

OpenSocialを使った 認証・認可

2012年04月20日

株式会社イーグル

菅井 康之

- SNS上の様々な情報にアクセスしたい
 - SNS自体が多く存在しているため、SNS毎にAPIを調べてアクセスするのは容易ではない
 - 複数のSNSにアクセスするための共通APIとして、Opensocialがあることを知る

- OpenSocial は、ウェブベースのソーシャルネットワークアプリケーションのための共通のアプリケーションプログラミングインタフェース (API) 群。Google によって開発され、2007年11月1日にリリースされた(Wikipedia)
- OAuth をベースとし、HTTP+Javascript等の既存のWeb技術で構成

Consumer Facing Containers

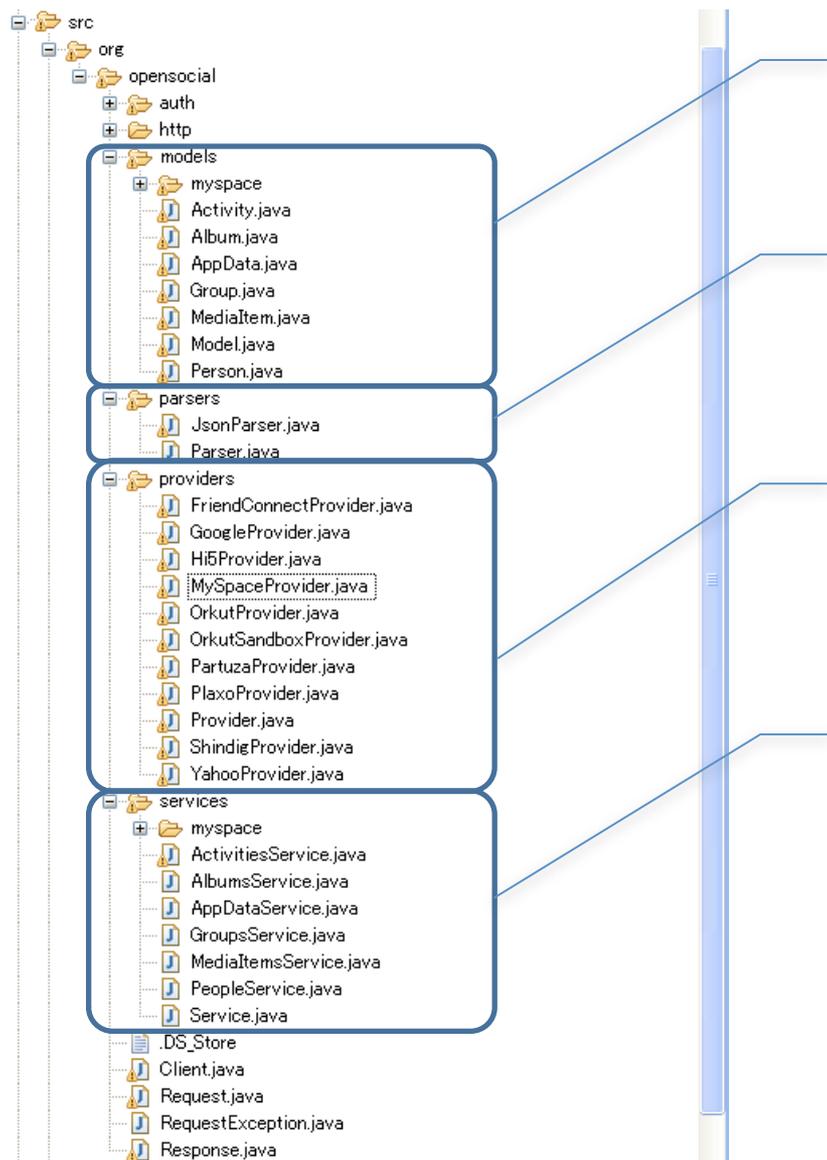
Container	OpenSocial Version	Developer Website
Avatars United 0.8.1	0.8.1	http://developer.avatarsunited.com/
Creyle	0.9	http://developer.creyle.jp/
Cyworld	0.9	http://devsquare.nate.com
eBay	0.9	http://developer.ebay.com/products/open-ebay-apps/
Friendster	0.7	http://www.friendster.com/developer
goo Home	0.8.1	http://developer.home.goo.ne.jp/
Google Friend Connect	0.9	http://code.google.com/apis/friendconnect
GROU.PS	0.9	http://grou.ps/groupsdev
hi5	0.8	http://developer.hi5.com/
Hypes	0.7	http://trac.hypes-api.nl/
iGoogle	0.9	http://code.google.com/apis/igoogle/
itimes	0.8	http://www.itimes.com/os_sandbox.php
iWiW	0.9	http://dev.iwiw.hu/
LinkedIn	0.8	http://developer.linkedin.com/index.jspa
Lonely Planet	0.8	http://lplabs.com/groups/
Mixi		http://developer.mixi.co.jp/
MySpace	1.0	http://developer.myspace.com/
Netlog	0.8	http://en.netlog.com/go/developer
Ning	0.8.1	http://developer.ning.com/
orkut	0.8.1	http://sandbox.orkut.com/
PayPal	0.8	https://www.x.com/community/ppx/paypal_apps
Renren		http://dev.renren.com/
Socialtext	0.8	http://www.socialtext.net/open/index.cgi?socialtext_widgets
Sonico	0.8	http://sandbox.sonico.com/app_dev_pres.php
Webjam	0.8.1	http://www.webjam.com/developers/opensocial
Webon	0.8	http://team.webonsites.com/
XING	0.9	https://www.xing.com/net/opensocialpartner
Yahoo!	0.9	http://developer.yahoo.com/yap
vkontakte.ru		http://vkontakte.ru/developers.php
VZ-Netzwerke	1.0	http://www.meinvz.net/Developer

