# 登录功能

## 目录

## 任务目标

- 完成基本的登录功能
- 经典图形验证码

## 知识点

1、用户登录

登录页面前端部分，关注的重点还是用户账号和密码在js代码里要客户端验证，这是验证的第一步，保证用户输入格式的正确性同时也从一方面减少用户向后台发送没必要的错误请求。

前端向后端请求的方式使用POST

在后台接收前端传送的信息同样要经过服务端验证， 因为有可能一些人并不是通过合法的方式进行访问， 可以减少没必要的数据库查询。

2、用户密码安全性

密码要如何保存， 明文存入数据库？当然非常非常非常不推荐， 由于大多数客户的习惯都是使用相同的密码， 如果明文在发生信息泄露的情况下容易发生撞库的事情。所以在密码的保存上最好在后台使用MD5的两次加盐加密的方式存储到数据库中。

3、登录状态的保存

由于http协议是无状态的， 所以要记录用户的登录状态就要靠后台相应数据的维护来记录， 我们通常都是登录成功后在seesion中保存登录用户。

4、防止恶意攻击

防止大量重复请求、机器人暴力访问等情况的，最有效的手段就是加入验证码， 同时记录某个用户在或

ip在某个时段内如果尝试的失败登录次数超过一定阈值就限制其在几分钟内不能继续登录。

## 实现思路

1. 完成基本的登录功能：

- Login 页面采集 username 和 Password
- 根据 useraname 判断 User 使用存在
- 如果存在，使用 MD5 的两次加盐加密比对数据库中密码是否行一致
- 将 User 保存到 Session 中

URL 请求分析：

1. get：/login

逻辑：跳转到注册页面

参数：username，password

返回值：ModelAndView

2. post：/login

逻辑：保存提交的用户名和密码

参数：

| 参数名 | 类型 | 是否必填 | 说明 |
|---|---|---|---|
| username | String | N |  |
| password | String | N |  |

