



AUTENTICACIÓN CON REDES SOCIALES

Utilizando la librería ScribeJava

Software Factory
2016-1

Roy Alejo Taza Rojas
U201213734@upc.edu.pe

Contenido

INTRODUCCION	2
INSTALACIÓN.....	2
IMPLEMENTACIÓN	2
Creación del proyecto	2
Creando la página de Inicio	2
Utilizando ScribeJava.....	4
Uso rápido	4
Aplicando ScribeJava en nuestro proyecto web	5
CONCLUSIONES	13

INTRODUCCION

ScribeJava es una librería de Java que sirve si para que la aplicación web utilice la autenticación de usuarios a través de proveedores externos OAuth/OAuth2 como Gmail, Hotmail, Yahoo, Twitter, Facebook, LinkedIn, Foursquare, MySpace, Salesforce, Yammer y con ello el posterior registro del usuario sea fácil. Todo lo que se necesita hacer es crear una página donde los usuarios puedan hacer clic en los botones para los proveedores con los que puede loguearse. Luego, sólo se tiene que llamar a ScribeJava y se pueden obtener todos los datos de su perfil.

INSTALACIÓN

Se puede descargar el jar manualmente desde el [centro de descargas](#) e incluir [apache commons doc](#), o simplemente dejar que maven descargue todas las dependencias añadiendo en el archivo pm.xml lo siguiente:

```
<dependency>
  <groupId>com.github.scribejava</groupId>
  <artifactId>scribejava-apis</artifactId>
  <version>2.5.3</version>
</dependency>
```

IMPLEMENTACIÓN

Creación del proyecto

Trabajaremos con un proyecto **Maven Web Application** de nombre **ScribeJavaDemo**. Instalaremos ScribeJava y adicionalmente **org.json** para poder leer fácilmente la respuesta de los proveedores:

```
<dependency>
  <groupId>org.json</groupId>
  <artifactId>json</artifactId>
  <version>20160212</version>
</dependency>
```

Asimismo añadiremos un servlet de nombre "ScribeJavaServlet" que procesará las acciones.

Creando la página de Inicio

En nuestra página de inicio tendremos lo siguiente:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Login</title>
  </head>
  <body>
    <h1>Iniciar Sesión con redes sociales</h1>
    <table>
      <tr>
        <td>
          <form id="formFb" action="ScribeJavaServlet" method="post">
            <input type="hidden" value="loginFb" name="accion"/>
            <input type="image" src="img/Facebook.png" alt="Submit Form" height="50"
width="50" />
          </form>
```

```

</td>

<td>
  <form id="formTwitter" action="ScribeJavaServlet" method="post">
    <input type="hidden" value="loginTwitter" name="accion"/>
    <input type="image" src="img/Twitter.png" alt="Submit Form" height="50"
width="60" />
  </form>

</td>

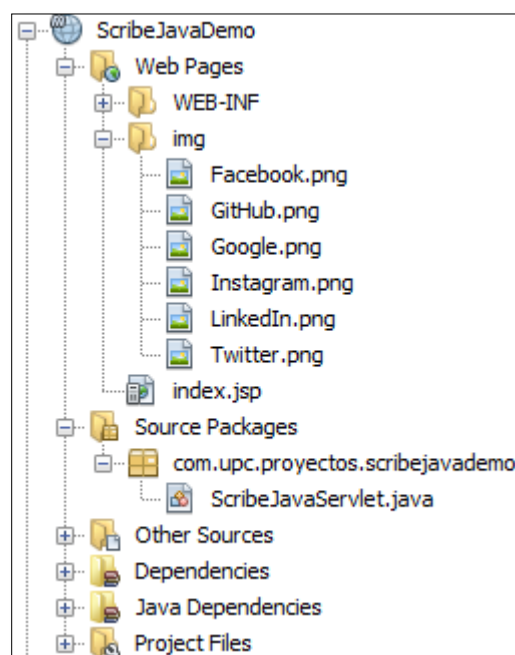
<td>
  <form id="formGoogle" action="ScribeJavaServlet" method="post">
    <input type="hidden" value="loginGoogle" name="accion"/>
    <input type="image" src="img/Google.png" alt="Submit Form" height="50"
width="50" />
  </form>

