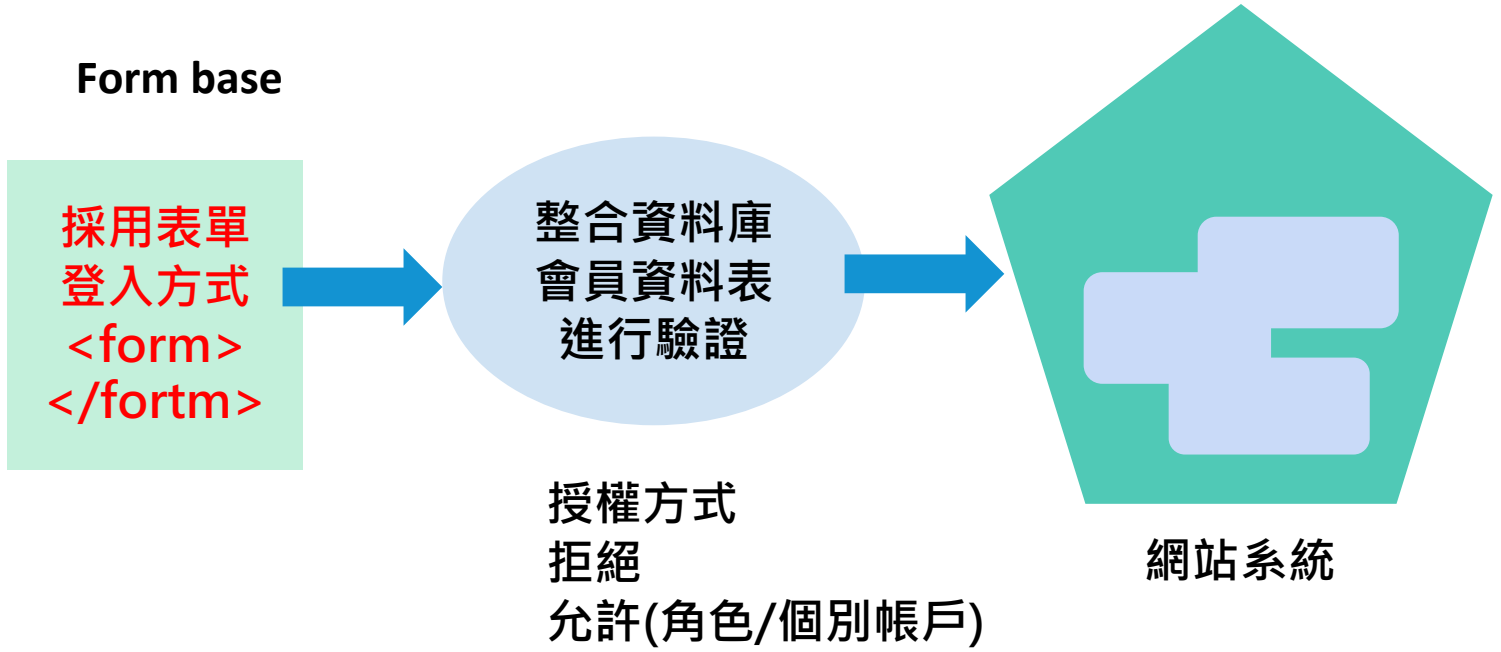
驗證與授權(authentication vs authorization)