返回值：ModelAndView

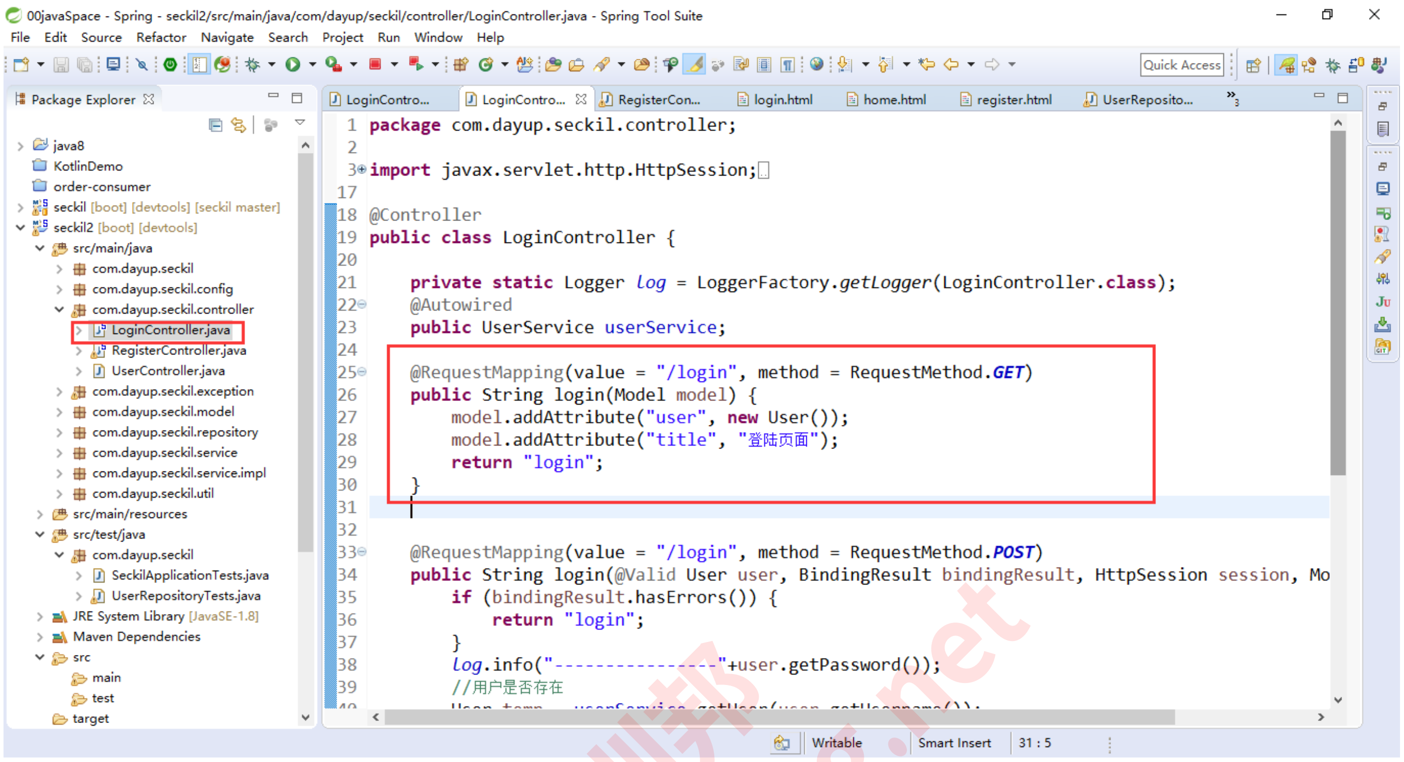
2. 单元测试之军工六性：

稳定性、适应性、安全性、保障性、维修性、测试性

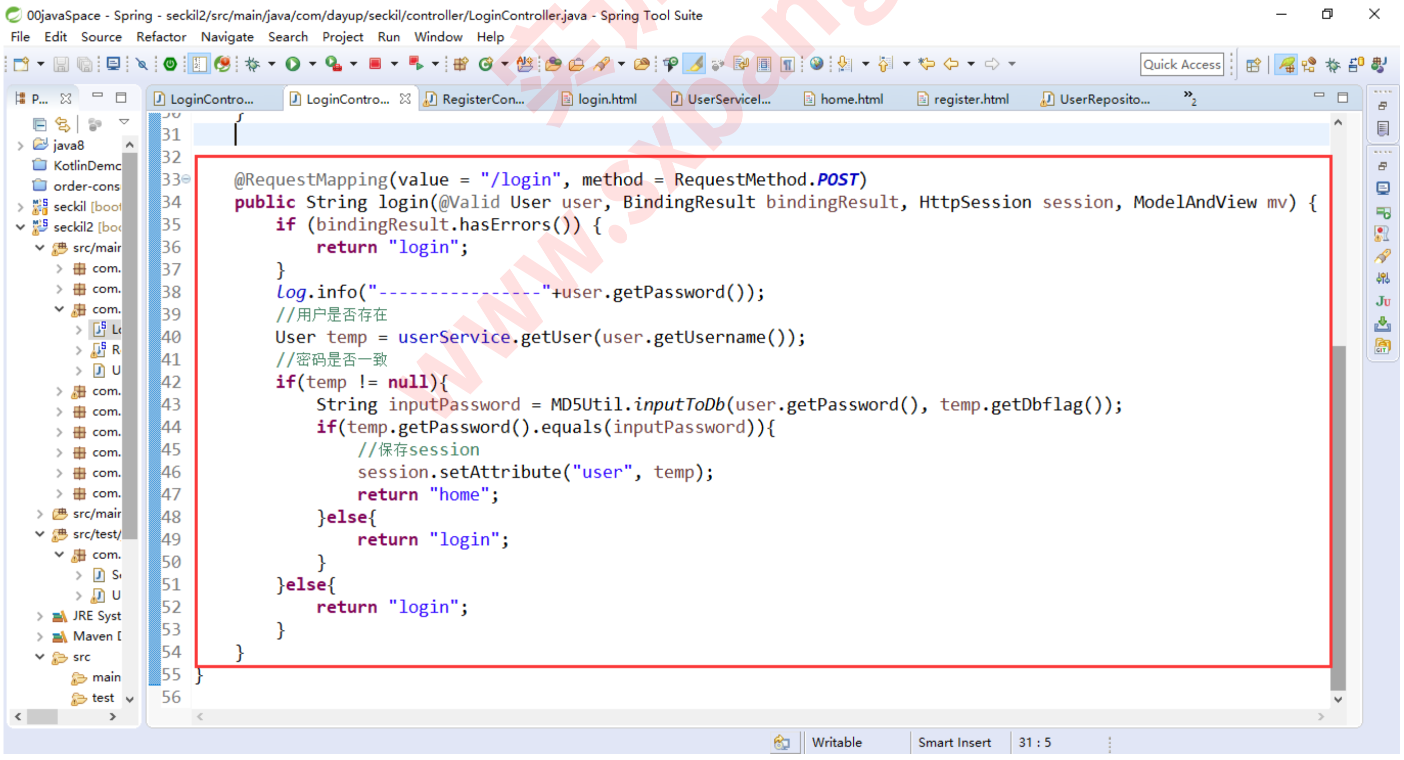
## 编码步骤

1. 完成基本的登录功能

Step1.LoginController 的编写

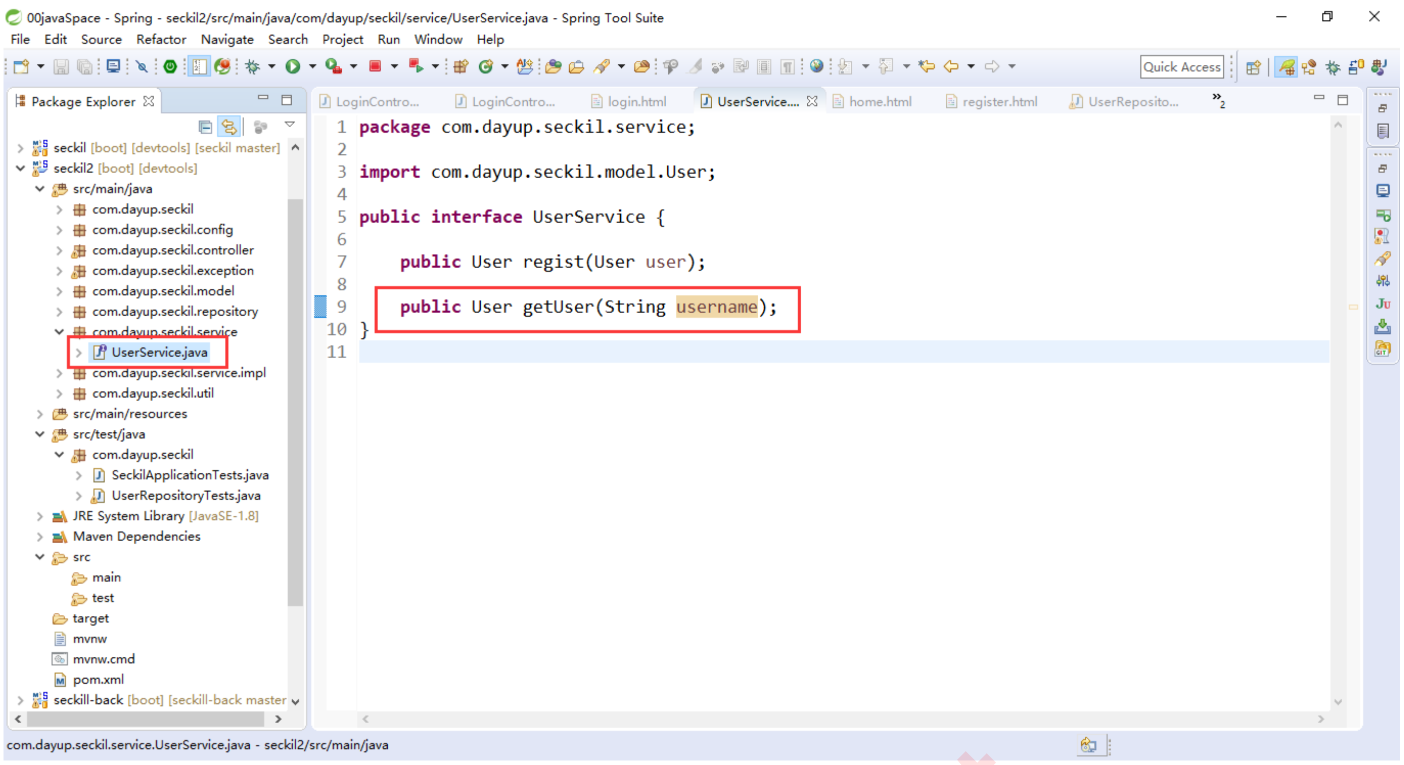


```java
package com.dayup.seckil.controller;

import javax.servlet.http.HttpSession;

@Controller
public class LoginController {

    private static Logger log = LoggerFactory.getLogger(LoginController.class);
    @Autowired
    public UserService userService;

    @RequestMapping(value = "/login", method = RequestMethod.GET)
    public String login(Model model) {
        model.addAttribute("user", new User());
        model.addAttribute("title", "登陆页面");
        return "login";
    }


    @RequestMapping(value = "/login", method = RequestMethod.POST)
    public String login(@Valid User user, BindingResult bindingResult, HttpSession session, Mo
        if (bindingResult.hasErrors()) {
            return "login";
        }
        log.info("---------------"+user.getPassword());
        //用户是否存在
```



