[Thực hành] Ứng dụng login

**Mục tiêu**

Luyện tập sử dụng Controller.

Luyện tập sử dụng cơ chế Data Binding thông qua @ModelAttribute.

**Điều kiện**

Biết cách sử dụng Controller.

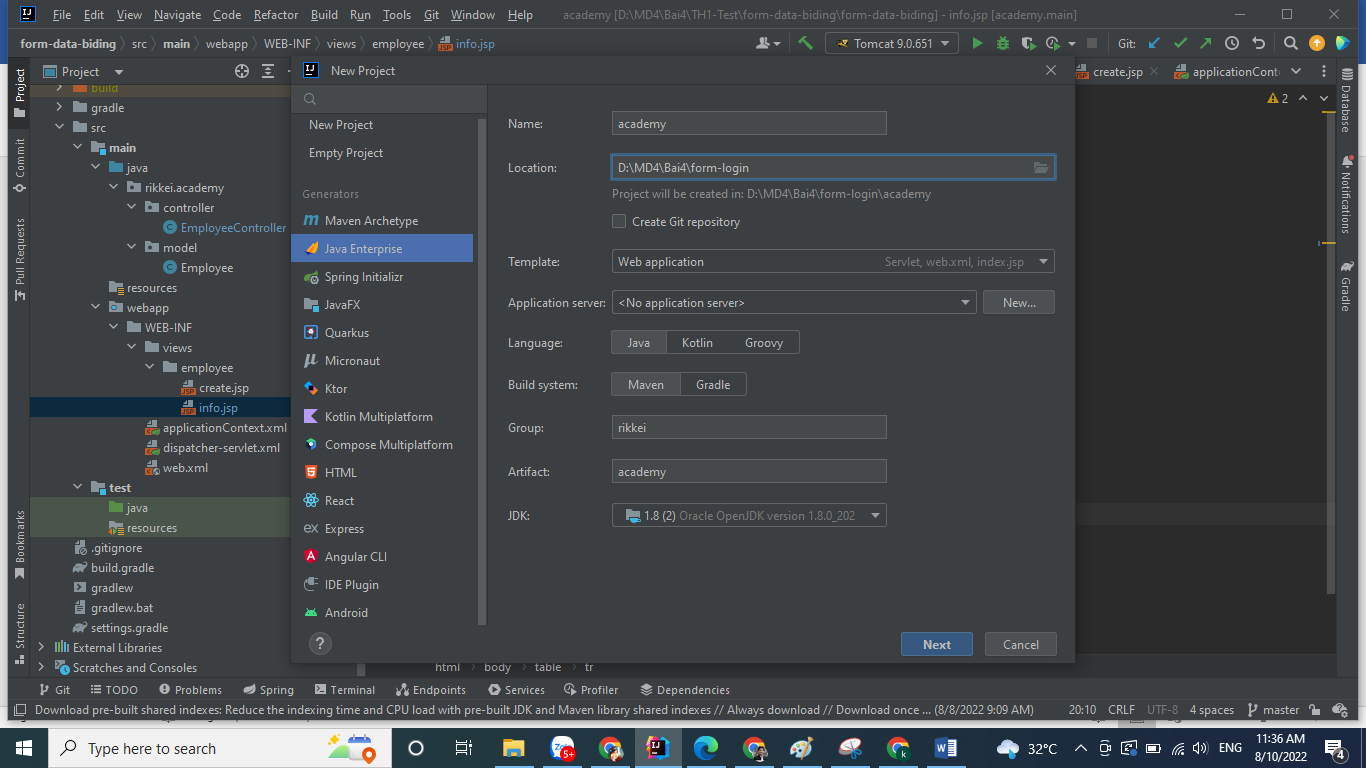
Biết cách sử dụng cơ chế Data Binding thông qua @ModelAttribute.

**Mô tả**

Trong phần này, chúng ta sẽ tạo một ứng dụng Login đơn giản. Ứng dụng cho phép người dùng nhập vào account và password. Nếu thông tin nhập vào là đúng thì sẽ hiển thị các thông số khác của người dùng, chẳng hạn như tên, email...