驗證模式

- 驗證方式的決定，如採用Form base配合會員資料表，或者是採用Windows整合帳號驗證，或者是採用基本驗證模式等

授權

- 可以採用角色(群組)或者個人識別帳號進行授權
- 授權為可進入這一個功能位址等



Spring Boot啟動安全性驗證

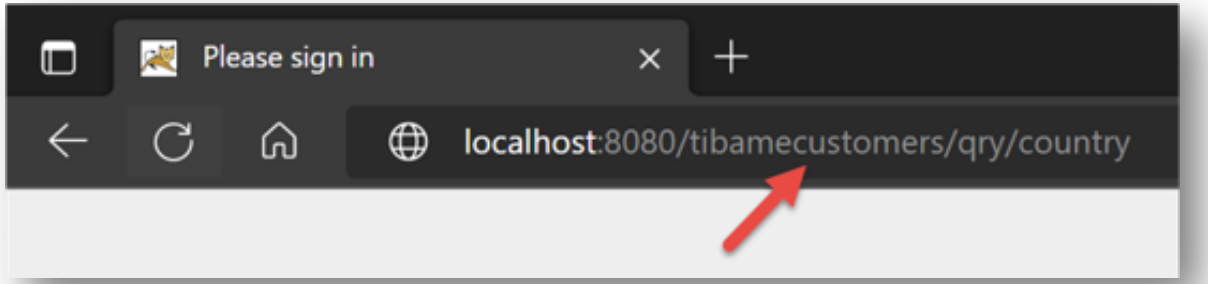
在pom.xml配置

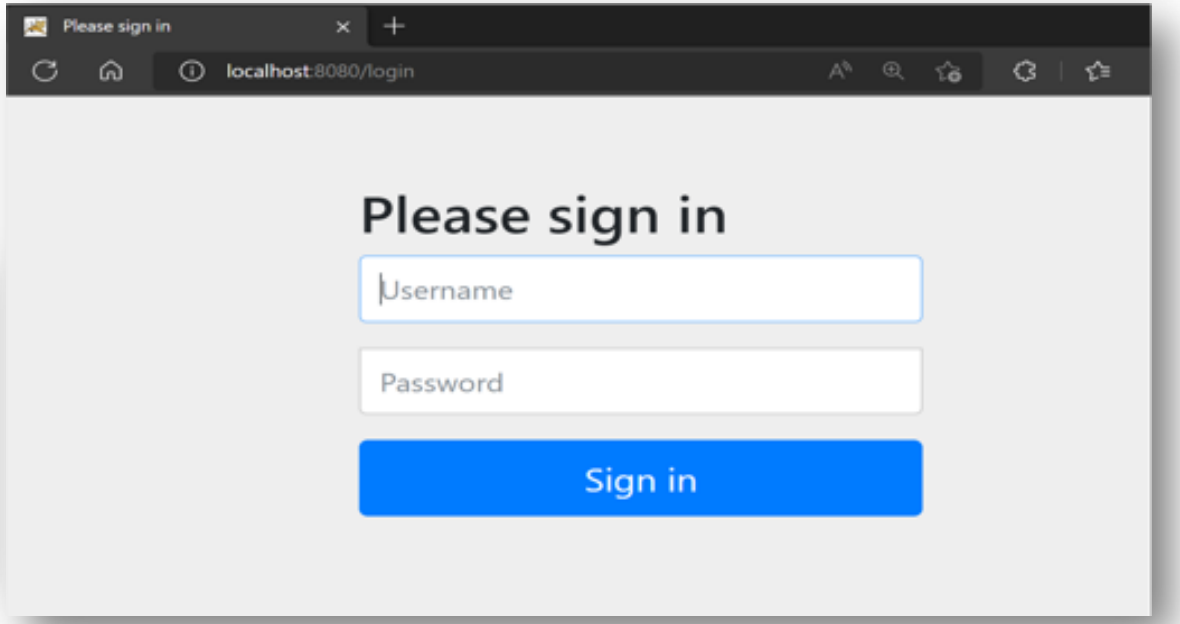
- spring-boot-starter-security

即刻啟動整個網站Form base安全驗證機制

未經驗證通過，直接調用至預設的登入頁面

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
    <version>2.7.3</version>
</dependency>
```





配置驗證模式與登入頁面

規劃一個POJO類別

使用@EnableWebSecurity進行網站安全性配置

使用@EnableGlobalMethodSecurity

- securedEnabled設定是否啟用@Secured
- prePostEnabled設定是否啟用@PreAuthorize
- jsr250Enabled設定是否啟用@RolesAllowed

配置SecurityFilterChain Bean，注入HttpSecurity進行驗證模式與授權模式設定

WebSecurityConfig.java

```
12
13 @EnableWebSecurity
14 @EnableGlobalMethodSecurity(prePostEnabled=true,securedEnabled = true, jsr250Enabled = true)
15 public class WebSecurityConfig {
16
```

```
//透過方法生產Spring Bean到Spring Container進行配置
@Bean
public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
    //進行安全性for Http 配置...
    http.csrf().disable();
    http.formLogin()
    .loginPage("/login")
    .loginProcessingUrl("/process-login");
```

Spring Boot Web MVC

MVC架構規劃一個login path

規劃一個MemberController

設計端點調用登入頁面

```java
@Controller
public class MemberController {

    //調用註冊表單
    @GetMapping(path="/member/register")
    public String register() {
        return "memberform";
    }

