

Google API 勉強会

SRA OSS, Inc. 日本支社

山本 博之

2010年12月13日

勉強会の流れ

- Google APIの概要
 - デモ
- cURLで実際に体験

Sylpheed Pro 2.0

- Sylpheed + Sylpheed Pro アドオン
 - ◆ メール全文検索
 - ◆ スケジューラー
 - ◆ デスクトップ検索
- DBにPostgreSQL 9.0を利用
- スケジューラーにGoogle カレンダー同期機能を搭載
 - ◆ cURLコマンドを利用して実装

Google APIの概要

- データの形式は Atom (XML) ベース
- データの通信は HTTP / HTTPS
- Google の各種サービスに対してAPIを提供：
 - ◆ Google カレンダー
 - ◆ Google ドキュメント
 - ◆ Google マップ
 - ◆ Picasa Web アルバム
 - ◆ YouTube
 - ◆ ...

Google Data API

- どのサービスのAPIもGoogle Data APIに基づく
- データの問い合わせ、追加、更新などをXMLで表現し、HTTPで送信する
- XMLで送られてきたレスポンスを受け取り、解析する
- 基本的にはこれだけ
- Java, JavaScript, .NET, PHP, Python, Objective-C についてはGoogleがライブラリを提供

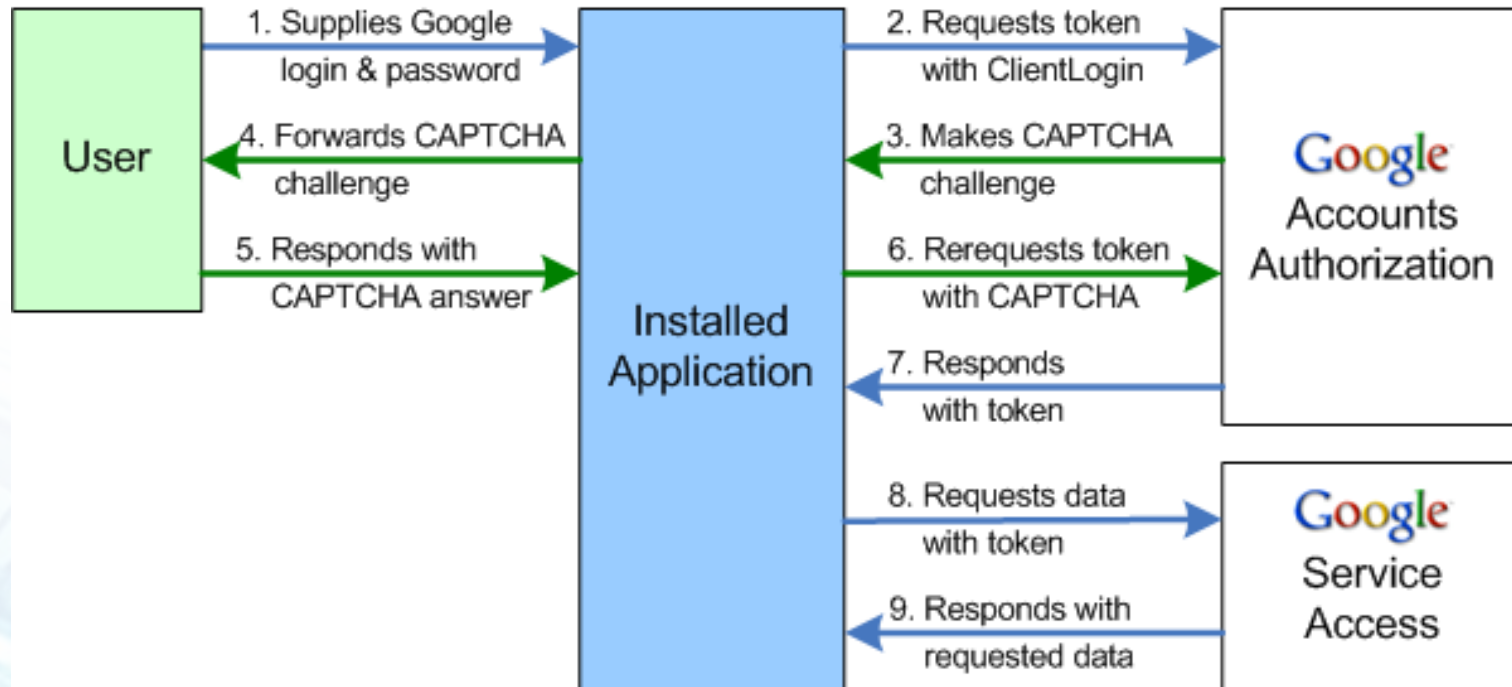
Google APIの認証方法

- **ClientLogin** : クライアントアプリケーションで使用
- **AuthSub** : Webアプリケーションで使用
- **OAuth** : クライアント、Webアプリから使用可能
 - ◆ オープンな規格 (Twitter等でも利用)

ClientLoginによる認証

- ユーザIDとパスワードを送信
- (サービスによっては CAPTCHA 要求)
- トークン文字列が返ってくる (Auth=...)
- 以降、このトークンを使ってデータを要求
- ClientLoginのメリット:
 - ◆ 毎回認証する必要がないので性能が向上する
 - ◆ セキュリティのリスクが軽減される

ClientLoginの認証プロセス



cURLでClientLogin認証

```
curl https://www.google.com/accounts/ClientLogin \  
-d Email=hogehoge@gmail.com \  
-d Passwd=password \  
-d source=SRAOSS-Sylpheed-2.0 \  
-d service=cl ← Googleカレンダーを指定
```

結果:

SID=DQAAAHYBADCV2pSv7nflac...

LSID=EUBBBIaBADCI-kNxvRVmc...

Auth=EUBBIacAAADK-kNxvRVmc...

← これを使う

認証後のトークンの使い方

```
curl --header \  
