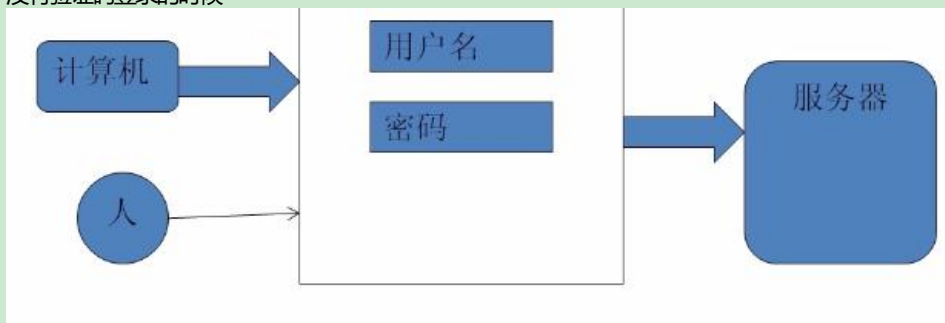


验证码

没有验证码登录的时候



没有验证码带来的问题

- 1 对特定用户不断登录破解密码
- 2 对某个网站创建用户
- 3 对某个网站提交垃圾
- 4 对某个网站刷票

有验证码登录的时候

我们需要通过验证码，由用户肉眼识别其中的验证码信息，从而区分是人还是计算机

验证码(CAPTCHA):是一种区分用户是计算机还是人的公共全自动程序。

作用：可以防止恶意破解密码，刷票，论坛灌水，有效防止了某个黑客对某一个特定注册用户特定程序暴力破解方式进行不断的登录方式。实际上验证码是现在很多网站通行的方式（比如招商银行的网上个人银行，百度社区），我们利用比较简易的方式实现了这个功能。

制作网页

验证码包含两个部分

- 1 验证码
- 2 显示验证码的图片

网页显示验证码：

- 网页显示验证码：
- ```
<input type="text" id="verifyCode" name="verifyCode" size="6" />
```
- ```

```

生成图片用到的类

- 1 bufferedImage用到的图像缓冲区
- 2 Graphics绘制图片
- 3 Color获取颜色
- 4 Random生成随机数
- 5 ImageIO输出图片

• ImageServlet类

- ① 定义BufferedImage对象
- ② 获得Graphics对象
- ③ 通过Random产生随记验证码信息
- ④ 使用Graphics绘制图片
- ⑤ 记录验证码信息到session中
- ⑥ 使用ImageIO输出图片