    //登入頁面
    @GetMapping(path="/login")
    public String loginForm() {
        return "loginform";
    }
```

允許網站匿名存取架構

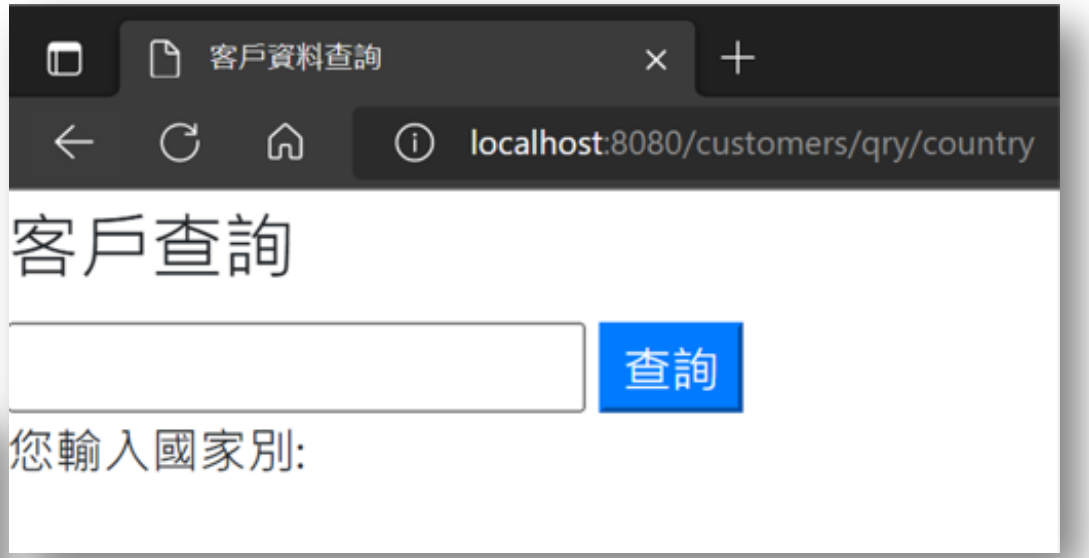
使用HttpSecurity authorizeHttpRequests()啟動請求須經安全性驗證

配置anyRequest().permitAll()

允許匿名存取

```
@Bean
public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
    //進行安全性for Http 配置...
    http.csrf().disable();
    http.formLogin()
    .loginPage("/login")
    .loginProcessingUrl("/process-login")