</td>

<td>
  <form id="formLinked" action="ScribeJavaServlet" method="post">
    <input type="hidden" value="loginLinkedIn" name="accion"/>
    <input type="image" src="img/LinkedIn.png" alt="Submit Form" height="50"
width="50" />
  </form>
</td>
</tr>
</table>
</body>
</html>

```

La estructura de nuestro proyecto es la siguiente:



Como resultado nuestra página de inicio será la siguiente:



Veremos que cada botón en nuestra página de inicio enviará una petición POST al servlet ScribeJavaServlet junto al valor de una acción.

El servlet recibirá y ejecutará la acción recibida de acuerdo al proveedor enviado y recibirá la respuesta del proveedor señalado por el usuario.

Utilizando ScribeJava

Uso rápido

En esta parte veremos los pasos para utilizar ScribeJava. En el siguiente punto veremos cómo lo utilizaremos en nuestro proyecto web.

1. Crear un objeto OAuthService

Los elementos que necesitaremos y que nos da el proveedor para poder conectarnos son:

- Api key: identificador de la aplicación
- Api secret: código de seguridad
- Callback url: url de re-direccionamiento. El proveedor re-direccionará a esta URL el resultado de la acción del usuario.

```
OAuth20Service service20 = new ServiceBuilder()  
    .apiKey("your_api_key ")  
    .apiSecret("your_api_secret ")  
    .callback("redirect_url")  
    .build(FacebookApi.instance());
```

2. Hacer que el usuario inicie sesión en el proveedor

Debemos hacer que usuario autorice la aplicación. Para ello utilizamos lo siguiente.

```
String authURL = service20.getAuthorizationUrl();
```

Obtenemos la dirección url donde el usuario debe autenticarse, lo siguiente será redireccionar:

```
response.sendRedirect(authURL);
```

El usuario ingresará sus credenciales y autorizará la aplicación.

3. Obtener el token de acceso

Una vez que el usuario ya se autenticó, el proveedor nos re-direccionará a la url que está configurada, se debe intercambiar el requestToken por un accessToken con el cual se harán las peticiones:

```
String code = request.getParameter("code");
accessToken2 = service20.getAccessToken(code);
```

4. Petición

Ya está todo listo, entonces hacemos la petición al proveedor:

```
OAuthRequest oauthRequest = new OAuthRequest(Verb.GET, "URL_API_SERVICE",
service20);
service20.signRequest(accessToken2, oauthRequest);

Response respuestaAPI = oauthRequest.send();
// getBody() devolverá una cadena de texto que se puede convertir a JSONObject
System.out.println(respuestaAPI.getBody());
```

Aplicando ScribeJava en nuestro proyecto web

Ahora que sabemos cómo es el uso de ScribeJava, lo aplicaremos a nuestro proyecto.

En nuestro servlet definiremos las siguientes propiedades estáticas:

```
private String loginProvider;

private OAuth10aService service10;
private OAuth20Service service20;

private OAuth1RequestToken requestToken;

private OAuthRequest oauthRequest;
private Response respuestaAPI;

private OAuth1AccessToken accessToken1;
private OAuth2AccessToken accessToken2;
```

- loginProvider: servirá para guardar la elección de proveedor del usuario.
- OAuth10aService: para proveedores que utilicen OAuth 1a.
- OAuth20Service: para proveedores que utilicen OAuth 2.
- OAuth1RequestToken: para almacenar el token que se requiere para las peticiones de proveedores OAuth 1a.
- OAuth1AccessToken: el token de acceso OAuth 1a.
- OAuth2AccessToken: el token de acceso OAuth 2.
- OAuthRequest: inicializar la respuesta del uso de api del proveedor.
- Response: la respuesta del proveedor, lo que buscamos.

En el **processRequest** de nuestro servlet pondremos lo siguiente:

```
if (request.getParameter("accion") != null) {
    String authURL = "http://localhost:8080/ScribeJavaDemo";
```

```

        if ("loginFb".equalsIgnoreCase(request.getParameter("accion"))) {
            loginProvider = "Facebook";
            service20 = new ServiceBuilder()
                .apiKey("201803806879061")
                .apiSecret("6cc1966c7ed62527dfccc580e7dc6b95")
                .callback("http://localhost:8080/ScribeJavaDemo/ScribeJavaServlet")
                .build(FacebookApi.instance());

            authURL = service20.getAuthorizationUrl();
        } else if ("loginTwitter".equalsIgnoreCase(request.getParameter("accion"))) {
            loginProvider = "Twitter";
            service10 = new ServiceBuilder()
                .apiKey("frB4Ap5vaDZ2IECXxBEIXlpRv")
                .apiSecret("6G3HONucjfNjWMYjr1vLHSpqNV2GWPCmSwZ1scmMKWrVYp6OrE")
                .callback("http://localhost:8080/ScribeJavaDemo/ScribeJavaServlet")
                .build(TwitterApi.instance());

            requestToken = service10.getRequestToken();
            authURL = service10.getAuthorizationUrl(requestToken);
        } else if ("loginGoogle".equalsIgnoreCase(request.getParameter("accion"))) {
            loginProvider = "Google";
            service20 = new ServiceBuilder()
                .apiKey("803104469809-djkqgh5e31aeb0ifotei46f8mve7q3j.apps.googleusercontent.com")
                .apiSecret("kEM9427g8V5J_8zXmXhnke-_")
                .callback("http://localhost:8080/ScribeJavaDemo/ScribeJavaServlet")
                .scope("openid email profile")
                .build(GoogleApi20.instance());

            authURL = service20.getAuthorizationUrl();
        } else if ("loginLinkedIn".equalsIgnoreCase(request.getParameter("accion"))) {
            loginProvider = "LinkedIn";
            service20 = new ServiceBuilder()
                .apiKey("7805y2q13pzk59")
                .apiSecret("nle9uHcJzhyt71fN")
                .callback("http://localhost:8080/ScribeJavaDemo/ScribeJavaServlet")
                .build(LinkedInApi20.instance());

            authURL = service20.getAuthorizationUrl();
        }
        System.out.println("URL: " + authURL);
        response.sendRedirect(authURL);
    }
}

```

Se lee el parámetro acción para saber el proveedor a utilizar, se setea loginProvider, se inicializa el **ServiceBuilder**, se obtiene la url de autenticación y se redirecciona. En el else de la sentencia else irá lo siguiente:

```

else {
    String code = request.getParameter("code");
    switch (loginProvider) {
        case "Facebook":

```

```

        accessToken2 = service20.getAccessToken(code);

        oauthRequest = new OAuthRequest(Verb.GET,
"https://graph.facebook.com/me?fields=id,name,email,age_range,birthday,gender,link,cover", service20);
        service20.signRequest(accessToken2, oauthRequest);
        break;
        case "Twitter":
            String verifier = request.getParameter("oauth_verifier");
            accessToken1 = service10.getAccessToken(requestToken, verifier);

            oauthRequest = new OAuthRequest(Verb.GET,
"https://api.twitter.com/1.1/account/verify_credentials.json?include_email=true",
service10);
            service10.signRequest(accessToken1, oauthRequest);
            break;
            case "Google":
                accessToken2 = service20.getAccessToken(code);

                oauthRequest = new OAuthRequest(Verb.GET,
"https://www.googleapis.com/userinfo/v2/me", service20);
                service20.signRequest(accessToken2, oauthRequest);
                break;
                case "LinkedIn":
                    accessToken2 = service20.getAccessToken(code);

                    oauthRequest = new OAuthRequest(Verb.GET,
"https://api.linkedin.com/v1/people/~:(id,firstName,last-name,headline,picture-url,picture-
urls::(original),siteStandardProfileRequest,public-profile-url,email-address)?format=json",
service20);
                    service20.signRequest(accessToken2, oauthRequest);
                    break;
            }
            respuestaAPI = oauthRequest.send();
            System.out.println(respuestaAPI.getBody());
            escribir(response, respuestaAPI.getBody());
        }
    }

```

Se verifica que proveedor estamos utilizando, se obtiene el access token, se configura la petición a la API del proveedor, se envía la petición y se obtiene la respuesta. Adicionalmente, la respuesta es enviada a una función dentro servlet llamada **escribir**.