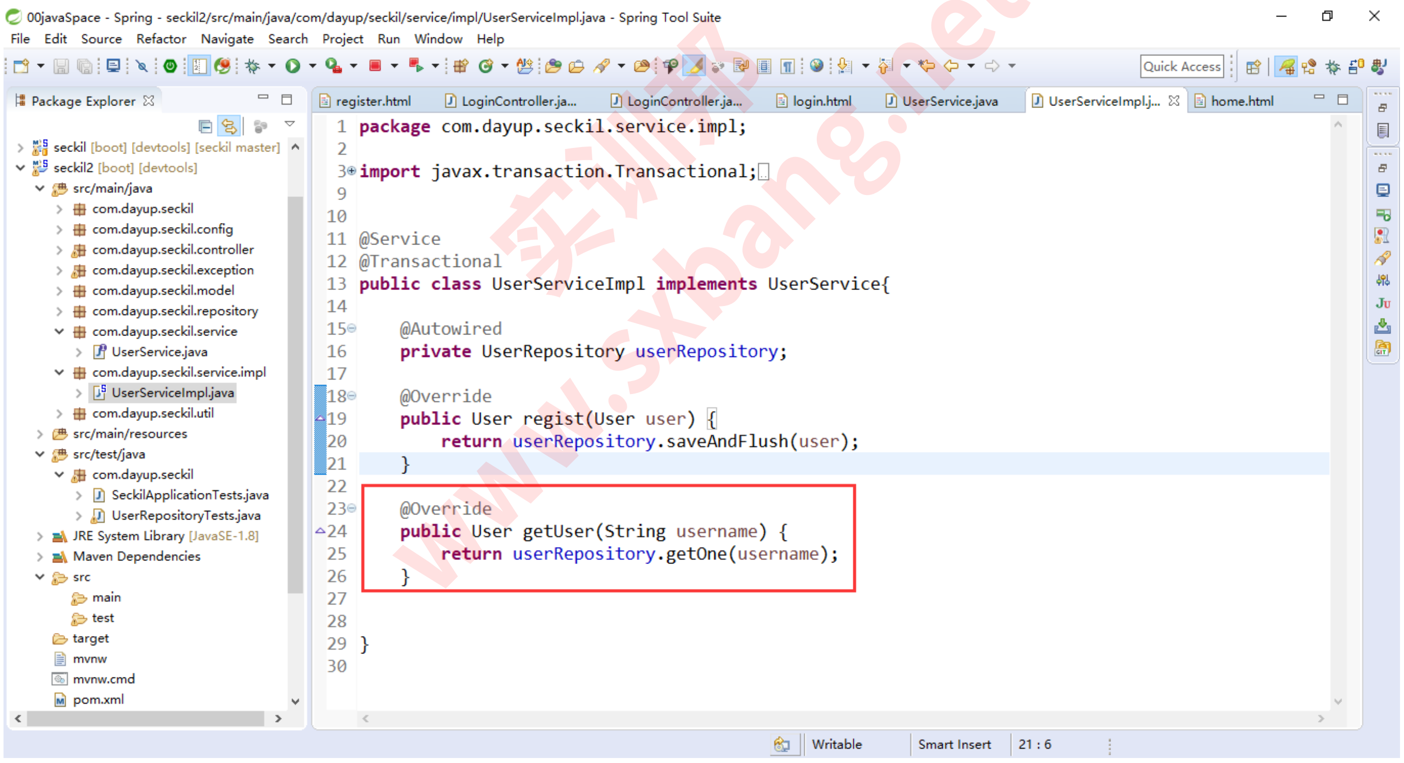
```java
    @RequestMapping(value = "/login", method = RequestMethod.POST)
    public String login(@Valid User user, BindingResult bindingResult, HttpSession session, ModelAndView mv) {
        if (bindingResult.hasErrors()) {
            return "login";
        }
        log.info("---------------"+user.getPassword());
        //用户是否存在
        User temp = userService.getUser(user.getUsername());
        //密码是否一致
        if(temp != null){
            String inputPassword = MD5Util.inputToDb(user.getPassword(), temp.getDbflag());
            if(temp.getPassword().equals(inputPassword)){
                //保存session
                session.setAttribute("user", temp);
                return "home";
            }else{
                return "login";
            }
        }else{
            return "login";
        }
    }
}
```

Step2.修改 UserService



00javaSpace - Spring - seckil2/src/main/java/com/dayup/seckil/service/UserService.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
seckil [boot] [devtools] [seckil master]
seckil2 [boot] [devtools]
src/main/java
com.dayup.seckil
com.dayup.seckil.config
com.dayup.seckil.controller
com.dayup.seckil.exception
com.dayup.seckil.model
com.dayup.seckil.repository
com.dayup.seckil.service
UserService.java
com.dayup.seckil.service.impl
com.dayup.seckil.util
src/main/resources
src/test/java
com.dayup.seckil
SeckilApplicationTests.java
UserRepositoryTests.java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
main
test
target
mvnw
mvnw.cmd
pom.xml
seckill-back [boot] [seckill-back master]
com.dayup.seckil.service.UserService.java - seckil2/src/main/java

```java
package com.dayup.seckil.service;

import com.dayup.seckil.model.User;

public interface UserService {

    public User regist(User user);

    public User getUser(String username);
}
```



