

목차

1.	Captcha 종류.....	2
2.	google recaptcha 자동가입방지	3
2.1	구글 리캡차(Google ReCaptcha) 적용 절차	3
2.2	동작 방식.....	3
2.3	구글 리캡차 api key 발급	3
2.4	build.gradle 에 dependency 추가.....	4
2.5	jsp 작성	4
2.6	자바스크립트 메서드 작성(본인은 ajax 처리).....	4
2.7	리캡차 인증 코드 소스 작성	5
2.8	controller 메서드 구현부 작성.....	6
2.9	Reference	7
3.	Simplecaptcha 를 이용한 자동 가입 방지	8
3.1	라이브러리 추가	8
3.2	build.gradle 에 dependency 추가.....	9
3.3	프로세스.....	9
3.4	CaptchaServlet 클래스 생성	9
3.5	web.xml 에 servlet 등록.....	10
3.6	Controller 메서드 생성	11
3.7	view 페이지 생성	12
3.8	Reference	13
4.	직접 구현하기.....	14
4.1	Captcha 클래스 구현.....	14
4.2	이미지 생성 코드	16
4.3	확인 코드.....	17
4.4	Reference	17

1. Captcha 종류

- ✓ google recaptcha 방식
- ✓ Simplecaptcha
- ✓ 직접 구현

2. google recaptcha 자동가입방지

2.1 구글 리캡차(Google ReCaptcha) 적용 절차

1. 리캡차 개발자 페이지에서 캡차를 등록한다.

<https://developers.google.com/recaptcha/docs/start>

2. 등록이 완료되면, 공개키와 비밀키가 발급된다.

공개키는 클라이언트에서 요청을 보내는 용도로, 비밀키는 서버에서 검증하는 용도로 사용된다.

3. 가이드를 보고, 서버와 클라이언트에 코드 스니펫을 적용하면 된다.

2.2 동작 방식

1. 클라이언트 API 가 캡차를 로드할 때, 공개키를 파라미터로 전달한다.

1.1. 서버에서는 세션을 생성하고, 1 회성 토큰을 내려준다.

2. 사용자가 `I'm not a robot`의 체크박스를 클릭한다.

2.1. 이 때 클라이언트 API 는 백그라운드 액션을 데이터로 만드는 것 같아 보인다...

2.2. 인터랙션을 보고 사람이라고 판단했다면, 공개키와 토큰과 데이터를 캡차 서버에 전달한다.

2.3. 캡차 서버에서는 토큰과 데이터의 값을 기반으로 새 값을 만들어 세션에 저장한다.

2.4. 생성한 값을 공개키로 암호화한 후 응답 토큰으로 내려준다.

2.5. 클라이언트 API 에서는 응답 토큰을 `

3. 캡차를 포함한 폼을 애플리케이션 서버로 서브밋한다. 이 때, 응답 토큰을 포함해 보낸다.

3.1. 애플리케이션 서버에서는 발급받은 비밀키와 응답 토큰으로 캡차 서버에 인증 요청을 보낸다.

3.2. 캡차 서버에서는 파라미터로 받은 비밀키로 응답 토큰을 복호화하고, 세션에 저장된 값과 비교해 응답한다.

3.3. 응답이 유효하면 성공!

2.3 구글 리캡차 api key 발급

<https://www.google.com/recaptcha/admin#list>

label : 이름

domain : URL 입력 (로컬이라면 localhost 입력)

owner : 소유자 이메일

발급받은 후 sitekey, secretkey 복사

2.4 build.gradle 에 dependency 추가

```
dependencies {
    // https://mvnrepository.com/artifact/net.tanesha.recaptcha4j/recaptcha4j
    compile group: 'net.tanesha.recaptcha4j', name: 'recaptcha4j', version: '0.0.8'
}
```

2.5 jsp 작성

```
<html>
<head>
<title>reCAPTCHA demo: Simple page</title>
<script src="https://www.google.com/recaptcha/api.js" async defer></script>
</head>
<body>
<form action="?" method="POST">
<div class="g-recaptcha" data-sitekey="${siteKey}"></div>
<br/>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

siteKey 는 발급받은 것 적으면 됨

2.6 자바스크립트 메서드 작성(본인은 ajax 처리)

