**Patchca简单而功能强大**

**Jar包下载：http://patchca.googlecode.com/files/patchca-0.5.0.zip**

**源代码：http://patchca.googlecode.com/svn/trunk/**

**一：定义一个servlet生成图片（先引入jar包）**

1. **：配置xml**

<servlet>

<servlet-name>patchca</servlet-name>

<servlet-class>com.ssh.servlet.CaptchaServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>patchca</servlet-name>

<url-pattern>/patchca.png</url-pattern>

</servlet-mapping>

1. **：servlet（自定义**CaptchaService**）**

**public** **class** CaptchaServlet **extends** HttpServlet {

**private** **static** **final** **long** *serialVersionUID* = 4968328161261528097L;

**private** **static** MyCaptchaService *cs* = **null**;

@Override

**public** **void** init() **throws** ServletException {

**super**.init();

//可直接使用ConfigurableCaptchaService，然后修改配置

*cs* = **new** MyCaptchaService();

}

@Override

**public** **void** destroy() {

*cs* = **null**;

**super**.destroy();

}

**public** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("image/png");

response.setHeader("cache", "no-cache");

HttpSession session = request.getSession(**true**);

OutputStream os = response.getOutputStream();

String patchca= EncoderHelper.*getChallangeAndWriteImage*(*cs*, "png", os);

session.setAttribute("PATCHCA", patchca);

os.flush();

os.close();

}

**public** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

**this**.doGet(request, response);

}

**private** **class** MyCaptchaService **extends** AbstractCaptchaService {

**public** MyCaptchaService() {

//文本内容

wordFactory = **new** MyWordFactory();

//字体

fontFactory = **new** RandomFontFactory();

//效果

textRenderer = **new** BestFitTextRenderer();

//背景

backgroundFactory = **new** SingleColorBackgroundFactory();

//字体颜色

colorFactory = **new** SingleColorFactory(**new** Color(25, 60, 170));

//样式(曲线波纹加干扰线)

filterFactory = **new** CurvesRippleFilterFactory(colorFactory);

//图片长宽

width = 150;

height = 50;

}

}

**private** **class** MyWordFactory **extends** RandomWordFactory {

**public** MyWordFactory() {

//文本范围和长度

characters = "absdekmnowx23456789";

minLength = 5;

maxLength = 4;

}

}

}

1. **包中提供的配置：**

**一：**ConfigurableCaptchaService：

FontFactory 字体

默认：RandomFontFactory（Verdana，Tahoma，minSize、maxSize=45）

WordFactory 单词范围和长度

默认：RandomWordFactory 单词6-6

其他：AdaptiveRandomWordFactory 自适应(mw字母宽度加长)

ColorFactory

默认：SingleColorFactory 黑色

BackgroundFactory

默认：SingleColorBackgroundFactory 白色

TextRenderer 文本渲染

默认：BestFitTextRenderer

FilterFactory 样式

默认：CurvesRippleFilterFactory（曲线波纹带干扰线，干扰线颜色为当前colorFactory）

width/height 图片长宽（默认160\*70）

**使用：**

**public** **class** CaptchaServlet **extends** HttpServlet {

**private** **static** **final** **long** *serialVersionUID* = 4968328161261528097L;

**private** **static** ConfigurableCaptchaService cs = **null**;

**private** **static** ColorFactory cf = **null**;

**private** **static** RandomWordFactory wf = **null**;

**private** **static** Random r = **new** Random();

**private** **static** CurvesRippleFilterFactory crff = **null**;

**private** **static** MarbleRippleFilterFactory mrff = **null**;

**private** **static** DoubleRippleFilterFactory drff = **null**;

**private** **static** WobbleRippleFilterFactory wrff = **null**;

**private** **static** DiffuseRippleFilterFactory dirff = **null**;

@Override

**public** **void** init() **throws** ServletException {

**super**.init();

cs = **new** ConfigurableCaptchaService();

cf = **new** SingleColorFactory(**new** Color(25, 60, 170));

wf = **new** RandomWordFactory();

crff = **new** CurvesRippleFilterFactory(cs.getColorFactory());

drff = **new** DoubleRippleFilterFactory();

wrff = **new** WobbleRippleFilterFactory();

dirff = **new** DiffuseRippleFilterFactory();

mrff = **new** MarbleRippleFilterFactory();

cs.setWordFactory(wf);

cs.setColorFactory(cf);

cs.setWidth(120);

cs.setHeight(50);

}

@Override

**public** **void** destroy() {

wf = **null**;

cf = **null**;

cs = **null**;

**super**.destroy();

}

**public** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("image/png");

response.setHeader("cache", "no-cache");

wf.setMaxLength(5);

wf.setMinLength(3);

HttpSession session = request.getSession(**true**);

OutputStream os = response.getOutputStream();

//随机产生5种效果（filter）

**switch** (r.nextInt(5)) {

**case** 0:

*cs*.setFilterFactory(crff);

**break**;

**case** 1:

*cs*.setFilterFactory(mrff);

**break**;

**case** 2:

*cs*.setFilterFactory(drff);

**break**;

**case** 3:

*cs*.setFilterFactory(wrff);

**break**;

**case** 4:

*cs*.setFilterFactory(dirff);

**break**;

}

String patchca= EncoderHelper.*getChallangeAndWriteImage*(*cs*, "png", os);

session.setAttribute("PATCHCA", patchca);

os.flush();

os.close();

}

}

}

**二：需要自己构造**SimpleCaptchaService(**int** width, **int** height, Color textColor,

Color backgroundColor, **int** fontSize, String[]fontNames, FilterFactory ff) {

backgroundFactory = **new** SingleColorBackgroundFactory(backgroundColor);

wordFactory = **new** AdaptiveRandomWordFactory();

fontFactory = **new** RandomFontFactory(fontNames);

textRenderer = **new** BestFitTextRenderer();

colorFactory = **new** SingleColorFactory(textColor);

filterFactory = ff;

**this**.width = width;

**this**.height = height;

}

1. **FilterFactory样式**



**二：jsp页面访问servlet**

<table><tr>

<!-- *patchca.png和servlet中的<url-pattern>对应* -->

<td><img src=*"patchca.png"* alt=*"验证码"*

style="cursor:*pointer*;vertical-align:*text-bottom*;"

onclick="this.src=this.src+'?'+Math.random();"></td>

<td valign=*"top"*>

<form method=*"POST"*>

<br>

验证码:<input type=*"text"* name=*"patchcafield"*>

<br />

<input type=*"submit"* name=*"submit"*>

</form></td></tr></table>

<br /><br /><br /><br />

**三：jsp页面取出session中的验证码**

<%

//"PATCHCA"和servlet中定义的session保存位置对应

String c = (String)session.getAttribute("PATCHCA");

String parm = (String) request.getParameter("patchcafield");

out.println("Parameter: " + parm + " ? Session Key: " + c + " : ");

**if** (c != **null** && parm != **null**) {

**if** (c.equalsIgnoreCase(parm)) {

out.println("<b>true</b>");

} **else** {

out.println("<b>false</b>");

}

}

%>