校验验证码是否正确

LoginServlet类

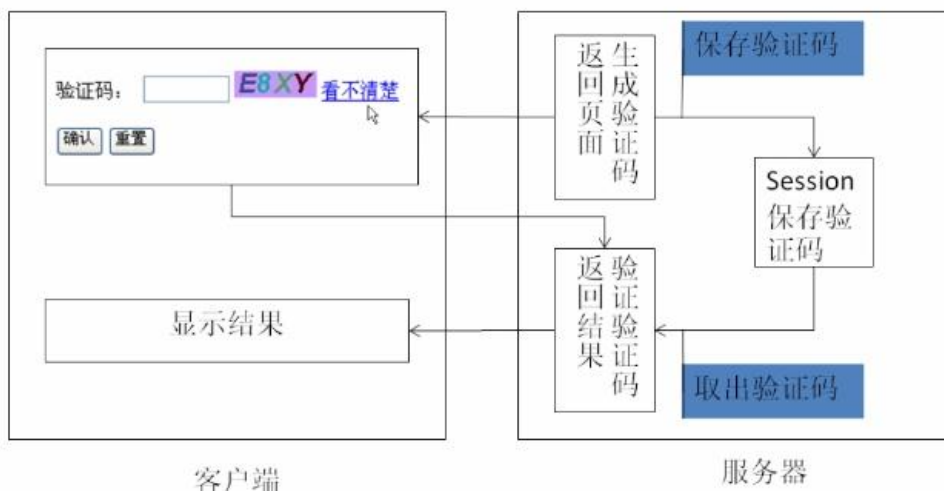
1 获取页面验证码

2 获取session保存的验证码

3 比较验证码

4 返回校验结果

使用servlet实现验证码流程



开源组件实现验证码

Jacptcha :一个用来生成图形验证码的java开源组件,使用起来也是非常的简单,与spring结合,可产生多种形式的验证码

Kaptcha:一个非常实用的验证码工具,有了它,你开源生成各种样式的验证码,因为他是可配置的

kaptcha实现组件实现验证码

下载kaptcha-2.3.jar

将jar包导入项目

编写页面

配置web.xml

启动项目

kaptcha组件扩展

1 获取随机数值 结果相加

2 将计算公式写到验证码图片中

3 将相加结果放到session中

以此为目标我们需要重写KaptchaServlet类

- Kaptcha图片边框颜色：

```
<init-param>  
  <description>  
    边框颜色，合法值：white,black,blue.等  
  </description>  
  <param-name>kaptcha.border.color</param-name>  
  <param-value>black</param-value>  
</init-param>
```

- Kaptcha图片实现类：

```
<init-param>  
  <description>图片实现类</description>  
  <param-name>kaptcha.producer.impl</param-name>  
  <param-value>  
    com.google.code.kaptcha.impl.DefaultKaptcha  
  </param-value>  
</init-param>
```

- Kaptcha文本实现类：

```
<init-param>  
  <description>文本实现类</description>  
  <param-name>kaptcha.textproducer.impl</param-name>  
  <param-value>  
    com.google.code.kaptcha.text.impl.DefaultTextCreator  
  </param-value>  
</init-param>
```

- Kaptcha验证码值配置：

```
<init-param>  
  <description>文本集合，验证码值从此集合中获</description>  
  <param-name>  
    kaptcha.textproducer.char.string  
  </param-name>  
  <param-value>abcde2345678gfynmnpwx</param-value>  
</init-param>
```

- Kaptcha验证码字体配置：

```
<init-param>  
  <description>字体 Arial, Courier</description>  
  <param-name>  
    kaptcha.textproducer.font.names  
  </param-name>  
  <param-value>Arial, Courier</param-value>  
</init-param>
```

- Kaptcha验证码字体大小配置：

```
<init-param>  
  <description>字体大小 40px.</description>  
  <param-name>  
    kaptcha.textproducer.font.size  
  </param-name>  
  <param-value>40</param-value>  
</init-param>
```

- Kaptcha验证码干扰实现类配置：

```
<init-param>  
  <description>干扰实现类</description>  
  <param-name>kaptcha.noise.impl</param-name>  
  <param-value>  
    com.google.code.kaptcha.impl.DefaultNoise  
  </param-value>  
</init-param>
```

- Kaptcha验证码在session中存放验证码的key键配置：

```
<init-param>  
  <description>session中存放验证码的key键  
  </description>  
  <param-name>kaptcha.session.key</param-name>  
  <param-value>KAPTCHA_SESSION_KEY  
  </param-value>  
</init-param>
```

... 很多

Kaptcha组件配置效果图



水波纹样式



鱼眼纹样式



阴影样式

算术验证码的实现

```
String capText = this.kaptchaProducer.createText();//获得输入值
```

```
String s1 = capText.substring(0, 1);//分割为第一个
```

```
String s2 = capText.substring(1, 2);//第二个
```

```
int r = Integer.valueOf(s1).intValue() + Integer.valueOf(s2).intValue();//将相加的值保存到r中
```

```
req.getSession().setAttribute(this.sessionKeyValue, String.valueOf(r));//保存到session 这个this.sessionKeyValue是这个kaptcha本来就有的一个session属性
```

```
BufferedImage bi = this.kaptchaProducer.createImage(s1+"+"+s2+"=?");//设置图片格式
```