```
$.ajax({
    url : '/member/captchaAjax',
    type : 'post',
    data : {
        recaptcha : document.getElementById("g-recaptcha-response").value
    },
    success : function(data){
        //성공여부
    }
});
```

2.7 리캡차 인증 코드 소스 작성

```
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.StringReader;
import java.net.URL;

import javax.json.Json;
import javax.json.JsonObject;
import javax.json.JsonReader;
import javax.net.ssl.HttpURLConnection;

public class VerifyRecaptcha {

    public static final String url = "https://www.google.com/recaptcha/api/siteverify";
    private final static String USER_AGENT = "Mozilla/5.0";
    private static String secret = ""; //local

    public static void setSecretKey(String key){
        secret = key;
    }

    public static boolean verify(String gRecaptchaResponse) throws IOException {
        if (gRecaptchaResponse == null || "".equals(gRecaptchaResponse)) {
            return false;
        }

        try{
            URL obj = new URL(url);
            HttpURLConnection con = (HttpURLConnection) obj.openConnection();

            // add request header
            con.setRequestMethod("POST");
            con.setRequestProperty("User-Agent", USER_AGENT);
            con.setRequestProperty("Accept-Language", "en-US,en;q=0.5");

            String postParams = "secret=" + secret + "&response="
                + gRecaptchaResponse;

            // Send post request
            con.setDoOutput(true);
            DataOutputStream wr = new DataOutputStream(con.getOutputStream());
            wr.writeBytes(postParams);
```

```

        wr.flush();
        wr.close();

        int responseCode = con.getResponseCode();
        //System.out.println("\nSending 'POST' request to URL : " + url);
        // System.out.println("Post parameters : " + postParams);
        // System.out.println("Response Code : " + responseCode);

        BufferedReader in = new BufferedReader(new InputStreamReader(
            con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();

        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();

        // print result
        System.out.println(response.toString());

        //parse JSON response and return 'success' value
        JsonReader jsonReader = Json.createReader(new StringReader(response.toString()));
        JsonObject jsonObject = jsonReader.readObject();
        jsonReader.close();

        return jsonObject.getBoolean("success");
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}
}

```

2.8 controller 메서드 구현부 작성

```

//secretKey 세팅
VerifyRecaptcha.setSecretKey(secretKey);

//recapcha 파라미터 가져오기
String gRecaptchaResponse = request.getParameter("recaptcha");

//리캡차 인증 true,false 성공 실패 리턴

```

```
boolean verify = VerifyRecaptcha.verify(gRecaptchaResponse);
String result = "fail";
if(verify){
    result = "success";
}

