

Sociale media zijn niet meer weg te denken van het Internet. Ze worden steeds meer geïntegreerd in websites. Denk maar aan inloggen met je Facebook of Twitter account op een website waarvoor normaal geregistreerd moet worden, of delen van een leuk artikel via Like of (Re)Tweet knoppen. Maar hoe wordt een sociaal medium geïntegreerd in een website en waar moet op gelet worden?

Social media en Java

Basisprincipes om Java-webapplicatie te maken

Dit artikel beschrijft basisprincipes en alle stappen om een Java-webapplicatie te maken waarmee twitterberichten kunnen worden gepost, twittervrienden kunnen worden opgehaald en twitterzoekacties kunnen worden gedaan.

Om twitterberichten te kunnen posten hebben we een twitteraccount nodig. Dus de gebruiker van onze website moet eerst aanmelden met zijn login credentials. Maar we willen natuurlijk niet dat ze op onze website hun gebruikersnaam en wachtwoord invoeren omdat dit dan totaal onveilig zou zijn. Om dit probleem op te lossen heeft men OAuth bedacht.

OAuth

OAuth (Open Authorization) is een open standaard voor autoriseren. Dit maakt het voor de gebruikers mogelijk om hun privé data (bv. foto's, video's, contacts) welke opgeslagen zijn op een website voor een bepaalde tijd beschikbaar te stellen voor een andere website zonder hun inlog gegevens te geven (gebruikersnaam / wachtwoord). Momenteel wordt OAuth versie 1.0 gebruikt. Sinds kort is er ook een draft versie beschikbaar van OAuth versie 2.0 die voor de client applicaties bepaalde dingen makkelijker moet maken.

Het OAuth protocol definieert een aantal stappen om toegang te verlenen/verkrijgen op je social media informatie. Deze serie stappen is ook bekend als "OAuth dance". OAuth dance bestaat uit de volgende drie (OAuth 1.0) hoofdstappen:

- Request token vaststellen
- De gebruiker naar de autorisatieserver doorsturen zodat de gebruiker kan inloggen en rechten geven.
- Toegang token opvragen

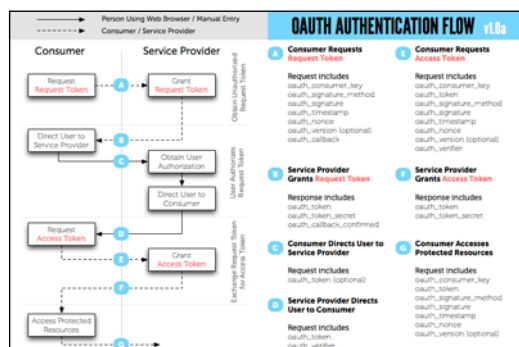
In figuur 1 wordt OAuth dance meer in details beschreven. Aan de linkerkant zien we welke acties worden uitgevoerd en door welke partijen. En aan de rechterkant zien we welke HTTP parameters worden meegestuurd met elke actie. Dus in het kort moeten de volgende stappen uitgevoerd worden om zo een OAuth dance met succes af te ronden. Deze stappen worden later in dit artikel met codevoorbeelden gedemonstreerd.

1. Registreren van de applicatie bij de service provider van de desbetreffende social media website. Omdat we in dit artikel Twitter als voorbeeld gaan gebruiken, moeten we onze applicatie registreren op de development website van Twitter [1]. Na het registreren krijgen we een "API Key" en een 'Secret Key' die bij onze applicatie hoort.
2. In de applicatie moet de gebruiker doorgestuurd worden naar de authenticatiepagina van de service provider. Hierin moet de API Key van de applicatie ook meegestuurd worden.
3. De gebruiker authenticceert zichzelf op de twitter aanmeldpagina.
4. De gebruiker geeft toestemming om de applicatie te gebruiken voor het delen van zijn informatie. Als dit al eerder gedaan is wordt deze stap overgeslagen.



Baris Dere

is Senior Java Consultant.
Hij is te bereiken op
baris.dere@gmail.com.



Figuur 1. OAuth dance stappen.

Een aantal grote bedrijven gebruikt OAuth al en daar komen er steeds meer bij.

- De gebruiker wordt door de service provider doorgevoerd naar onze applicatie. In deze stap wordt ook de request token en de verificatie token meegevoerd door de service provider.
- Verkrijg een access token met behulp van de request- en verificatie token.
- Start met communiceren met de social media website door middel van de access token en API key. Momenteel wordt OAuth gebruikt door grote bedrijven zoals Google, Twitter, Facebook, LinkedIn etc. en er komen steeds meer bij. Verder in dit artikel wordt beschreven hoe we met Java kunnen communiceren met Twitter door gebruik te maken van OAuth (dance).