**Hướng dẫn**

**Bước 1:** Tạo dự án: File => New => Project => Chọn cá thẻ giống như hình dưới:



* Thêm và buil thư viện từ **buil.gradle**

implementation group: 'org.springframework', name: 'spring-webmvc', version: '5.3.2'

* Thêm các file config giống như bài thực hành 1:
* Tạo file: **applicationContext.xml** trong thư mục WEB-INF và thêm code vào config:
* <?xml version="1.0" encoding="UTF-8"?>  
  <beans xmlns="http://www.springframework.org/schema/beans"  
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
   xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">  
    
  </beans>
* Tạo thêm file: **dispatcher-servlet.xml** trong thư mục WEB-INF và thêm code vào như sau:
* <?xml version="1.0" encoding="UTF-8"?>  
  <beans xmlns="http://www.springframework.org/schema/beans"  
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
   xmlns:context="http://www.springframework.org/schema/context"  
   xmlns:mvc="http://www.springframework.org/schema/mvc"  
   xsi:schemaLocation="http://www.springframework.org/schema/beans  
   http://www.springframework.org/schema/beans/spring-beans.xsd  
   http://www.springframework.org/schema/context  
   http://www.springframework.org/schema/context/spring-context.xsd  
   http://www.springframework.org/schema/mvc  
   http://www.springframework.org/schema/mvc/spring-mvc.xsd">  
    
   <context:component-scan base-package="rikkei.academy.controller"/>  
    
   <mvc:annotation-driven/>  
    
   <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">  
   <property name="prefix" value="/WEB-INF/views/"/>  
   <property name="suffix" value=".jsp"/>  
   </bean>  
  </beans>
* Lưu ý: Đọc lại cách hoạt động của ***component-scan, prefix, suffix*** từ bài thực hành trước.
* Cấu hình web.xml
* <?xml version="1.0" encoding="UTF-8"?>  
  <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"  
   version="3.1">  
   <context-param>  
   <param-name>contextConfigLocation</param-name>  
   <param-value>/WEB-INF/applicationContext.xml</param-value>  
   </context-param>  
   <listener>  
   <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  
   </listener>  
   <servlet>  
   <servlet-name>dispatcher</servlet-name>  
   <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
   <load-on-startup>1</load-on-startup>  
   </servlet>  
   <servlet-mapping>  
   <servlet-name>dispatcher</servlet-name>  
   <url-pattern>/</url-pattern>  
   </servlet-mapping>  
  </web-app>
* Tạo pakcage: model => Tạo Class: User => Thêm code vào như sau:

Lớp **User** mô tả các thông tin của người dùng, bao gồm: account, password, name, email, age.

* package rikkei.academy.model;  
    
  public class User {  
   private String account;  
   private String password;  
   private String name;  
   private String email;  
   private int age;  
    
   public User() {  
   }  
    
   public User(String account, String password, String name, String email, int age) {  
   this.account = account;  
   this.password = password;  
   this.name = name;  
   this.email = email;  
   this.age = age;  
   }  
    
   public String getAccount() {  
   return account;  
   }  
    
   public void setAccount(String account) {  
   this.account = account;  
   }  
    
   public String getPassword() {  
   return password;  
   }  
    
   public void setPassword(String password) {  
   this.password = password;  
   }  
    
   public String getName() {  
   return name;  
   }  
    
   public void setName(String name) {  
   this.name = name;  
   }  
    
   public String getEmail() {  
   return email;  
   }  
    
   public void setEmail(String email) {  
   this.email = email;  
   }  
    
   public int getAge() {  
   return age;  
   }  
    
   public void setAge(int age) {  
   this.age = age;  
   }  
    
  }
* Tạo Class: **Login** trong package: model

Lớp **Login** mô tả thông tin của một Login, bao gồm: account và password.

package rikkei.academy.model;  
  
public class Login {  
 private String account;  
 private String password;  
  
 public Login() {  
 }  
  
 public Login(String account, String password) {  
 this.account = account;  
 this.password = password;  
 }  
  
 public String getAccount() {  
 return account;  
 }  
  
 public void setAccount(String account) {  
 this.account = account;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
}

* Tạo package: service => Tạo Class: UserService

Lớp UserService là một lớp mô phỏng nguồn dữ liệu. Lớp UserService chứa một danh sách các user cho trước và phương thức checkLogin() cho phép kiểm tra thông tin login.

* package rikkei.academy.service;  
    
  import rikkei.academy.model.Login;  
  import rikkei.academy.model.User;  
    
  import java.util.ArrayList;  
  import java.util.List;  
    
  public class UserService {  
   private static List<User> *users*;  
    
   static {  
   *users* = new ArrayList<>();  
   User u1 = new User();  
   u1.setAge(10);  
   u1.setName("Chinh");  
   u1.setAccount("chinhcomhut");  
   u1.setEmail("chinhcomhut@rikkei.vn");  
   u1.setPassword("123456");  
   *users*.add(u1);  
    
   User u2 = new User();  
   u2.setAge(15);  
   u2.setName("Dung");  
   u2.setAccount("dung");  
   u2.setEmail("dung@rikkei.academy");  
   u2.setPassword("123456");  
   *users*.add(u2);  
    
