[Thực hành] Đếm số lượt view trang

Mục tiêu

Luyện tập triển khai cơ chế session trong Spring MVC.

Mô tả

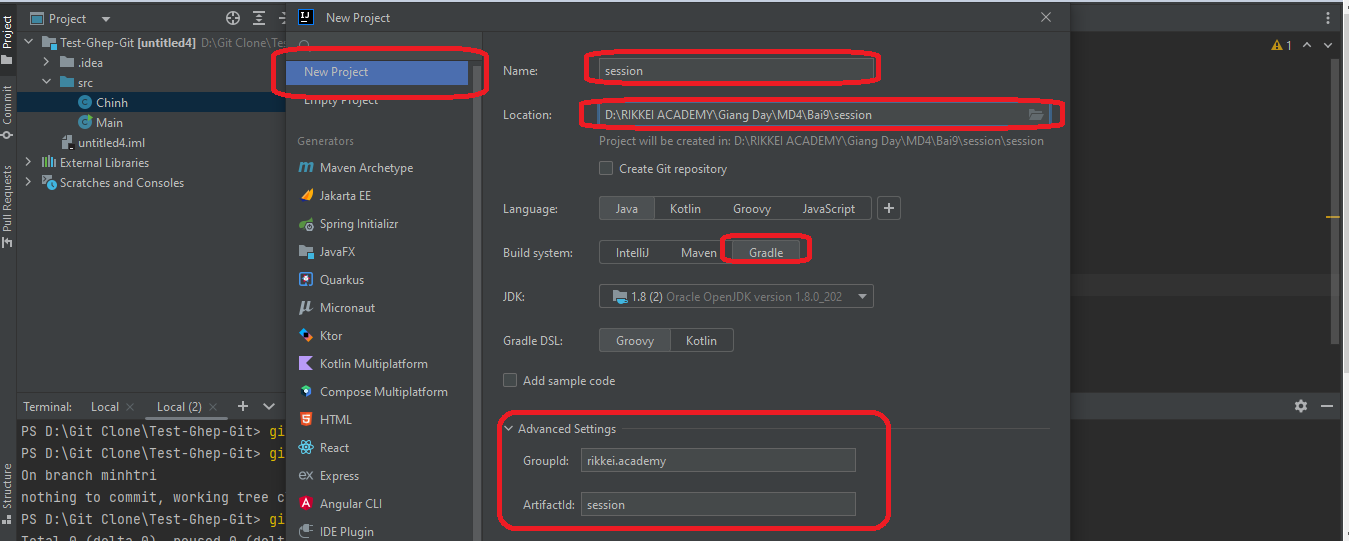
Trong phần này, chúng ta sẽ phát triển một tính năng cho phép ghi nhận số lần người dùng truy cập trang web trong một phiên làm việc.

Ứng dụng sẽ ghi nhận và hiển thị số lần truy cập. Khi người dùng truy cập vào một trang trong ứng dụng thì số lần truy cập sẽ tăng lên.

**Để hoàn thành bài thực hành, học viên cần:**

* Đưa mã nguồn lên GitHub

Hướng dẫn



* Thêm thư viện vào **build.gradle:**

compileOnly('javax.servlet:javax.servlet-api:4.0.1')  
implementation group: 'org.springframework', name: 'spring-webmvc', version: '5.3.2'  
implementation group: 'org.thymeleaf', name: 'thymeleaf-spring5', version: '3.0.11.RELEASE'

* Tạo package: **rikkei.academy.config** => Tạo class: **AppConfig**:

package rikkei.academy.config;  
  
import org.springframework.beans.BeansException;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.ApplicationContextAware;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.ComponentScan;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.web.servlet.config.annotation.EnableWebMvc;  
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;  
import org.thymeleaf.spring5.SpringTemplateEngine;  
import org.thymeleaf.spring5.templateresolver.SpringResourceTemplateResolver;  
import org.thymeleaf.spring5.view.ThymeleafViewResolver;  
import org.thymeleaf.templatemode.TemplateMode;  
@Configuration  
@EnableWebMvc  
@ComponentScan("rikkei.academy.controller")  
  
public class AppConfig implements WebMvcConfigurer, ApplicationContextAware {  
 private ApplicationContext applicationContext;  
  
 @Override  
 public void setApplicationContext(ApplicationContext applicationContext) throws BeansException {  
 this.applicationContext = applicationContext;  
 }  
  
 //Cấu hình Thymleaf  
 @Bean  
 public SpringResourceTemplateResolver templateResolver() {  
 SpringResourceTemplateResolver templateResolver = new SpringResourceTemplateResolver();  
 templateResolver.setApplicationContext(applicationContext);  
 templateResolver.setPrefix("/WEB-INF/views");  
 templateResolver.setSuffix(".html");  
 templateResolver.setTemplateMode(TemplateMode.*HTML*);  
 templateResolver.setCharacterEncoding("UTF-8");  
 return templateResolver;  
 }  
  
 @Bean  
 public SpringTemplateEngine templateEngine() {  
 SpringTemplateEngine templateEngine = new SpringTemplateEngine();  
 templateEngine.setTemplateResolver(templateResolver());  
 return templateEngine;  
 }  
  
 @Bean  
 public ThymeleafViewResolver viewResolver() {  
 ThymeleafViewResolver viewResolver = new ThymeleafViewResolver();  
 viewResolver.setTemplateEngine(templateEngine());  
 viewResolver.setCharacterEncoding("UTF-8");  
 return viewResolver;  
 }  
  