```

private void escribir(HttpServletResponse response, String responseText) throws IOException
{
    JSONObject responseJSON = new JSONObject(responseText);

    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Usuario Autenticado</title>");
    }
}

```



```

out.println("</head>");
out.println("<body>");
out.println("<h1>Bienvenido</h1>");
out.println("<p>A continuación se visualizará la información del usuario</p>");
out.println("<h3>Información del usuario</h3>");
out.println("<ul id='userInfo'>");
String id = "";
String nombre = "";
String email = "";
switch (loginProvider) {
    case "Facebook":
        id = responseJSON.getString("id");
        nombre = responseJSON.getString("name");
        email = responseJSON.getString("email");
        break;
    case "Twitter":
        id = responseJSON.get("id").toString();
        nombre = responseJSON.getString("name");
        email = responseJSON.getString("email");
        break;
    case "Google":
        id = responseJSON.getString("id");
        nombre = responseJSON.getString("name");
        email = responseJSON.getString("email");
        break;
    case "LinkedIn":
        id = responseJSON.getString("id");
        nombre = responseJSON.getString("firstName") + " " +
responseJSON.getString("lastName");
        email = responseJSON.getString("emailAddress");
        break;
}
out.println("<li>ID: " + id + "</li>");
out.println("<li>Nombre: " + nombre + "</li>");
out.println("<li>Email: " + email + "</li>");
out.println("</ul>");

out.println("<h4>Foto de Perfil</h4>");
String srcImage = "";
switch (loginProvider) {
    case "Facebook":
        //srcImage=responseJSON.getString("pictureUrl");
        break;
    case "Twitter":
        srcImage = responseJSON.getString("profile_image_url");
        break;
    case "Google":
        srcImage = responseJSON.getString("picture");
        break;
    case "LinkedIn":
        srcImage = responseJSON.getString("pictureUrl");
        break;
}

```

```

    }

    out.println("<img id='userPhotoPerfil' src='" + srcImage + "'>");
    out.println("<h4>Respuesta JSON Completa</h4>");
    out.println("<p>" + responseJSON.toString() + "</p>");

    out.println("</body>");
    out.println("</html>");
  }
}

```

La función **processRequest** de nuestro servlet quedará de la siguiente manera:

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    try {
        if (request.getParameter("accion") != null) {
            String authURL = "http://localhost:8080/ScribeJavaDemo";
            if ("loginFb".equalsIgnoreCase(request.getParameter("accion"))) {
                loginProvider = "Facebook";
                service20 = new ServiceBuilder()
                    .apiKey("201803806879061")
                    .apiSecret("6cc1966c7ed62527dfccc580e7dc6b95")
                    .callback("http://localhost:8080/ScribeJavaDemo/ScribeJavaServlet")
                    .build(FacebookApi.instance());

                authURL = service20.getAuthorizationUrl();
            } else if ("loginTwitter".equalsIgnoreCase(request.getParameter("accion"))) {
                loginProvider = "Twitter";
                service10 = new ServiceBuilder()
                    .apiKey("frB4Ap5vaDZ2IECXxBEIXlpRv")
                    .apiSecret("6G3HONucjfNjWMYjr1vLHSpqNV2GWPCmSwZ1scmMKWrVYp6OrE")
                    .callback("http://localhost:8080/ScribeJavaDemo/ScribeJavaServlet")
                    .build(TwitterApi.instance());

                requestToken = service10.getRequestToken();
                authURL = service10.getAuthorizationUrl(requestToken);
            } else if ("loginGoogle".equalsIgnoreCase(request.getParameter("accion"))) {
                loginProvider = "Google";
                service20 = new ServiceBuilder()
                    .apiKey("803104469809-djkqgh5e31aeb0ifotei46f8mve7q3j.apps.googleusercontent.com")
                    .apiSecret("kEM9427g8V5J_8zXmXhnke-_")
                    .callback("http://localhost:8080/ScribeJavaDemo/ScribeJavaServlet")
                    .scope("openid email profile")
                    .build(GoogleApi20.instance());

                authURL = service20.getAuthorizationUrl();
            } else if ("loginLinkedIn".equalsIgnoreCase(request.getParameter("accion"))) {
                loginProvider = "LinkedIn";
            }
        }
    }
}