00javaSpace - Spring - seckil2/src/main/java/com/dayup/seckil/service/impl/UserServiceImpl.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
UserServiceImpl.java
Writable
Smart Insert
21 : 6

```java
package com.dayup.seckil.service.impl;

import javax.transaction.Transactional;


@Service
@Transactional
public class UserServiceImpl implements UserService{

    @Autowired
    private UserRepository userRepository;

    @Override
    public User regist(User user) {
        return userRepository.saveAndFlush(user);
    }

    @Override
    public User getUser(String username) {
        return userRepository.getOne(username);
    }


}
```

## Step3.编写 login.html 以及 home.html

### login.html:

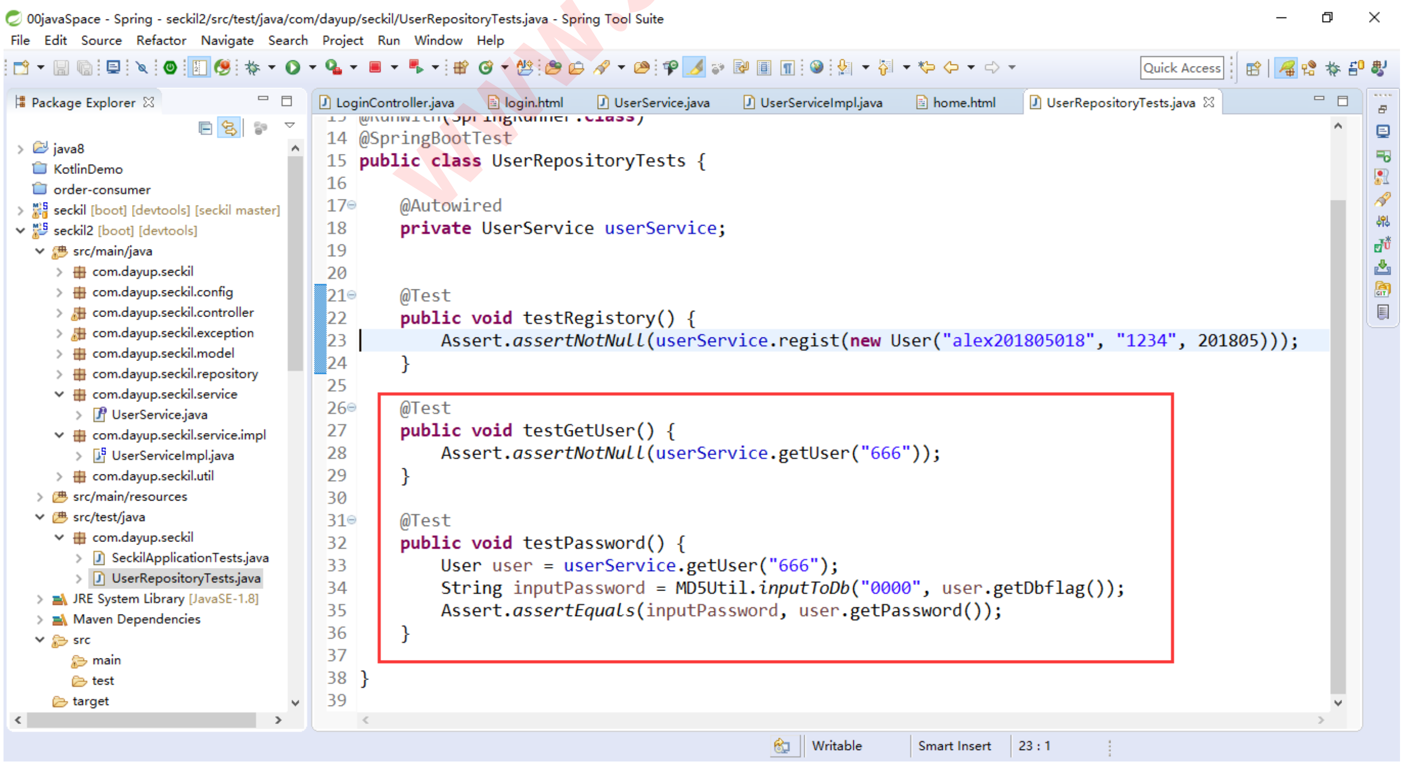
```html
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
        xmlns:th="http://www.thymeleaf.org"
        xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3"
        xmlns:layout="http://www.ultrag.net.nz/thymeleaf/layout">
```

```
<head>
<meta charset="UTF-8">
<title>注册</title>
<link th:href="@{/css/bootstrap.min.css}" rel="stylesheet"/>
</head>
<body>
        <div class="container" style="text-align:center;margin-top:50px;">
            <div class="row col-md-6 col-md-offset-3">
                 <div class="panel panel-default">
                           <div class="panel-heading" th:text="${title}">title</div>
                           <div class="panel-body">
                             <form id="registerForm"   th:action="@{/login}" th:object="${user}" method="post">
                                    <div class="input-group">
                                            <span class="input-group-addon">username</span>
                                            <input id="username" type="text" th:field="*{username}" class="form-control" placeholder="用户名"
required>
                                       </div>
                                       <div><span th:if="${#fields.hasErrors('username')}" th:errors="*{username}"
style="color:red;"></span></div>
                                       <br>
                                       <div class="input-group">
                                            <span class="input-group-addon">password</span>
                                            <input id="password" type="password" name="password" class="form-control" placeholder="密码"
required>
                                       </div>
                                       <div><span th:if="${#fields.hasErrors('password')}" th:errors="*{password}"
style="color:red;"></span></div>
                                       <br>
                                           <button type="submit" class="btn btn-primary register-btn">登陆</button>
                                           <button type="reset" class="btn btn-warning">重置</button>
                                    </form>
                           </div>
                      </div>
            </div>
        </div>
</body>
<script th:src="@{/js/jquery-3.3.1.js}"></script>
<script th:src="@{/js/jquery.validate.min.js}"></script>
<script th:src="@{/js/additional-methods.min.js}"></script>
<script th:src="@{/js/messages_zh.min.js}"></script>
<script th:src="@{/js/jquery.md5.js}"></script>
<script>

        $("#registerForm").validate({
           rules: {
                        password: {
                         required: true,
                }
            },
        errorPlacement: function(error, element) {
        error.appendTo(element.parent().next());
        }
    });

</script>
    </html>
```