<http://docs.opensocial.org/display/OSREF/List+of+OpenSocial+Containers>

何か遊べそう

- サーバサイドで処理させたいため、Javaでの実装を試みる
 - 1から作成するのは面倒なので、何か無いか調べた所、OpenSocialのJava-Client APIを発見
 - <http://code.google.com/p/opensocial-java-client/>
 - 2010年2月 Update以降、更新されていない
 - ソースが公開されているので何とか使えるだろうと使ってみることに

Java-Clientの構成



レスポンスの格納先

APIのレスポンスはJSON形式で返却
JSONを解析し、Modelを生成する

サービスプロバイダ毎に定義
EndPointやAccessTokenの取得先
URL等を定義

どの情報にアクセスするか

- Opensocialを使用する上で必要なもの
 - コンシューマキー (consumer_key)
 - コンシューマシークレット (consumer_secret)
 - サービスプロバイダのEndPoint URL
- 入手方法
 - consumer_key 、 consumer_secret
 - サービスプロバイダにコンシューマ登録することで発行される
 - コンシューマ登録には、ドメインを持つサーバが必要
 - 今回はGoogle App Engineを使用
 - サービスプロバイダのEndPoint URL
 - 各プロバイダのDeveloperサイトに記載

書いてみた

```
private HashMap<String, OAuth3LeggedScheme> schemeMap
    = new HashMap<String, OAuth3LeggedScheme>();
public void access(String requestToken) throws OAuthException, URISyntaxException, IOException, RequestException {
    //サービスプロバイダ毎に異なる定義-----
    final String CONSUMER_KEY      = "hoge hoge";
    final String CONSUMER_SECRET   = "piyo piyo";
    final String REDIRECT_URL      = "huga huga";
    final Provider provider        = new GoogleProvider();
    //-----
    if(requestToken == null) {
        OAuth3LeggedScheme scheme = new OAuth3LeggedScheme(
            provider, CONSUMER_KEY, CONSUMER_SECRET);

        //ユーザに認可要求を行う (getAuthorizationUrl内でrequestToken取得も行っている)
        String authUrl = scheme.getAuthorizationUrl(REDIRECT_URL);
        schemeMap.put(scheme.getRequestToken().token, scheme);
        redirectTo(authUrl);
        return;
    }

    OAuth3LeggedScheme scheme = schemeMap.get(requestToken);
    schemeMap.remove(requestToken);

    //アクセストークン要求(ユーザ認可済み)
    scheme.requestAccessToken(scheme.getRequestToken().token);

    //情報の取得要求 (とりあえずiGoogleで言うところの友だち)
    Client client = new Client(scheme.getProvider(), scheme);
    Response response = client.send(PeopleService.getFriends());
    //response出力
    for(Iterator<Model> entry = response.getEntries().iterator(); entry.hasNext(); ) {
        Person person = (Person)entry.next();
        System.out.println("-----");
        System.out.println(person.getDisplayName());
    }
}
```