ModelAndView view = new ModelAndView();
view.addObject("result",result);
view.setViewName("jsonView");
return view;
```

2.9 Reference

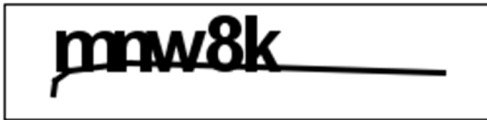
<http://nkcnow.tistory.com/157>

<http://huskdoll.tistory.com/114>

3. Simplecaptcha 를 이용한 자동 가입 방지

simplecaptcha 를 이용해서 자동 가입 및 자동 로그인을 방지하는 방법을 살펴보겠다.

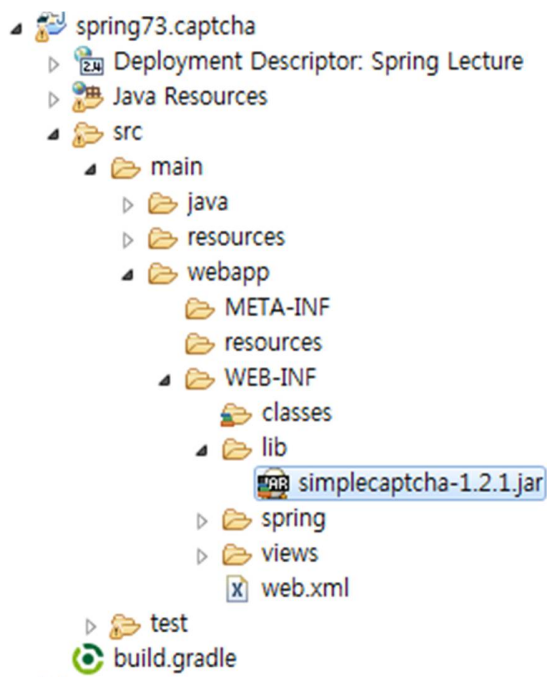
- ✓ simplecaptcha 방식은 servlet 을 사용하여 작동 한다.
- ✓ img 태그의 src 속성을 이용하여 url 이동 없이 이미지를 새로고침 하기 위함



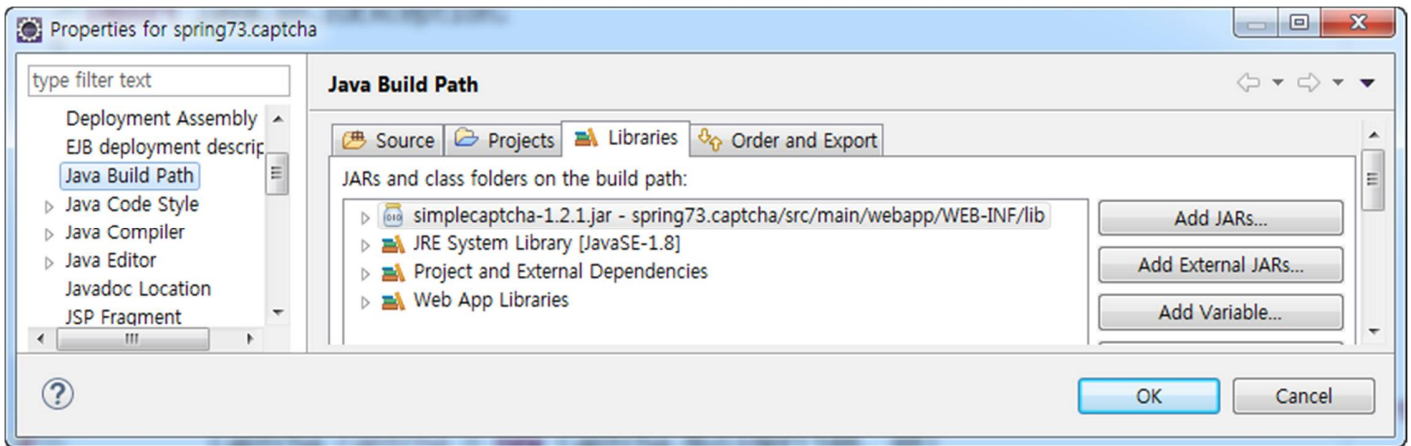
3.1 라이브러리 추가

simplecaptcha 2016.10 월 기준으로 현재 maven 으로 다운로드 되지 않기 때문에 아래 방법 사용

1. simplecaptcha-1.2.1.jar 파일 다운로드
<http://mvnrepository.com/artifact/nl.captcha/simplecaptcha> 에서 다운로드
2. src/main/webapp/WEB-INF/lib 폴더 생성 후 jar 파일 이동



3. 빌드 패스 추가



3.2 build.gradle 에 dependency 추가

dependencies {

```
// https://mvnrepository.com/artifact/nl.captcha/simplecaptcha
compile group: 'nl.captcha', name: 'simplecaptcha', version: '1.2.1'
}
```

3.3 프로세스

1. view 페이지 img 태그의 src 속성에 OCR 전용 servlet 에 mapping 된 url 설정
2. OCR 전용 servlet 에서 코드 생성 및 session 의 Attribute 에 저장 및 이미지 생성
3. response 된 이미지 를 view 페이지의 img 태그가 표현

3.4 CaptchaServlet 클래스 생성

- ✓ 이미지 요청 url 을 처리할 OCR 전용 servlet 생성
- ✓ HttpServlet 을 상속 받아서 CaptchaServlet 클래스 생성한다.