home.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org"
      xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3"
      xmlns:layout="http://www.ultrag.net.nz/thymeleaf/layout">
<head>
<meta charset="UTF-8">
<title>Home</title>
<link th:href="@{/css/bootstrap.min.css}" rel="stylesheet"/>
</head>
<body>
    <div th:text="${session.user.username}"></div>
</body>
</html>
```

## Step4.Junit 测试


00javaSpace - Spring - seckil2/src/test/java/com/dayup/seckil/UserRepositoryTests.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
LoginController.java
login.html
UserService.java
UserServiceImpl.java
home.html
UserRepositoryTests.java
Writable
Smart Insert
23 : 1

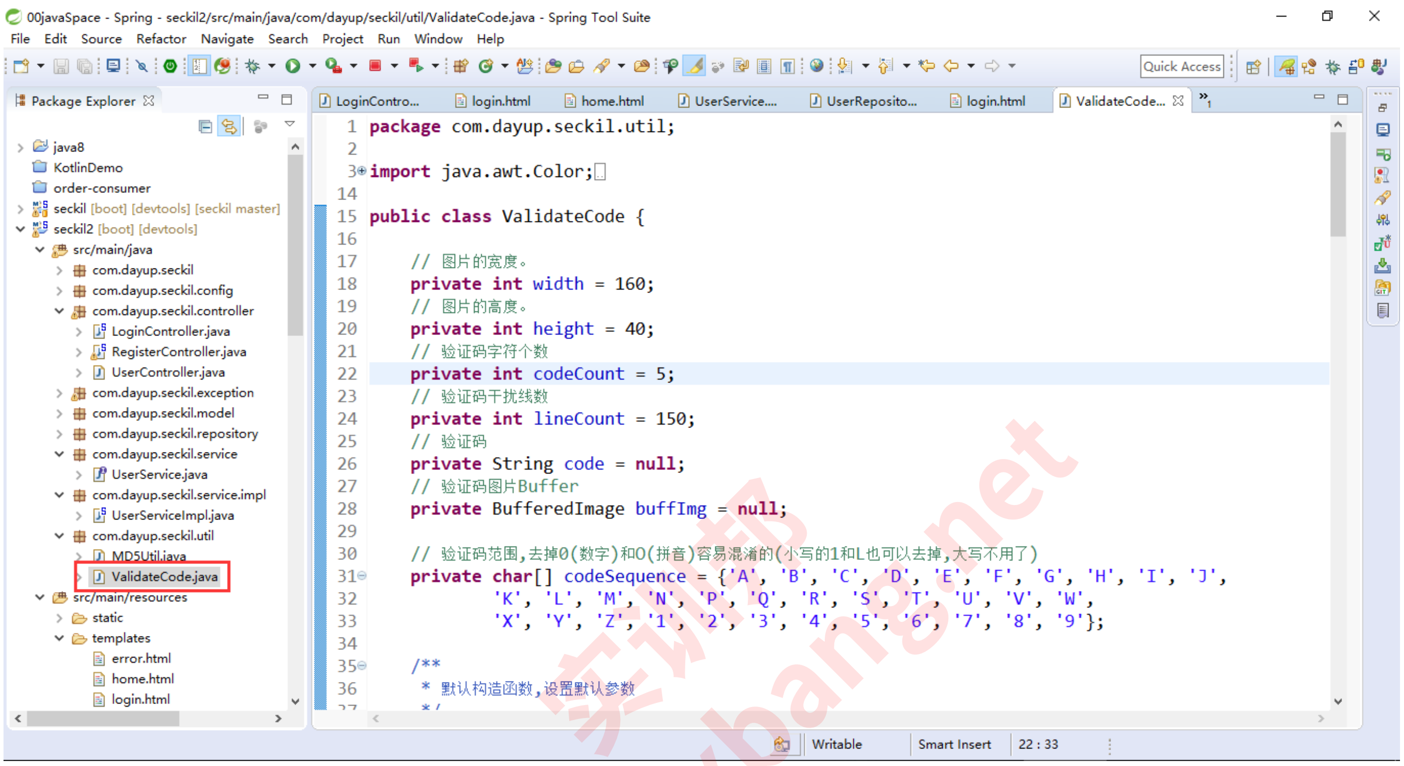
```
java8
KotlinDemo
order-consumer
seckil [boot] [devtools] [seckil master]
seckil2 [boot] [devtools]
  src/main/java
    com.dayup.seckil
    com.dayup.seckil.config
    com.dayup.seckil.controller
    com.dayup.seckil.exception
    com.dayup.seckil.model
    com.dayup.seckil.repository
    com.dayup.seckil.service
      UserService.java
    com.dayup.seckil.service.impl
      UserServiceImpl.java
    com.dayup.seckil.util
  src/main/resources
  src/test/java
    com.dayup.seckil
      SeckilApplicationTests.java
      UserRepositoryTests.java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
    main
    test
  target
```

```java
@RunWith(SpringRunner.class)
@SpringBootTest
public class UserRepositoryTests {

    @Autowired
    private UserService userService;


    @Test
    public void testRegistory() {
        Assert.assertNotNull(userService.regist(new User("alex201805018", "1234", 201805)));
    }

    @Test
    public void testGetUser() {
        Assert.assertNotNull(userService.getUser("666"));
    }