Google アカウント

サイト は下記のサービスで使用するため Google アカウント へのアクセスをリクエストしています。

 Google Friends and Updates

 Google Friends and Updates

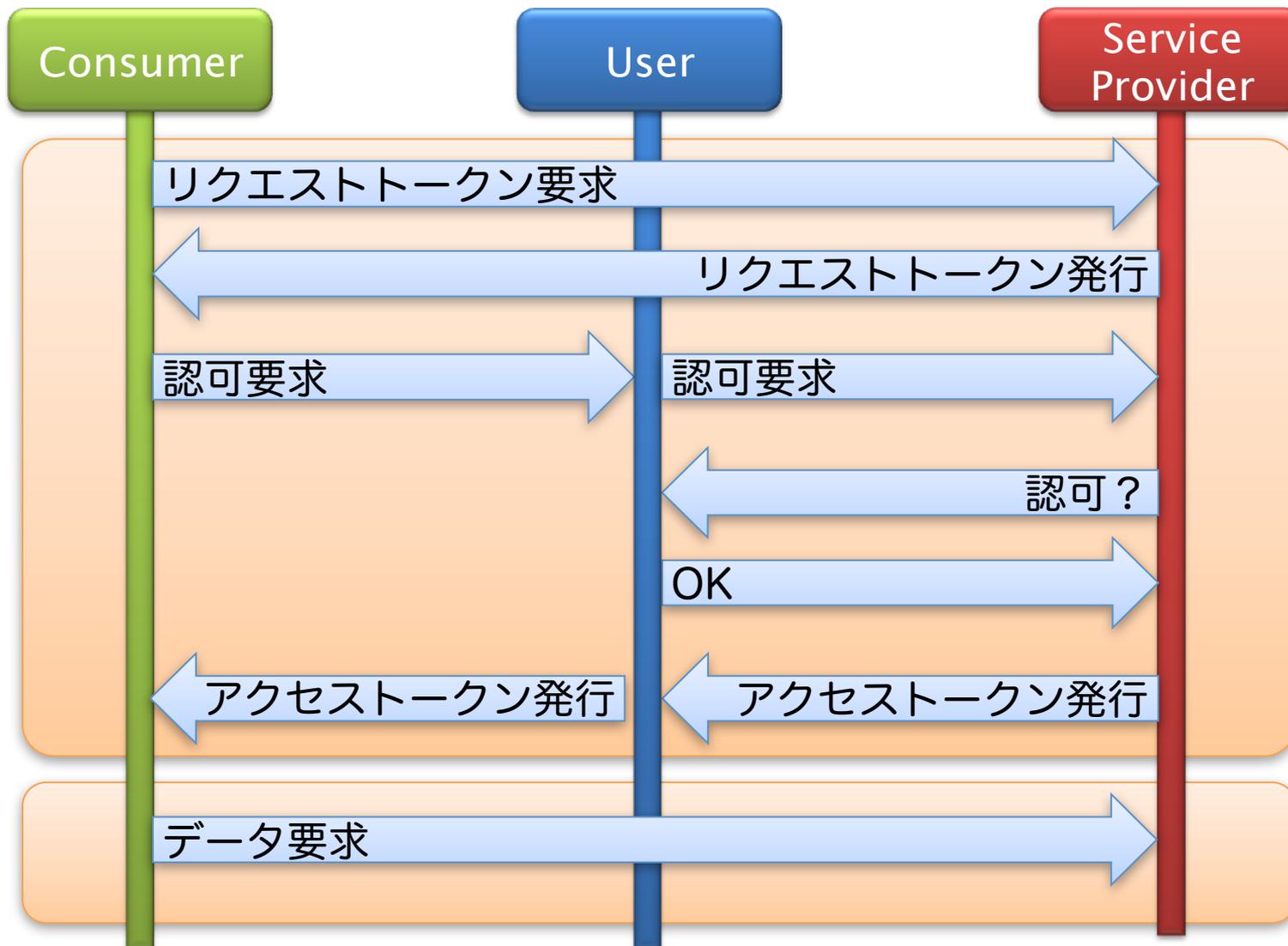
Google と は提携関係がありません。サイトを信用できる場合のみアクセスを許可してください。