```

```

        service20 = new ServiceBuilder()
            .apiKey("7805y2q13pzk59")
            .apiSecret("nle9uHcJzhyt71fN")
            .callback("http://localhost:8080/ScribeJavaDemo/ScribeJavaServlet")
            .build(LinkedInApi20.instance());

        authURL = service20.getAuthorizationUrl();
    }
    System.out.println("URL: " + authURL);
    response.sendRedirect(authURL);
} else {
    String code = request.getParameter("code");
    switch (loginProvider) {
        case "Facebook":
            accessToken2 = service20.getAccessToken(code);

            oauthRequest = new OAuthRequest(Verb.GET,
"https://graph.facebook.com/me?fields=id,name,email,age_range,birthday,gender,link,cov
er", service20);
            service20.signRequest(accessToken2, oauthRequest);
            break;
        case "Twitter":
            String verifier = request.getParameter("oauth_verifier");
            accessToken1 = service10.getAccessToken(requestToken, verifier);

            oauthRequest = new OAuthRequest(Verb.GET,
"https://api.twitter.com/1.1/account/verify_credentials.json?include_email=true",
service10);
            service10.signRequest(accessToken1, oauthRequest);
            break;
        case "Google":
            accessToken2 = service20.getAccessToken(code);

            oauthRequest = new OAuthRequest(Verb.GET,
"https://www.googleapis.com/userinfo/v2/me", service20);
            service20.signRequest(accessToken2, oauthRequest);
            break;
        case "LinkedIn":
            accessToken2 = service20.getAccessToken(code);

            oauthRequest = new OAuthRequest(Verb.GET,
"https://api.linkedin.com/v1/people/~:(id,firstName,last-name,headline,picture-url,picture-
urls::(original),siteStandardProfileRequest,public-profile-url,email-address)?format=json",
service20);
            service20.signRequest(accessToken2, oauthRequest);
            break;
    }
    respuestaAPI = oauthRequest.send();
    System.out.println(respuestaAPI.getBody());
    escribir(response, respuestaAPI.getBody());
}
} catch (Exception ex) {

```

```
response.sendRedirect("http://localhost:8080/ScribeJavaDemo");
}
}
```

Tenemos todo listo. Probemos:

- Facebook



NOTA: para obtener la foto de perfil del usuario en Facebook es necesario utilizar **me/picture** de la API Graph de Facebook.

- Twitter



Usuario Autenticado x

localhost:8080/ScribeJavaDemo/ScribeJavaServlet?oauth_token=PPHX2gAAAAAAvMYRAABVNqtl0&☆