    @Test
    public void testPassword() {
        User user = userService.getUser("666");
        String inputPassword = MD5Util.inputToDb("0000", user.getDbflag());
        Assert.assertEquals(inputPassword, user.getPassword());
    }
}
```

2. 经典图形验证码

参考地址：https://blog.csdn.net/cjm1103/article/details/71171842

Step1.新增图形验证码生成工具


00javaSpace - Spring - seckil2/src/main/java/com/dayup/seckil/util/ValidateCode.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
java8
KotlinDemo
order-consumer
seckil [boot] [devtools] [seckil master]
seckil2 [boot] [devtools]
src/main/java
com.dayup.seckil
com.dayup.seckil.config
com.dayup.seckil.controller
LoginController.java
RegisterController.java
UserController.java
com.dayup.seckil.exception
com.dayup.seckil.model
com.dayup.seckil.repository
com.dayup.seckil.service
UserService.java
com.dayup.seckil.service.impl
UserServiceImpl.java
com.dayup.seckil.util
MD5Util.java
ValidateCode.java
src/main/resources
static
templates
error.html
home.html
login.html
LoginContro... login.html home.html UserService.... UserReposito... login.html ValidateCode...
Writable
Smart Insert
22 : 33

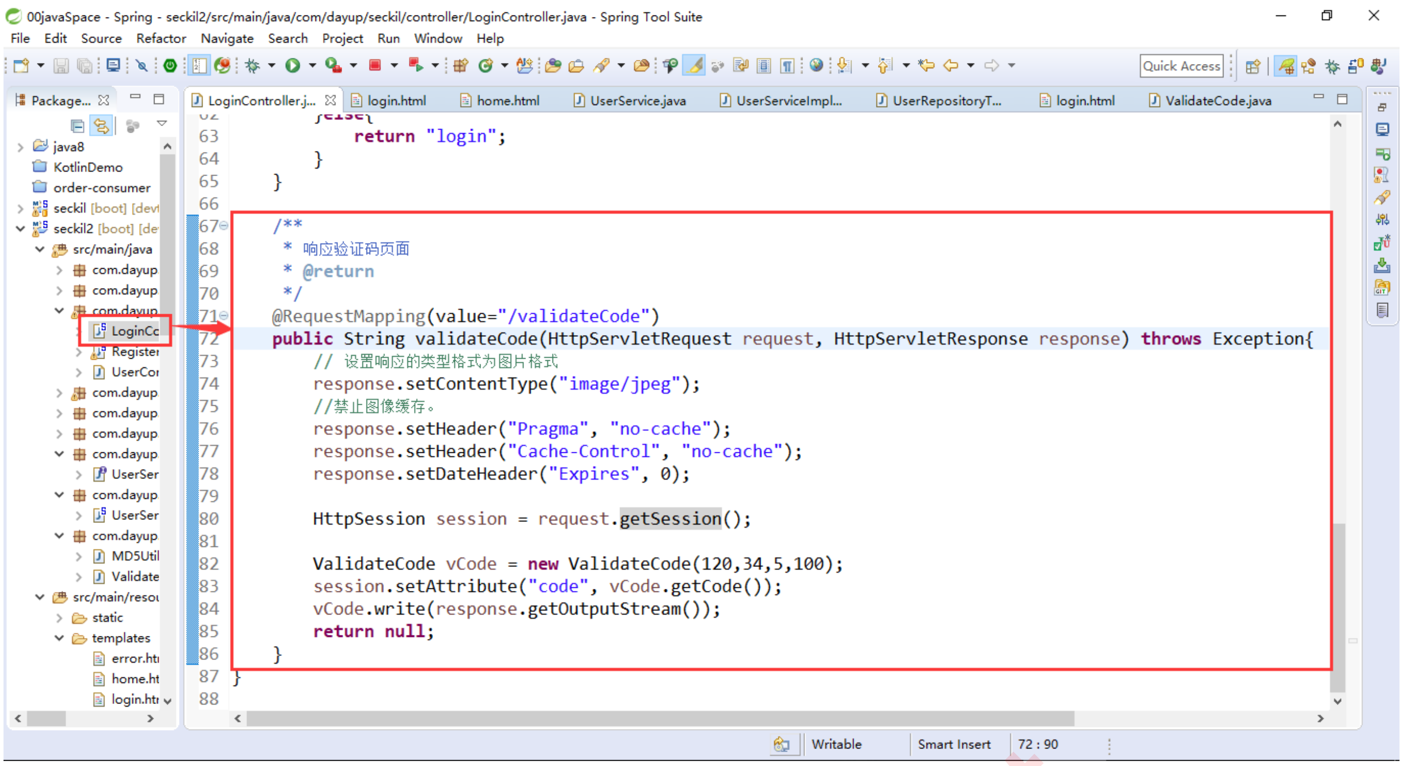
```java
package com.dayup.seckil.util;

import java.awt.Color;

public class ValidateCode {

    // 图片的宽度。
    private int width = 160;
    // 图片的高度。
    private int height = 40;
    // 验证码字符个数
    private int codeCount = 5;
    // 验证码干扰线数
    private int lineCount = 150;
    // 验证码
    private String code = null;
    // 验证码图片Buffer
    private BufferedImage buffImg = null;

    // 验证码范围,去掉0(数字)和O(拼音)容易混淆的(小写的1和L也可以去掉,大写不用了)
    private char[] codeSequence = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
            'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W',
            'X', 'Y', 'Z', '1', '2', '3', '4', '5', '6', '7', '8', '9'};

    /**
     * 默认构造函数,设置默认参数
```

```java
package com.dayup.seckil.util;


import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics2D;
import java.awt.image.BufferedImage;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.util.Date;
import java.util.Random;


import javax.imageio.ImageIO;


public class ValidateCode {


     // 图片的宽度。
    private int width = 160;
    // 图片的高度。
    private int height = 40;
```

```java
// 验证码字符个数
private int codeCount = 5;
// 验证码干扰线数
private int lineCount = 150;
// 验证码
private String code = null;
// 验证码图片Buffer
private BufferedImage buffImg = null;

// 验证码范围,去掉0(数字)和O(拼音)容易混淆的(小写的1和L也可以去掉,大写不用了)
private char[] codeSequence = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
        'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W',
        'X', 'Y', 'Z', '1', '2', '3', '4', '5', '6', '7', '8', '9'};