   User u3 = new User();  
   u3.setAge(16);  
   u3.setName("Hai");  
   u3.setAccount("hai");  
   u2.setEmail("hai@rikkeisoft.vn");  
   u3.setPassword("123456");  
   *users*.add(u3);  
   }  
   public static User checkLogin(Login login){  
   for (User u: *users*) {  
   if(u.getAccount().equals(login.getAccount())  
   && u.getPassword().equals(login.getPassword())){  
   return u;  
   }  
   }  
   return null;  
   }  
    
  }
* Tạo package: controller => Tạo Class: UserController

Lớp UserController có phương thức home() trả về view **home** kèm theo một người dùng mặc định.

package rikkei.academy.controller;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.servlet.ModelAndView;  
import rikkei.academy.model.Login;  
  
@Controller  
public class UserController {  
 @GetMapping(value = {"/","/home"})  
 public ModelAndView home() {  
 ModelAndView modelAndView = new ModelAndView("home", "login", new Login());  
 return modelAndView;  
 }  
}

* Tạo thư mục views trong WEB-INF => Tạo file: home.jsp

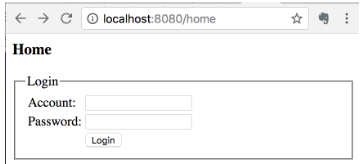
View home.jsp hiển thị form login.

<%@ **taglib** prefix="form" uri="http://www.springframework.org/tags/form" %>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Login</title>  
</head>  
<body>  
<h3>Home</h3>  
<**form:form** action="login" method="post" modelAttribute="login">  
 <fieldset>  
 <legend>Login</legend>  
 <table>  
 <tr>  
 <td><**form:label** path="account">Account:</**form:label**></td>  
 <td><**form:input** path="account" /></td>  
 </tr>  
 <tr>  
 <td><**form:label** path="password">Password:</**form:label**></td>  
 <td><**form:input** path="password"/></td>  
 </tr>  
 <tr>  
 <td></td>  
 <td><**form:button**>Login</**form:button**></td>  
 </tr>  
 </table>  
 </fieldset>  
</**form:form**>  
</body>  
</html>

Thuộc tính modelAttribute của thẻ <form:form> được liên kết tới thuộc tính login của ModelAndView trả về từ phương thức home().

* Xóa file index.jsp, xóa Class: HelloServlet => Add Tomcat => Chạy ứng dụng:

Đi đến đường dẫn http://localhost:8080/home để quan sát form login.



* Bổ sung phương thức login cho UserController

Phương thức login() nhận vào một model Login và thực hiện việc kiểm tra thông qua UserDao.

Nếu đăng nhập thành công thì trả về view **user** kèm theo đối tượng User.

Nếu đăng nhập thất bại thì trả về view **error**.

@PostMapping("/login")  
public ModelAndView login(@ModelAttribute("login") Login login){  
 User user = UserService.*checkLogin*(login);  
 if(user == null){  
 ModelAndView modelAndView = new ModelAndView("error");  
 return modelAndView;  
 } else {  
 ModelAndView modelAndView = new ModelAndView("user");  
 modelAndView.addObject("user", user);  
 return modelAndView;  
 }  
}

* Tạo file: user.jsp trong thư mục views:

View user.jsp hiển thị thông tin của người dùng trong trường hợp đăng nhập thành công.

<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Title</title>  
</head>  
<body>  
<h1>Welcome</h1>  
<h3>Account: ${user.account}</h3>  
<h3>Name: ${user.name}</h3>  
<h3>Email: ${user.email}</h3>  
<h3>Age: ${user.age}</h3>  
</body>  
</html>

* Tạo file: error.jsp trong thư mục views:

View error.jsp hiển thị trang thông báo lỗi trong trường hợp đăng nhập thất bại.

<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Error</title>  
</head>  
<body>  
<h1>Login error</h1>  
</body>  
</html>

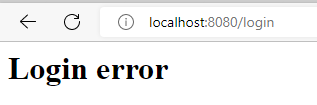
* Chạy ứng dụng

Đi đến đường dẫn http://localhost:8080/home để quan sát form login.

Nhập vào thông tin đúng account và password, quan sát kết quả.



Nhập và thông tin sai account và password, quan sát kết quả.



Mã nguồn tham khảo:

https://github.com/rikkei-academy/MD4-Bai4-TH2-Form-Login.git

## ****Hướng dẫn nộp bài:****

Up code lên github.

Paste link github vào phần nộp bài.