}

* Tạo class: **AppInit** trong package: **config**

package rikkei.academy.config;  
  
import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;  
  
public class AppInit extends AbstractAnnotationConfigDispatcherServletInitializer {  
 @Override  
 protected Class<?>[] getRootConfigClasses() {  
 return new Class[]{AppConfig.class};  
 }  
  
 @Override  
 protected Class<?>[] getServletConfigClasses() {  
 return new Class[0];  
 }  
  
 @Override  
 protected String[] getServletMappings() {  
 return new String[]{"/"};  
 }  
}

* Tạo package: model => Tạo class: **Counter:**

package rikkei.academy.model;  
  
public class Counter {  
 private int count;  
  
 public Counter() {  
 }  
  
 public Counter(int count) {  
 this.count = count;  
 }  
  
 public int getCount() {  
 return count;  
 }  
  
 public void setCount(int count) {  
 this.count = count;  
 }  
  
 public int increment() {  
 return count++;  
 }  
  
}

Trong lớp Counter chúng ta khai báo một biến count để tăng lượt view sau mỗi lần người dùng truy cập trang và một phương thức increment() để thi hành việc tăng giá trị cho biến count.

* Tạp package: controller trong package: rikkei.academy => Tạo class: **CounterController**

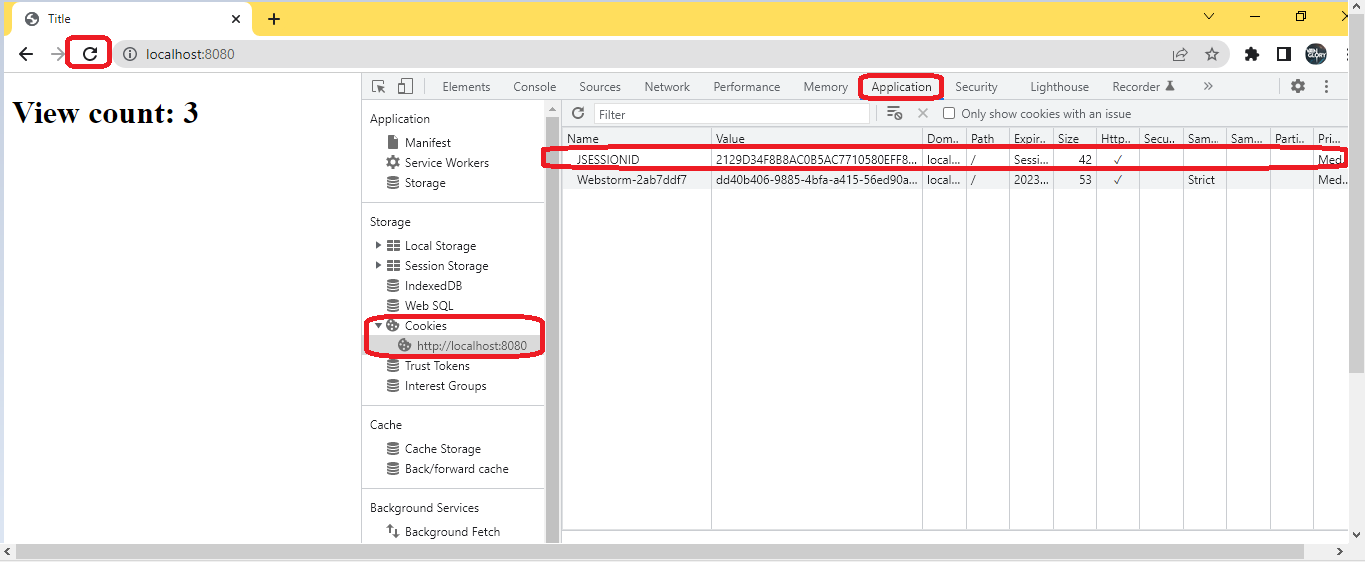
package rikkei.academy.controller;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.ModelAttribute;  
import org.springframework.web.bind.annotation.SessionAttributes;  
import rikkei.academy.model.Counter;  
@Controller  
@SessionAttributes("counter")  
public class CounterController {  
 @ModelAttribute("counter")  
 public Counter setUpCounter() {  
 return new Counter();  
 }  
  
 @GetMapping("/")  
 public String get(@ModelAttribute("counter") Counter counter) {  
 counter.increment();  
 return "/index";  
 }  
  
}

Trong đó:

* Annotation @SessionAttributes("counter") được sử dụng để lưu trữ model attribute trong session. Trong đoạn mã trên, model attribute "counter"  sẽ được thêm vào session nếu tên attribute của @ModelAttribute và @SessionAttributes giống nhau.
* Annotation @ModelAttribute liên kết một tham số phương thức hoặc một phương thức trả về giá trị cho một model attribute và sau đó trả nó về một view cụ thể.
* hàm increment() có nhiệm vụ tăng giá trị thuộc tính count của đối tượng counter
* Tạo tầng views: **webapp/WEB-INF/views** => Tạo class: **index.html**

<!DOCTYPE html>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
</head>  
<body>  
<h1 th:text="${'View count: ' + counter.count}"></h1>  
</body>  
</html>

* Add Tomcat chạy chương trình => F5: reload trang liên tục và quan sát kết quả: Có 1 phiên session được lưu như hình ảnh dưới. Khi đóng trình duyệt web thì phiên này sẽ tự động mất count về 0 hoặc ta có thể xóa Session này đi thì count cũng về 0.



### https://github.com/nguyendongminhtri/MD4-Bai10-Session.git

### Hướng dẫn nộp bài:

Up code lên github.

Paste link github vào phần nộp bài.