    //使用Form base(結合登入頁面與資料庫會員) Authentication驗證模式
    //授權模式
    .and()
    .authorizeHttpRequests().anyRequest().permitAll(); //任何請求需要被驗證與授權
```



Spring Boot Web MVC

設定授權模式，允許匿名與非匿名存取path

使用 antMatchers()配置允許的path

- /path/* 表示此目錄下
- /path/** 表示此目錄下任盒子目錄

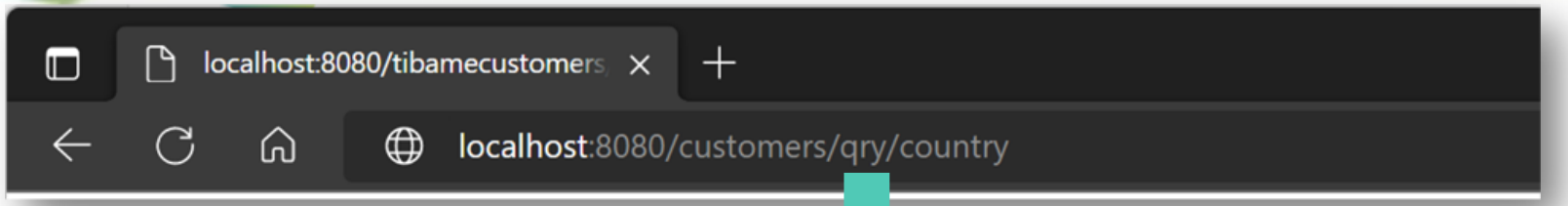
使用hasAnyRole()進行角色授權

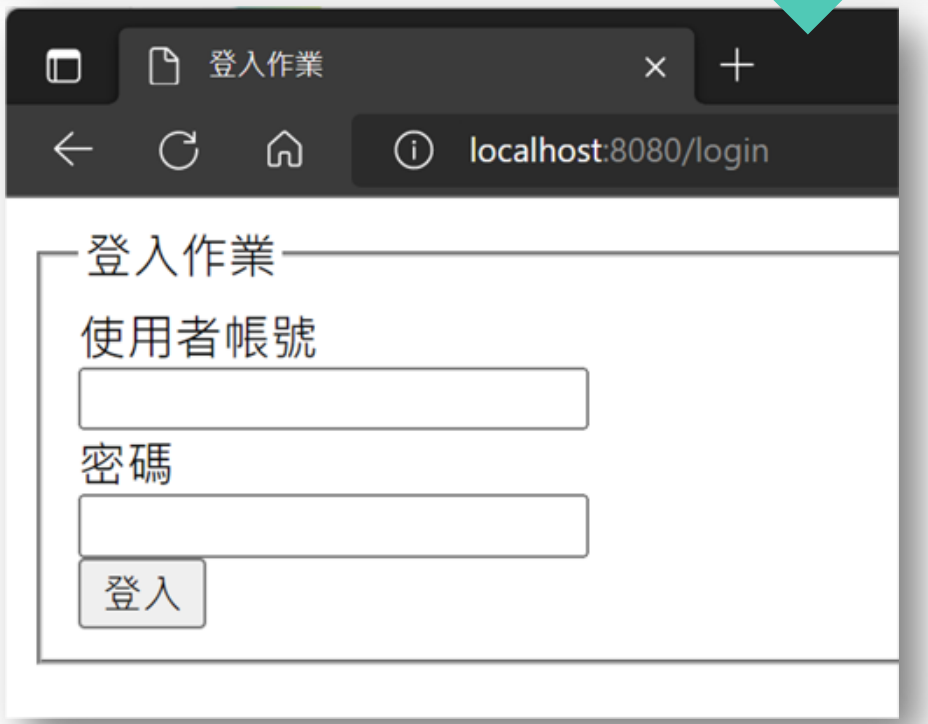
permitAll()設定允許存取權限

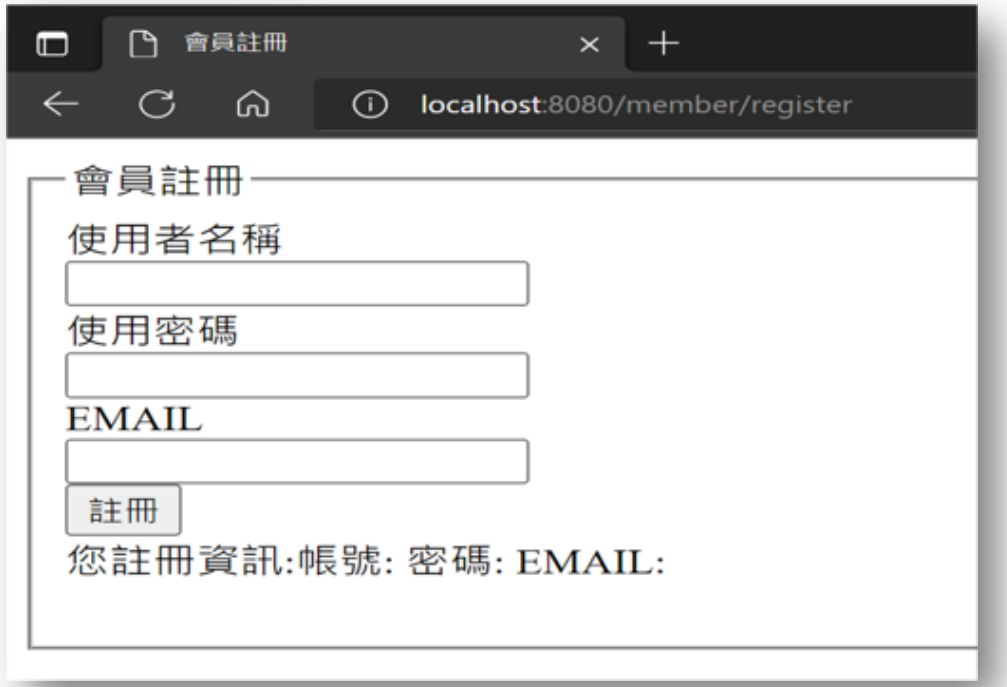
```
@Bean
public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
    //進行安全性for Http 配置...
    http.csrf().disable();
    http.formLogin()
    .loginPage("/login")
    .loginProcessingUrl("/process-login")

    //使用Form base(結合登入頁面與資料庫會員) Authentication驗證模式
    //授權模式
    .and()
    .authorizeHttpRequests() //任何請求需要被驗證與授權
    .antMatchers("/login","/login.html","/member/register","/","/api/member/register"
            ,"/principal","/getprincipal").permitAll()
    .antMatchers("/customers/**").hasAnyRole("IT", "ADMIN").anyRequest().permitAll();
```

設定授權模式登入


localhost:8080/tibamecustomers
localhost:8080/customers/qry/country


登入作業
localhost:8080/login
登入作業
使用者帳號
密碼
登入


會員註冊
localhost:8080/member/register
會員註冊
使用者名稱
使用密碼
EMAIL
註冊
您註冊資訊:帳號: 密碼: EMAIL:

總結：15-1 驗證與授權應用

學習到如何設定Spring Boot MVC網站安全性之後，接下來我們來看看如何自訂login登入頁面整合會員資料表驗證，與發出許可證進行通行。