アクセスを許可した後でも、いつでも [アカウント] でアクセス権を取り消すことができます。 は Google アカウントのパスワードや他の個人情報にアクセスできなくなります。 [詳細](#)

⚠ このウェブサイトは、認証リクエストを行うよう Google に登録されていますが、リクエストを安全に送信するように設定されていません。

でこのリクエストを開始しなかった場合にアクセスを許可すると、 の他のユーザーがあなたのデータにアクセスできる可能性があります。このリクエストを で直接開始したことが確実にない場合は、アクセスを拒否することをおすすめします。

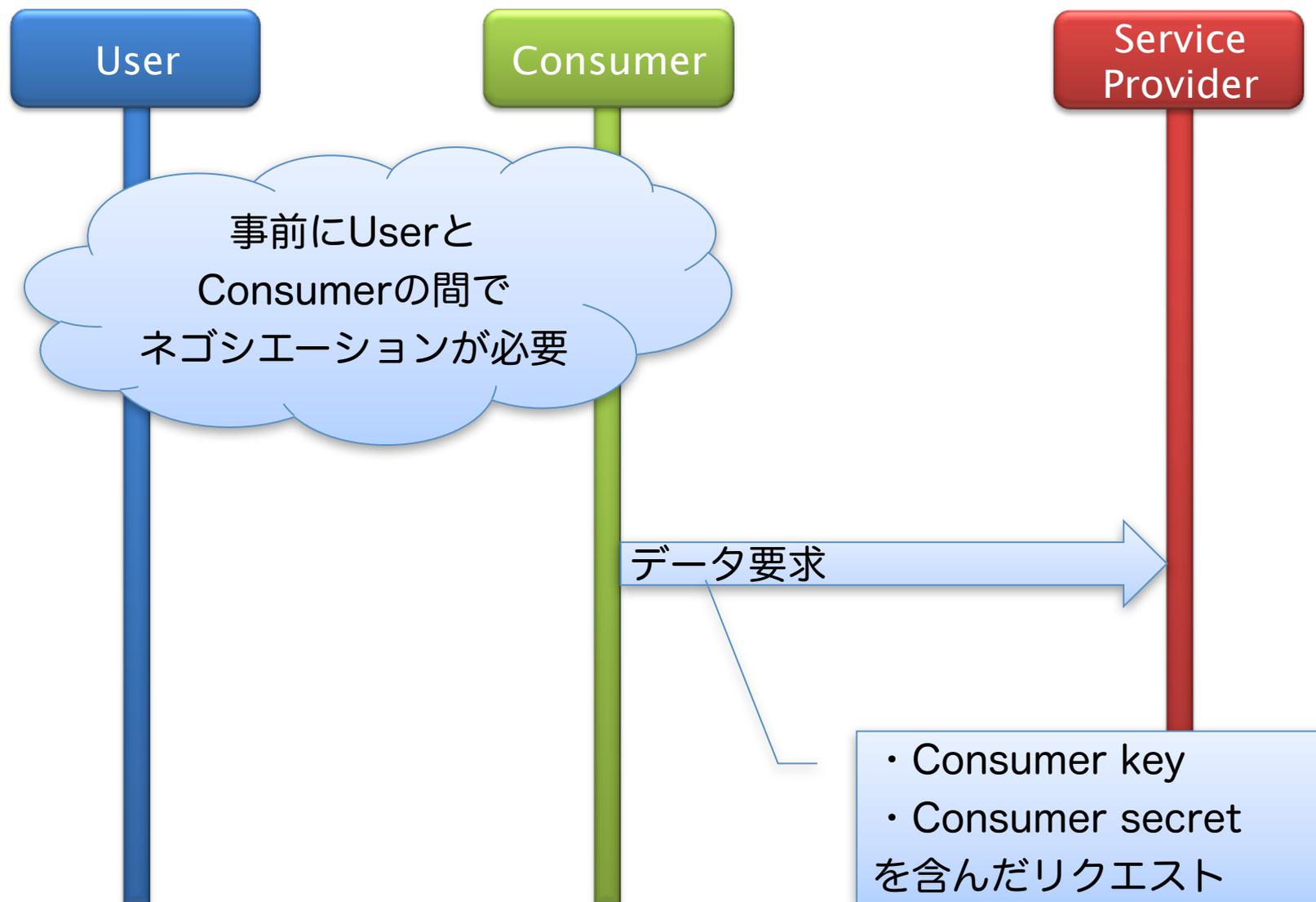
データアクセスの流れ



```
public void access(String requestToken) throws OAuthException, URISyntaxException, IOException, RequestException {  
    //サービスプロバイダ毎に異なる定義-----  
    final String CONSUMER_KEY      = "hogeHoge";  
    final String CONSUMER_SECRET   = "piyopiyo";  
    final String REQUESTER_ID      = "hugahuga";  
    final Provider provider        = new MixiProvider();  
    //-----  
    OAuth2LeggedScheme scheme = new OAuth2LeggedScheme(  
        CONSUMER_KEY, CONSUMER_SECRET, REQUESTER_ID);  
  
    //情報の取得要求  
    Client client = new Client(provider, scheme);  
    Response response = client.send(PeopleService.getFriends());  
    //response出力  
    for(Iterator<Model> entry = response.getEntries().iterator(); entry.hasNext(); ) {  
        Person person = (Person)entry.next();  
        System.out.println("-----");  
        System.out.println(person.getDisplayName());  
    }  
}
```