Java libraries

Een aantal Java libraries maken het mogelijk OAuth te gebruiken om met een serviceprovider te communiceren. Dit zijn onder meer Scribe, Spring Social, Jersey en SignPost. In dit artikel worden Spring Social en Scribe gebruikt. Scribe wordt gebruikt om OAuth autorisatie af te handelen met de service provider en Spring Social wordt gebruikt om allerlei Twitter operaties uit te voeren zoals post, search en getFriends.

Spring Social

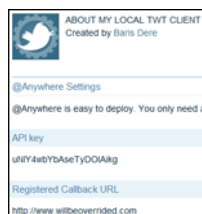
Spring Social is een subproject van SpringSource en maakt het mogelijk om met verschillende social media websites te communiceren. Momenteel worden Twitter, Facebook, LinkedIn en TripIt ondersteund. Voor elk van deze netwerken bestaat een Template implementatie op basis van de RestTemplate implementatie, waarmee OAuth providers RestFull benaderd kunnen worden. Hiernaast is er een aantal annotations om Facebook response tokens direct te kunnen benaderen.

Registreren applicatie in OAuth

Als eerste van de OAuth dance stappen moeten we onze applicatie registreren bij Twitter [1]. Vul alle informatie in en registreer je applicatie. Nadat je applicatie met succes is geregistreerd zul je een bevestigingspagina zien waarop allerlei URL's en key's te zien zijn. Al deze info wordt straks in de code gebruikt. Zie voor een voorbeeld figuur 2.

OAuthService

Om met de serviceprovider te kunnen communiceren hebben we de OAuthService implementatie van



Figuur 2. Twitter applicatie registratie bevestigingspagina.

Scribe framework nodig. In de configuratie moet alle informatie geset te worden. Zie onderaan een voorbeeld OAuthService implementatie voor Twitter.

```
static String apiKey = "uNlY4wbYbAseTyD0IAiKg";
static String apiSecret =
    "iy4vZ1M7qfb72LHCbnIoJpIncCLV0xWSQXbvvc9w04k";

public OAuthService getOAuthService() {
    OAuthConfig config = new OAuthConfig();
    config.setRequestTokenEndpoint("https://
api.twitter.com/oauth/request_token");
    config.setAccessTokenEndpoint("https://
api.twitter.com/oauth/access_token");
    config.setAccessTokenVerb(Verb.POST);
    config.setRequestTokenVerb(Verb.POST);
    config.setApiKey(apiKey);
    config.setApiSecret(apiSecret);
    config.setCallback("http://www.barisdere.com/
twitterclient/twitter/callback.html");

    return new OAuth10aServiceImpl(
        new HMACSha1SignatureService(),
        new TimestampServiceImpl(),
        new BaseStringExtractorImpl(),
        new HeaderExtractorImpl(),
        new TokenExtractorImpl(),
        new TokenExtractorImpl(),
        config);
}
```

Al deze informatie is terug te vinden op de dev applicatie site van Twitter [1]. In de callback attribute vullen we onze eigen URL in waarnaar de service provider de gebruiker na de autorisatie door zal sturen. Als we dit op onze localhost willen uitproberen moet de callback domein worden veranderd in localhost. Deze getOAuthService() methode zullen we in een andere methode gebruiken die eerst door de gebruiker aangeroepen zal worden om de applicatie binnen te treden (/twitter).

```
Static final String ACCESS_TOKEN = "user_access_token";
String authorizeUrl = "https://api.twitter.com/oauth/
authorize";

@RequestMapping(value="/twitter", method=RequestMethod.
GET )
public String updateStatus(WebRequest request) {
    Token accessToken = (Token)request.
getAttribute(ACCESS_TOKEN, WebRequest.SCOPE_SESSION);
    if (accessToken == null) {
        Token requestToken =
getOAuthService().getRequestToken();
        request.setAttribute
(REQUEST_TOKEN, requestToken, WebRequest.SCOPE_SESSION);
        return "redirect:" + authorizeUrl
+ "?oauth_token=" + requestToken.getToken();
    } else {
        return "statusUpdate";
    }
}
```

In deze code controleren we of de gebruiker al eerder autorisatie heeft gegeven voor onze applicatie. Als dat niet het geval is wordt de gebruiker doorgevoerd naar de autorisatiepagina van Twitter. Hierbij wordt ook de API key van de applicatie meegevoerd.

Authenticatie en autorisatie

De gebruiker komt op het autorisatiescherm (zie figuur 3) van Twitter. Na het aanmelden wordt op de knop Allow geklikt om rechten te geven aan onze applicatie om zijn account te kunnen gebruiken.