/**
 * 默认构造函数,设置默认参数
 */
public ValidateCode() {
    this.createCode();
}

/**
 * @param width  图片宽
 * @param height 图片高
 */
public ValidateCode(int width, int height) {
    this.width = width;
    this.height = height;
    this.createCode();
}

/**
 * @param width     图片宽
 * @param height    图片高
 * @param codeCount 字符个数
 * @param lineCount 干扰线条数
 */
public ValidateCode(int width, int height, int codeCount, int lineCount) {
    this.width = width;
    this.height = height;
    this.codeCount = codeCount;
    this.lineCount = lineCount;
    this.createCode();
}

public void createCode() {
    int x = 0, fontHeight = 0, codeY = 0;
```

```
        int red = 0, green = 0, blue = 0;

        x = width / (codeCount + 2);//每个字符的宽度(左右各空出一个字符)
        fontHeight = height - 2;//字体的高度
        codeY = height - 4;

        // 图像buffer
        buffImg = new BufferedImage(width, height, BufferedImage.TYPE_INT_RGB);
        Graphics2D g = buffImg.createGraphics();
        // 生成随机数
        Random random = new Random();
        // 将图像填充为白色
        g.setColor(Color.WHITE);
        g.fillRect(0, 0, width, height);
        // 创建字体,可以修改为其它的
        Font font = new Font("Fixedsys", Font.PLAIN, fontHeight);
//        Font font = new Font("Times New Roman", Font.ROMAN_BASELINE, fontHeight);
        g.setFont(font);

        for (int i = 0; i < lineCount; i++) {
            // 设置随机开始和结束坐标
            int xs = random.nextInt(width);//x坐标开始
            int ys = random.nextInt(height);//y坐标开始
            int xe = xs + random.nextInt(width / 8);//x坐标结束
            int ye = ys + random.nextInt(height / 8);//y坐标结束

            // 产生随机的颜色值，让输出的每个干扰线的颜色值都将不同。
            red = random.nextInt(255);
            green = random.nextInt(255);
            blue = random.nextInt(255);
            g.setColor(new Color(red, green, blue));
            g.drawLine(xs, ys, xe, ye);
        }

        // randomCode记录随机产生的验证码
        StringBuffer randomCode = new StringBuffer();
        // 随机产生codeCount个字符的验证码。
        for (int i = 0; i < codeCount; i++) {
            String strRand = String.valueOf(codeSequence[random.nextInt(codeSequence.length)]);
            // 产生随机的颜色值，让输出的每个字符的颜色值都将不同。
            red = random.nextInt(255);
            green = random.nextInt(255);
            blue = random.nextInt(255);
            g.setColor(new Color(red, green, blue));
            g.drawString(strRand, (i + 1) * x, codeY);
            // 将产生的四个随机数组合在一起。
            randomCode.append(strRand);
        }
        // 将四位数字的验证码保存到Session中。
        code = randomCode.toString();
    }

    public void write(String path) throws IOException {
        OutputStream sos = new FileOutputStream(path);
        this.write(sos);
    }

    public void write(OutputStream sos) throws IOException {
        ImageIO.write(buffImg, "png", sos);
        sos.close();
    }

    public BufferedImage getBuffImg() {
        return buffImg;
    }

    public String getCode() {
        return code;
    }

    /**
     * 测试函数,默认生成到d盘
     * @param args
     */
    public static void main(String[] args) {
        ValidateCode vCode = new ValidateCode(160,40,5,150);
        try {
            String path="E:/"+new Date().getTime()+".png";
            System.out.println(vCode.getCode()+" >"+path);
            vCode.write(path);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

## Step2.输出生成好的图片流


00javaSpace - Spring - seckil2/src/main/java/com/dayup/seckil/controller/LoginController.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
LoginController.j... login.html home.html UserService.java UserServiceImpl... UserRepositoryT... login.html ValidateCode.java
Writable Smart Insert 72 : 90

```
            return "login";
        }
    }

    /**
     * 响应验证码页面
     * @return
     */
    @RequestMapping(value="/validateCode")
    public String validateCode(HttpServletRequest request, HttpServletResponse response) throws Exception{
        // 设置响应的类型格式为图片格式
        response.setContentType("image/jpeg");
        //禁止图像缓存。
        response.setHeader("Pragma", "no-cache");
        response.setHeader("Cache-Control", "no-cache");
        response.setDateHeader("Expires", 0);

        HttpSession session = request.getSession();

        ValidateCode vCode = new ValidateCode(120,34,5,100);
        session.setAttribute("code", vCode.getCode());
        vCode.write(response.getOutputStream());
        return null;
    }
}
```