OpenSocialには認証方式が2つある事が判明

- 3-legged (iGoogle等)
- 2-legged (mixi等)



- ここまででOpensocialを利用するという点については、ある程度目処が付いた
- 使えそうなのはmixi位だが、mixiもAPIを移行し始めている(Opensocialでは無いものに。。。)
- 今SNSといえばFacebook
- Facebook APIとOpensocial APIとで何が違うのか
 - どちらもOAuthでの認証であるため、考え方は同じでは・・・

```
public void access(String code) throws OAuthException, URISyntaxException, IOException, RequestException {
    //-----
    final String APP_ID          = "hogeHoge";
    final String SECRET_KEY      = "piyopiyo";
    final String REDIRECT_URL    = "hugahuga";
    //-----

    if(code == null) {
        //ユーザ認可要求
        redirectTo(
            "http://www.facebook.com/dialog/oauth?"
            + "client_id=" + APP_ID
            + "&redirect_uri=" + REDIRECT_URL);

        return;
    }

    //アクセストークン要求(ユーザ認可済み)
    String accessToken = sendMessage(
        "https://graph.facebook.com/oauth/access_token?"
        + "client_id=" + APP_ID
        + "&redirect_uri=" + REDIRECT_URL
        + "&client_secret=" + SECRET_KEY
        + "&code=" + code
        + "&scope=offline_access");

    //情報の取得要求
    String response = sendMessage(
        "https://graph.facebook.com/me/friends?" + accessToken);
    //response出力
    System.out.println(response);
}
```

アプリのしくみについて詳しくはこちら。



Test

アプリへ移動

キャンセル

このアプリについて

このアプリによるFacebookタイムラインへの投稿の公開
範囲: [?]

※ カスタム ▼

このアプリが受け取る情報:

- 基本データ [?]

By proceeding, you will be taken to  · アプリを報告

- 認可の流れが把握出来た
 - OAuthによる認可
 - シグネチャーの生成
 - HMAC-SHA1 + BASE64
- サービスプロバイダになるには
 - OpenPNE 3
 - Apache Shindig
- APIで定義された以外にもっと他の情報を扱いたい
 - アクセストークンを払い出すI/Fとして活用
 - アクセストークン払い出し後、独自のAPIを呼び出す
 - YahooはYahooIDに関連する情報にアクセスする際のAPIを提供
 - アクセストークンを払い出すためにOpensocialのI/Fを利用している

- 今までの話はいくまで認可の話
 - 認証（身元確認）はしていない
- サービスプロバイダに依存するところが多い
 - 認証結果と属性情報を扱うための規定が無い
 - スコープの定義



OpenID Connect

- OAuth 2.0と親和性が高い
 - 認可フローが同じ
- OpenId 2.0の後継？
- Single Sign On の実現
- 認証レベルに応じたスコープ定義
- Etc...

クラウド研究会と一緒に活動しませんか？