Bienvenido

A continuación se visualizará la información del usuario

Información del usuario

- ID: 2576397392
- Nombre: Roy Taza Rojas
- Email: roytazarojas@gmail.com

Foto de Perfil



Respuesta JSON Completa

```
{
  "utc_offset": null,
  "friends_count": 55,
  "profile_image_url_https": "https://pbs.twimg.com/profile_images/700552259185152000/9s9-L-Zm_normal.jpg",
  "listed_count": 0,
  "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
  "default_profile_image": false,
  "created_at": "Jun 19 07:58:33 +0000",
  "2014",
  "is_translator": false,
  "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png",
  "protected": false,
  "screen_name": "L-Zm",
  "normal.jpg",
  "time_zone": null,
  "url": null,
  "contributors_enabled": false,
  "profile_background_tile": false,
  "entities": {
    "description": {
      "urls": []
    },
    "status": {
      "count": 1,
      "follow_request_sent": false,
      "followers_count": 1,
      "profile_use_background_image": true,
      "default_profile": true,
      "following": false,
      "Taza Rojas",
      "location": "",
      "profile_sidebar_fill_color": "DDEEF6",
      "notifications": false,
      "status": {
        "in_reply_to_status_id_str": null,
        "in_reply_to_status_id": null,
        "coordinates": null,
        "created_at": "Fri Feb 19 05:30:59 +0000",
        "2016",
        "truncated": false,
        "in_reply_to_user_id_str": "2437927682",
        "source": "Twitter Web"
      }
    }
  }
}
```

NOTA: para obtener el email en Twitter se debe enviar una petición al servicio para desarrolladores de Twitter.

- Google

localhost:8080/ScribeJavaDemo/ScribeJavaServlet?code=4/37N6Rmpzbi☆

Bienvenido

A continuación se visualizará la información del usuario

Información del usuario

- ID: 106037126302926835386
- Nombre: Roy Taza Rojas
- Email: roytazarojas@gmail.com

Foto de Perfil



Respuesta JSON Completa

```
{
  "gender": "male",
  "name": "Roy Taza Rojas",
  "link": "https://plus.google.com/106037126302926835386",
  "id": "106037126302926835386",
  "verified_email": "Roytas",
  "email": "roytazarojas@gmail.com",
  "picture": "https://lh4.googleusercontent.com/-1e_y4wjOU8/AAAAAAAAA"
}
```

- LinkedIn

Usuario Autenticado x

localhost:8080/ScribeJavaDemo/ScribeJavaServlet?code=AQQU_JRxHHIm5wAYksiF9agAih0yR664DI8s4

Bienvenido

A continuación se visualizará la información del usuario

Información del usuario

- ID: SeCdS8o02P
- Nombre: Roy AlejoTaza Rojas
- Email: roytazarojas@gmail.com

Foto de Perfil

Respuesta JSON Completa

```
{
  "firstName": "Roy Alejo",
  "lastName": "Taza Rojas",
  "emailAddress": "roytazarojas@gmail.com",
  "publicProfileUrl": "https://www.linkedin.com/in/roytaza",
  "pictureUrls": {
    "_total": 1,
    "values": [
      "https://media.licdn.com/mpr/mprx/0_PgENJatXKkLMuYPft0BcwVCkChkMjpg254Brur_k1tWyjrKd-4nthHHik_xHyj-PWB4B92NC3Ks5y0Drul9RRazGCFs5p0DTol9RnUaOGt8qz0sr_hlBllBvNp"]
    ],
    "pictureUrl": "https://media.licdn.com/mpr/mprx/0_02S6r5UkatM1dSQALGLN-7I6TKxeV4IrdLNP7I7hKxeL1KLXuXT05jNypaOwPXizFKm_6I4AjP2oNC90IuF",
    "id": "SeCdS8o02P",
    "headline": "Practicante Pre-Profesional de Gestión de Información",
    "siteStandardProfileRequest": {
      "url": "https://www.linkedin.com/profile/view?id=AAoAABTwOYcB7x5POI2ues9CNAaVVyddIEpqW14&authType=name&authToken=qIXy&trk=api*a4167604*s4234304*"
    }
  }
}
```

CONCLUSIONES

Utilizar ScribeJava es fácil, basta con 5 minutos de experiencia para poder utilizarlo con efectividad. Adicionalmente, esta librería permite conectarse con proveedores implementados por uno mismo al crear una clase que implemente DefaultApi10a o DefaultApi20 según sea el caso. Puede verse una guía en el siguiente [enlace](#).

Es importante revisar la respuesta del proveedor, ya que los parámetros de respuesta son diferentes para cada uno de ellos.