```
import java.io.IOException;

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

import nl.captcha.Captcha;
import nl.captcha.servlet.CaptchaServletUtil;

public class CaptchaServlet extends HttpServlet {
```

```

private static final long serialVersionUID = 1L;

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    Captcha captcha = new Captcha.Builder(148, 48)
        .addText() // default: 5 개의 숫자+문자
        .addNoise().addNoise().addNoise() // 시야 방해 라인 3 개
        .addBackground() // 기본 하얀색 배경
        .build();

    response.setHeader("Cache-Control", "no-cache");
    response.setDateHeader("Expires", 0);
    response.setHeader("Pragma", "no-cache");
    response.setDateHeader("Max-Age", 0);
    response.setContentType("image/png");
    CaptchaServletUtil.writeImage(response, captcha.getImage()); // 이미지 그리기
    request.getSession().setAttribute("captcha", captcha.getAnswer()); // 값 저장
}
}

```

3.5 web.xml 에 servlet 등록

- ✓ 기존 사용하던 DispatcherServlet 과 별도로 추가 mapping
- ✓ com.spring73.captcha 는 실제 패키지명으로 맞게 바꾸어야 한다.

```

<!-- Captcha request -->
<servlet>
    <servlet-name>captcha</servlet-name>
    <servlet-class>com.spring73.captcha.CaptchaServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>captcha</servlet-name>
    <url-pattern>/captcha</url-pattern>
</servlet-mapping>

```

자신의 패키지명으로 맞게 바꾸어야 한다.

```

web.xml
23
24 <!-- Processes application requests -->
25 <servlet>
26     <servlet-name>appServlet</servlet-name>
27     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
28     <init-param>
29         <param-name>contextConfigLocation</param-name>
30         <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
31     </init-param>
32     <load-on-startup>1</load-on-startup>
33 </servlet>
34 <servlet-mapping>
35     <servlet-name>appServlet</servlet-name>
36     <url-pattern>/</url-pattern>
37     <!-- <url-pattern>*.do</url-pattern> -->
38 </servlet-mapping>
39
40 <!-- Captcha request -->
41 <servlet>
42     <servlet-name>captcha</servlet-name>
43     <servlet-class>servlet.CaptchaServlet</servlet-class>
44     <load-on-startup>1</load-on-startup>
45 </servlet>
46 <servlet-mapping>
47     <servlet-name>captcha</servlet-name>
48     <url-pattern>/captcha</url-pattern>
49 </servlet-mapping>
50

```

3.6 Controller 메서드 생성

- ✓ view 페이지에서 넘어온 값(captcha)를 확인할

```

// 글 작성 captcha 테스트
@RequestMapping(value = "/write", method = RequestMethod.GET)
public String boardWrite( Model model) {

    return "write";
}

// captcha 테스트
@RequestMapping(value = "/write", method = RequestMethod.POST)
public String boardWrite( Model model
                        , HttpSession session
                        , @RequestParam String captcha      ) {

    String captchaValue = (String) session.getAttribute("captcha");

    if (captcha == null || !captchaValue.equals(captcha)) {

```

```

    return "redirect:/write";
}

return "redirect:/";
}

```

3.7 view 페이지 생성

- captcha 새로그침시 param 을 넘기는 이유

- 1) ie에서는 param 을 넘기지 않을시 이미지를 새로그침 하지 않음
- 2) 그 외 브라우저는 param 을 넘기지 않아도 정상 작동함
- 3) 즉, 아무 의미없는 param 이다.

```

<form action="./write" id="writeForm" method="post">
  <fieldset>
    <legend class="screen_out">게시글 작성 폼</legend>

    <div class="box captcha">
      <div class="loading"></div>
      <!-- // .loading -->

      <label for="captcha">자동 방지 코드</label>
      <input type="text" id="captcha" name="captcha" autocomplete="off" required />
      
    </div>
    <!-- // .box.captcha -->

    <div class="box btn">
      <button type="reset" class="btn write">취소</button>
      <button type="submit" class="btn write">작성 완료</button>
    </div>
  </fieldset>
</form>

<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
  var $captchaLoader = $("#writeForm .box.captcha .loading");
  var $captchaImg = $("#writeForm .box.captcha img");

  // captcha 새로그침
  $captchaImg.click(function() {
    $captchaLoader.show();
    $(this).attr("src", "/captcha?ran=" + Math.random());
  });

```

```
$captchaLoader.fadeOut(500);  
});  
</script>
```

3.8 Reference

- ✓ <http://simplecaptcha.sourceforge.net>
- ✓ <http://winflahed.tistory.com/132>
- ✓ <https://github.com/ndkhoiits/Simplecaptcha>

4. 직접 구현하기

4.1 Captcha 클래스 구현

미리 저장된 문자열을 사용하게 되어 있었는데, 6 자리 문자열을 랜덤하게 만들도록 수정했다. generateKeyText 메서드가 원본, generateKeyRandomText 메서드가 수정본이다. 사용하는 것은 간단하다.

