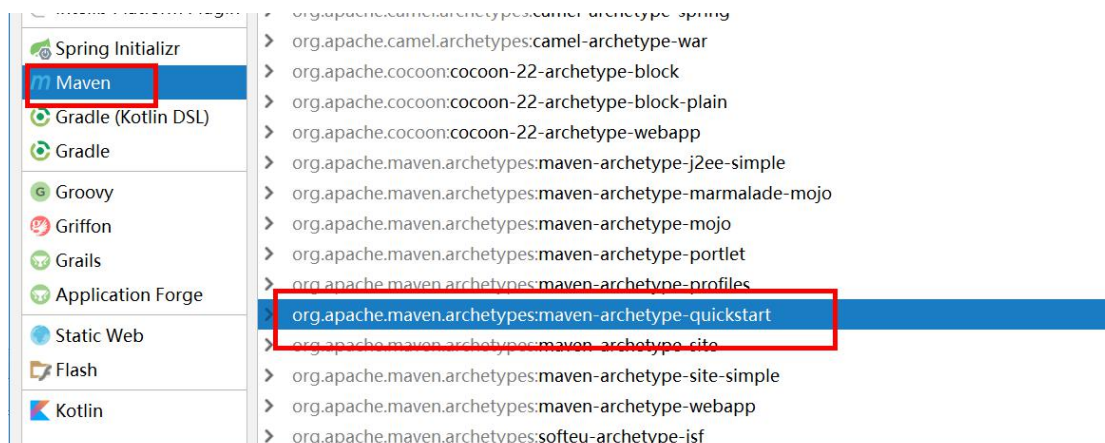


学习目标

- 快速搭建 Spring Boot 运行环境
- 整合 Spring Security - HttpBasic 权限实现
- 整合 Spring Security - FormLogin 权限实现
- 自定义登录页面，自定义用户权限判断

1. 快速搭建 Spring Boot 运行环境

1.1. 建立 maven 项目，导入坐标



```
<?xml version="1.0" encoding="UTF-8" ?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>cn.sm1234</groupId>

  <artifactId>03_spring_security</artifactId>

  <version>1.0-SNAPSHOT</version>
```

```
<!-- Spring Boot 父工程 -->

<parent>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-parent</artifactId>

    <version>1.5.4.RELEASE</version>

</parent>

<dependencies>

    <!-- web 支持, SpringMVC, Servlet 支持等 -->

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-web</artifactId>

    </dependency>

    <!-- thymeleaf -->

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-thymeleaf</artifactId>

    </dependency>

</dependencies>

<properties>

    <java.version>1.9</java.version>

    <!-- 修改 thymeleaf 的版本 -->

    <thymeleaf.version>3.0.2.RELEASE</thymeleaf.version>

    <thymeleaf-layout-dialect.version>2.0.4</thymeleaf-layout-dialect
```

```
.version>  
  
    </properties>  
  
</project>
```

1.2. 编写 SpringBoot 启动类

```
package cn.sm1234;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
/**  
 * @author http://www.sm1234.cn  
 * @version 1.0  
 * @description cn.sm1234  
 * @date 18/4/16  
 */  
  
@SpringBootApplication  
public class SpringSecurityApp {  
  
    public static void main(String[] args) {  
  
        SpringApplication.run(SpringSecurityApp.class, args);  
  
    }  
  
}
```

2. 准备访问的资源

2.1. 编写 Controller

```
package cn.sm1234.controller;

import org.springframework.stereotype.Controller;
import
org.springframework.web.bind.annotation.RequestMapping;

/**
 * @author http://www.sm1234.cn
 * @version 1.0
 * @description cn.sm1234.controller
 * @date 18/4/16
 */
@Controller
public class MainController {

    @RequestMapping("index")
    public String index() {
        return "index";
    }
}
```

```
package cn.sm1234.controller;

import org.springframework.stereotype.Controller;
```

```
import
org.springframework.web.bind.annotation.RequestMapping;

/**
 * @author http://www.sm1234.cn
 * @version 1.0
 * @description cn.sm1234.controller
 * @date 18/4/16
 */
@Controller
@RequestMapping("product")
public class ProductController {

    @RequestMapping("add")
    public String add(){
        return "product/add";
    }

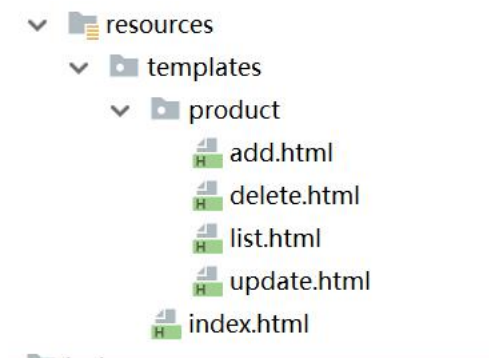
    @RequestMapping("update")
    public String update(){
        return "product/update";
    }

    @RequestMapping("delete")
    public String delete(){
        return "product/delete";
    }

    @RequestMapping("list")
```

```
public String list(){  
    return "product/list";  
}  
}
```

2.2. 编写页面



3. 整合 Spring Security - HttpBasic 权限实现

3.1. 导入 SpringSecurity 坐标

```
<!-- Spring Security -->  
<dependency>  
    <groupId>org.springframework.security</groupId>  
    <artifactId>spring-security-web</artifactId>  
    <version>4.2.3.RELEASE</version>  
</dependency>  
  
<dependency>  
    <groupId>org.springframework.security</groupId>  
    <artifactId>spring-security-config</artifactId>
```

```
<version>4.2.3.RELEASE</version>  
</dependency>
```

3.2. 编写 SpringSecurityConfig 配置类

```
package cn.sm1234.security;  
  
import org.springframework.context.annotation.Configuration;  
  
import  
org.springframework.security.config.annotation.authentication.builders.Authe  
nticationManagerBuilder;  
  
import  
org.springframework.security.config.annotation.web.builders.HttpSecurity;  
  
import  
org.springframework.security.config.annotation.web.configuration.EnableWebSe  
curity;  
  
import  
org.springframework.security.config.annotation.web.configuration.WebSecurity  
ConfigurerAdapter;  
  
/**  
 * @author http://www.sm1234.cn  
 * @version 1.0  
 * @description cn.sm1234.security  
 * @date 18/4/16  
 */  
  
@Configuration  
@EnableWebSecurity //启动 SpringSecurity 过滤器链
```

```
public class SpringSecurityConfig extends WebSecurityConfigurerAdapter {

    //该方法的作用就是代替之前配置: <security:authentication-manager>

    @Override

    protected void configure(AuthenticationManagerBuilder auth) throws Exception
    {

        auth.inMemoryAuthentication().withUser("eric").password("123456").authorities("PRODUCT_ADD");

    }

    //该方法的作用就是代替之前配置: <security:http>

    @Override

    protected void configure(HttpSecurity http) throws Exception {

        http.authorizeRequests()

            .antMatchers("/**")

            .fullyAuthenticated()

            .and()

            .httpBasic()

        ;

    }

}
```

4. 整合 Spring Security - FormLogin 权限实现

改为 formLogin:

```
//该方法的作用就是代替之前配置: <security:http>

@Override
```



```
protected void configure (HttpSecurity http) throws Exception
{
    http.authorizeRequests()
        .antMatchers ("/**")
        .fullyAuthenticated()
        .and()
        .formLogin();
    ;
}
```