



Code

Archive



kaptcha - SpringUsage.wiki

[Export to GitHub](#)

```
<wiki:gadget url="http://kaptcha.googlecode.com/svn/wiki/adsense3.xml" border="0" width="468" height="60" />
```

```
<wiki:gadget url="http://kaptcha.googlecode.com/svn/wiki/donate.xml" border="0" width="150" height="75" />
```

Introduction

Marc Schipperheyn (m.schipperheynNOSPAMMERSgmail.com) gratefully contributed the following documentation on how to use Kaptcha with Spring. I don't personally use Spring so please contact him with any questions you have.

Details

security.xml

```
<bean id="captchaProducer" class="com.google.code.kaptcha.impl.DefaultKaptcha"> <property name="config"> <bean class="com.google.code.kaptcha.util.Config"> <constructor-arg type="java.util.Properties" value="null"></constructor-arg> </bean> </property> </bean>
```

To use properties:

```
<bean id="captchaProducer" class="com.google.code.kaptcha.impl.DefaultKaptcha"> <property name="config"> <bean class="com.google.code.kaptcha.util.Config"> <constructor-arg type="java.util.Properties"> <props> <prop key="kaptcha.image.width">300</prop> <prop key="kaptcha.image.height">25</prop> <prop key="kaptcha.textproducer.char.string">0123456789</prop> <prop key="kaptcha.textproducer.char.length">4</prop> </props> </constructor-arg> </bean> </property> </bean>
```

Kaptcha controller

```

package nl.yourproject.webapp.controller;

import java.awt.image.BufferedImage;

import javax.imageio.ImageIO; import javax.servlet.ServletOutputStream; import
javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired; import
org.springframework.stereotype.Controller; import
org.springframework.web.bind.annotation.RequestMapping; import
org.springframework.web.servlet.ModelAndView;

import com.google.code.kaptcha.Constants; import com.google.code.kaptcha.Producer;

@Controller public class CaptchaImageCreateController { private Producer captchaProducer = null;

@Autowired
public void setCaptchaProducer(Producer captchaProducer) {
    this.captchaProducer = captchaProducer;
}

@RequestMapping("/captcha-image.html")
public ModelAndView handleRequest(
    HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    // Set to expire far in the past.
    response.setDateHeader("Expires", 0);
    // Set standard HTTP/1.1 no-cache headers.
    response.setHeader("Cache-Control", "no-store, no-cache, must-revalidate");
    // Set IE extended HTTP/1.1 no-cache headers (use addHeader).
    response.addHeader("Cache-Control", "post-check=0, pre-check=0");
    // Set standard HTTP/1.0 no-cache header.
    response.setHeader("Pragma", "no-cache");

    // return a jpeg
    response.setContentType("image/jpeg");

    // create the text for the image
    String capText = captchaProducer.createText();

    // store the text in the session
    request.getSession().setAttribute(Constants.KAPTCHA_SESSION_KEY, capText);

    // create the image with the text
    BufferedImage bi = captchaProducer.createImage(capText);

    ServletOutputStream out = response.getOutputStream();

    // write the data out
    ImageIO.write(bi, "jpg", out);
    try
    {
        out.flush();
    }
}

```

```

        finally
        {
            out.close();
        }
        return null;
    }
}'''

```

Kaptcha check in regular controller

```

''' package nl.yourproject.webapp.controller; import java.util.HashMap; import java.util.Map;

import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse;

import nl.msw.compraventa.VO.RegistrationVO; import
nl.msw.compraventa.validator.RegistrationValidator;

import org.apache.commons.lang.StringUtils; import org.apache.commons.logging.Log; import
org.apache.commons.logging.LogFactory; import org.springframework.beans.BeanUtils; import
org.springframework.beans.factory.annotation.Autowired; import
org.springframework.stereotype.Controller; import org.springframework.ui.ModelMap; import
org.springframework.validation.BindingResult; import org.springframework.validation.Errors; import
org.springframework.web.bind.annotation.ModelAttribute; import
org.springframework.web.bind.annotation.RequestMapping; import
org.springframework.web.bind.annotation.RequestMethod;

import com.google.code.kaptcha.Constants; import com.google.code.kaptcha.Producer;

@Controller @RequestMapping("/register.html") public class RegistrationFormController {

    private Log log = LogFactory.getLog(RegistrationFormController.class);
    private Producer captchaProducer;

    @Autowired
    public void setCaptchaService(Producer captchaProducer) {
        this.captchaProducer = captchaProducer;
    }

    @RequestMapping(method = RequestMethod.GET)
    public String setupForm(
        ModelMap model,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception {
        RegistrationVO vo = new RegistrationVO();
        model.addAttribute("vo", vo);

        return "registrationform";
    }

    @RequestMapping(method = RequestMethod.POST)
    public String processForm(
        @ModelAttribute("vo")

```

```

        RegistrationVO vo,
        BindingResult result,
        ModelMap model,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception {

        validateCaptcha(request, (Object) vo, (Errors) result);

//handle your form
return "registrationform";
}

/**
 * Validate the captcha response
 */
protected void validateCaptcha(
        HttpServletRequest request,
        Object command,
        Errors errors) throws Exception {
String captchaId = (String) request.getSession().getAttribute(Constants.KAPTCHA_SESSION_KEY);
String response = ((RegistrationVO) command).getCaptchaResponse();

if (log.isDebugEnabled()) {
    log.debug("Validating captcha response: '" + response + "'");
}

if (!StringUtils.equalsIgnoreCase(captchaId, response)) {
    errors.rejectValue("captchaResponse", "error.invalidcaptcha",
        "Invalid Entry");
}
}

}'''

```

```

<wiki:gadget url="http://kaptcha.googlecode.com/svn/wiki/adsense3.xml" border="0" width="470"
height="60" />

```