```
import java.awt.Color;
import java.awt.Font;
import java.awt.GradientPaint;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.io.OutputStream;
import java.util.Random;

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

public class Captcha {

    public static void makeTextImage( HttpServletRequest request
                                     , HttpServletResponse response)
                                     throws ServletException, IOException {

        int width = 150;
        int height = 50;

        BufferedImage bufferedImage = new BufferedImage(width, height,
        BufferedImage.TYPE_INT_RGB);

        Graphics2D g2d = bufferedImage.createGraphics();

        Font font = new Font("Georgia", Font.BOLD, 18);
        g2d.setFont(font);

        RenderingHints rh = new RenderingHints(
        RenderingHints.KEY_ANTIALIASING,
```

```

RenderingHints.VALUE_ANTIALIAS_ON);

rh.put(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY);

g2d.setRenderingHints(rh);

GradientPaint gp = new GradientPaint(0, 0,
Color.green, 0, height/2, Color.black, true);

g2d.setPaint(gp);
g2d.fillRect(0, 0, width, height);

g2d.setColor(new Color(255, 153, 0));

char[] data = generateKeyRandomText();
String captcha = String.valueOf(data);
request.getSession().setAttribute("captcha", captcha );

int x = 0;
int y = 0;

Random r = new Random();
for (int i=0; i<data.length; i++) {
x += 10 + (Math.abs(r.nextInt()) % 15);
y = 20 + Math.abs(r.nextInt()) % 20;
g2d.drawChars(data, i, 1, x, y);
}

g2d.dispose();

response.setContentType("image/png");
OutputStream os = response.getOutputStream();
ImageIO.write(bufferedImage, "png", os);
os.close();
}

private static char[] generateKeyText(){
char data[][] = {
    { 'z', 'e', 't', 'c', 'o', 'd', 'e' },
    { 'l', 'i', 'n', 'u', 'x' },
    { 'f', 'r', 'e', 'e', 'b', 's', 'd' },
    { 'u', 'b', 'u', 'n', 't', 'u' },
    { 'j', 'e', 'e' }
}

```

```

    };

    Random r = new Random();
    int index = Math.abs(r.nextInt()) % 5;

    return data[index];
}

private static char[] generateKeyRandomText(){
    char data[] = {0, 0, 0, 0, 0, 0};

    char alphaCnt = ('Z'-'A'+1);
    char numCnt = ('9'-'0'+1);
    char totalCnt = (char)((alphaCnt * 2) + numCnt);

    char zero = '0';
    char lowerA = 'a';
    char upperA = 'A';

    Random r = new Random();
    for(int i=0; i<data.length; i++){
        int index = Math.abs(r.nextInt()) % totalCnt;

        if(index < numCnt){
            data[i] = (char) (zero + index);
        }else if(index < (numCnt + alphaCnt)){
            data[i] = (char) (upperA + index - numCnt);
        }else{
            data[i] = (char) (lowerA + index - numCnt - alphaCnt);
        }
    }

    return data;
}
}

```

4.2 이미지 생성 코드.

```

@RequestMapping("/capchaTest")
protected void capchaTest( HttpServletRequest request
                           , HttpServletResponse response)
    throws ServletException, IOException {
    Captcha.makeTextImage(request, response);
}

```


4.3 확인 코드

```
String captcha = (String) session.getAttribute("captcha");
String code = (String) request.getParameter("code");

if (captcha != null && code != null) {
    if (captcha.equals(code)) {
        out.print("<p class='alert'>Correct</p>");
    } else {
        out.print("<p class='alert'>Incorrect</p>");
    }
}
```

4.4 Reference

<http://zetcode.com/tutorials/jaetutorials/captcha/>

<http://junemoon.tistory.com/132>