## Step3.页面调用

### Html 编写：

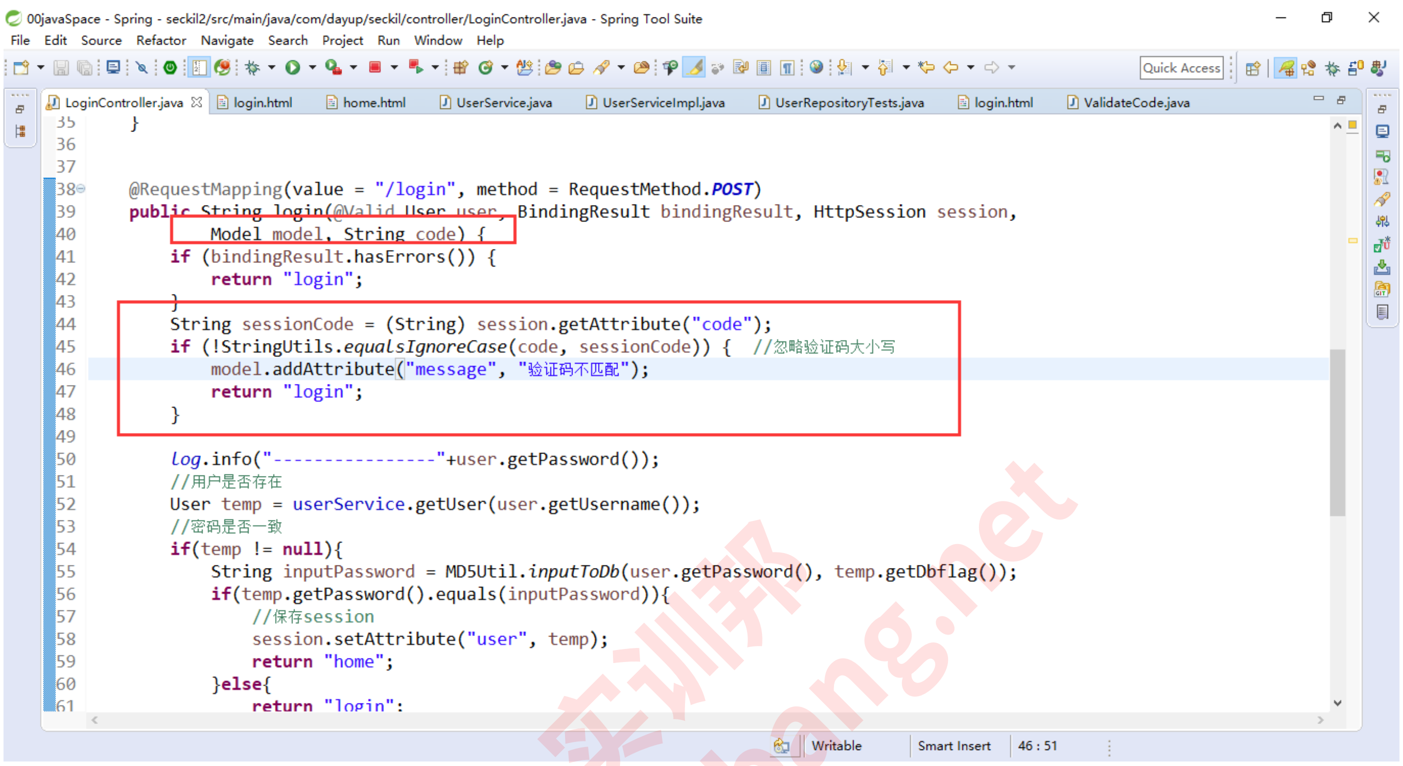
```
<div class="input-group">
  <span class="input-group-addon">验证码</span>
  <input type="text" id="code" name="code" class="form-control" style="width:250px;"/>
  <img id="imgObj" alt="验证码" src="/validateCode" class="img-responsive" onclick="changeImg()"/>
</div>
```

### 脚本编写：

```
// 刷新图片
 function changeImg() {
     var imgSrc = $("#imgObj");
     var src = imgSrc.attr("src");
     imgSrc.attr("src", changeUrl(src));
 }
 //为了使每次生成图片不一致，即不让浏览器读缓存，所以需要加上时间戳
 function changeUrl(url) {
     var timestamp = (new Date()).valueOf();
     var index = url.indexOf("?",url);
     if (index > 0) {
         url = url.substring(index, url.indexOf(url, "?"));
     }
     if ((url.indexOf("&") >= 0)) {
         url = url + "xtamp=" + timestamp;
     } else {
         url = url + "?timestamp=" + timestamp;
```

```
        }
        return url;
    }
```

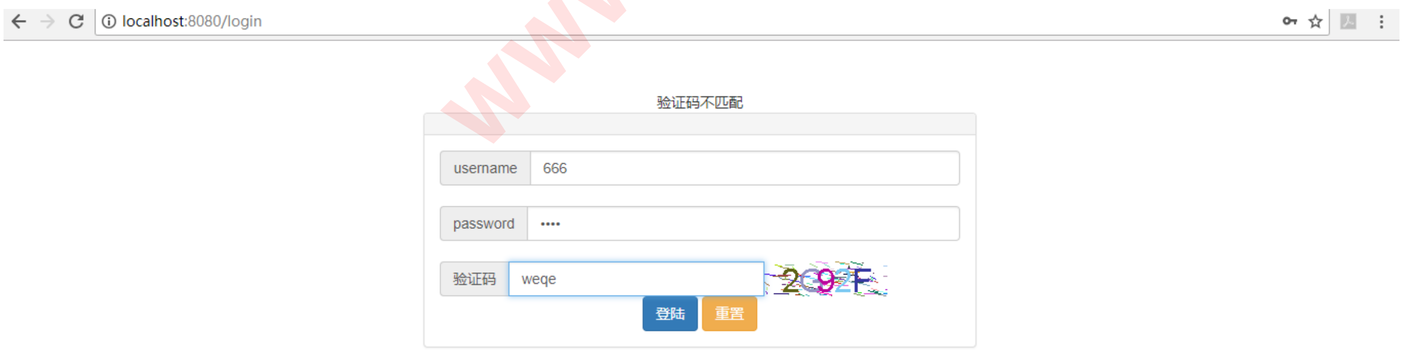
## Step4. 在登陆时检查验证码是否正确：



00javaSpace - Spring - seckil2/src/main/java/com/dayup/seckil/controller/LoginController.java - Spring Tool Suite

```
    }


    @RequestMapping(value = "/login", method = RequestMethod.POST)
    public String login(@Valid User user, BindingResult bindingResult, HttpSession session,
            Model model, String code) {
        if (bindingResult.hasErrors()) {
            return "login";
        }
        String sessionCode = (String) session.getAttribute("code");
        if (!StringUtils.equalsIgnoreCase(code, sessionCode)) {  //忽略验证码大小写
            model.addAttribute("message", "验证码不匹配");
            return "login";
        }

        log.info("----------------"+user.getPassword());
        //用户是否存在
        User temp = userService.getUser(user.getUsername());
        //密码是否一致
        if(temp != null){
            String inputPassword = MD5Util.inputToDb(user.getPassword(), temp.getDbflag());
            if(temp.getPassword().equals(inputPassword)){
                //保存session
                session.setAttribute("user", temp);
                return "home";
            }else{
                return "login";
```

## Step5.运行验证



localhost:8080/login

验证码不匹配

|  |  |
|---|---|
| username | 666 |
| password | •••• |
| 验证码 | weqe |

## 源码参考

请从我们提供的码云上获取。

## 回马枪总结

NA