Figuur 3. Autorisatieschem.

Als dit gelukt is wordt de gebruiker door de service provider doorgestuurd naar de door ons opgegeven callback URL. In deze stap worden ook de request token en de verificatie token meegestuurd. Na het verifiëren van de tokens sturen we de gebruiker door naar ons scherm waarop berichten verstuurd kunnen worden. Deze access token slaan we op in de session om te voorkomen dat de gebruiker volgende keer opnieuw moet autoriseren.

```
@RequestMapping(value="/twitter/callback",
method=RequestMethod.GET, params="oauth_token")
public String authorizeCallback(@RequestParam(value =
"oauth_verifier", defaultValue = "verifier")
String verifier, WebRequest request) {
    Token requestToken = (Token)request.
getAttribute(REQUEST_TOKEN, WebRequest.SCOPE_SESSION);
    Token accessToken = oauth.
getAccessToken(requestToken, new Verifier(verifier));
    request.setAttribute(ACCESS_TOKEN, accessToken,
WebRequest.SCOPE_SESSION);
    return "redirect:../twitter";
}
```

Twitterberichten versturen

We gaan nu nog een andere methode toevoegen om twitterberichten te kunnen versturen. Deze methode wordt aangeroepen als we een POST verzoek sturen naar /twitter met een parameter genaamd 'message' welke het twitterbericht bevat. Als voorbeeld gebruiken we de tekst 'Hello World!!!' voor onze eerste bericht.

```
@RequestMapping(value="/twitter", method=RequestMethod.
POST )
public String updateStatus(@RequestParam("message")
String message, WebRequest request) {
    Token accessToken = (Token)request.
getAttribute(ACCESS_TOKEN, WebRequest.SCOPE_SESSION);

    TwitterTemplate template = new
TwitterTemplate(apiKey, apiSecret, accessToken.
getToken(), accessToken.getSecret());
    template.updateStatus(message);
    return "statusUpdateComplete";
}
```

Na het uitvoeren van deze method wordt ons twitterbericht gepost (zie figuur 4).

Followers ophalen

De TwitterTemplate class bevat verscheidene operations om met Twitter te kunnen communiceren. Een van deze methodes is getFriends(...). Hiermee kunnen we de opvolgers ophalen van de meegegeven screenName. Zo kunnen we alle opvolgers ophalen van de gebruiker:



Figuur 4. Twitterbericht gepost.

```
@RequestMapping(value="/friends", method=RequestMethod.
GET )
public ModelAndView myFriends(WebRequest request) {
    TwitterTemplate template = new
TwitterTemplate(myApiKey, myApiSecret,
myAccessToken,
myAccessTokenSecret);
    return new ModelAndView("friends", "list",
template.getFriends( template.getProfileId()));
}
```

Twitter zoekacties doen

Ook is het mogelijk om te gaan zoeken op Twitter. Hiervoor gebruiken we de methode search. We geven dan een string mee als zoekcriteria en krijgen een SearchResults object terug met daarin de gevonden tweets.

```
@RequestMapping(value="/search", method=RequestMethod.
GET )
public String search(WebRequest request) {
    TwitterTemplate template = new
TwitterTemplate(myApiKey, myApiSecret,
myAccessToken,
myAccessTokenSecret);

    String query = "#javamagazine";
    SearchResults searchResults = template.
search(query);
    return "searchResult";
}
```

Gebruikerssessies opslaan

We willen natuurlijk niet dat de gebruiker iedere keer als hij op ons website komt opnieuw autorisatie moet verlenen. Daarom kunnen we ervoor kiezen om de accesstoken en accesstokensecret op te slaan om volgende keer te kunnen hergebruiken. Er zit momenteel in Spring Social geen mogelijkheid om dit te kunnen doen. Totdat zo'n optie komt kunnen we ervoor kiezen om de tokens in een database op te slaan. We kunnen de tokens blijven gebruiken zolang die geldig zijn.

Conclusie

We hebben gekeken naar de mogelijkheden van Spring Social om met Twitter allerlei acties uit te voeren. We hebben gezien dat basisoperaties makkelijk gedaan kunnen worden met Spring Social. Nog lang niet alle operaties zijn ondersteund maar het is wel een stap in de goede richting. Twitter is een van de weinige social media websites die helemaal conform OAuth is. Omdat de OAuth steeds meer in populariteit toeneemt kunnen we verwachten dat straks meerdere websites overgaan op OAuth en dat de libraries meer volwassener worden. <<

Vanwege de groeiende populariteit gaan meer websites over op OAuth.

Referenties

[1] – Twitter Dev applicatie registreren (<http://dev.twitter.com/apps/new>)