  "Authorization: GoogleLogin auth=EUBBIac..." \  
  http://www.google.com/calendar/feeds/default/private/full
```

Google カレンダーのデータを取得

- <http://www.google.com/calendar/feeds/default/private/basic> (デフォルトのカレンダー)
- <http://www.google.com/calendar/feeds/<カレンダーID>/private/basic> (カレンダーIDを指定)
- <http://www.google.com/calendar/feeds/<カレンダーID>/private-<magic>/basic> (認証なしで取得可能)
- <http://www.google.com/calendar/ical/<カレンダーID>/private-<magic>/basic.ics> (iCalendar形式)

マイ カレンダー → 対象 ▼ → カレンダー 設定 → 限定公開URL

デフォルトのカレンダーを取得する

```
curl --header \  
  "Authorization: GoogleLogin auth=<トークン>" \  
  http://www.google.com/calendar/feeds/default/private/full
```

→「Moved Temporarily」が返ってくるので、URLを転送先に変えて再度実行

```
curl --header \  
  "Authorization: GoogleLogin auth=<トークン>" \  
  http://www.google.com/calendar/feeds/default/private/full?  
  gsessionid=CpWKjb-7x6NeABC2qjzNA
```

取得したカレンダーのデータ（抜粋）

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns='http://www.w3.org/2005/Atom' xmlns:openSearch='http://a9.com/-/spec/opensearchrss/1.0/'
xmlns:gCal='http://schemas.google.com/gCal/2005'>
<id>http://www.google.com/calendar/feeds/default/private/basic</id>
<updated>2010-11-30T13:24:58.000Z</updated>
<title type='text'>Hiroyuki Yamamoto</title>
<subtitle type='text'>Hiroyuki Yamamoto</subtitle>
<author><name>Hiroyuki Yamamoto</name><email>hiroyuki.yamamoto@gmail.com</email></author>
<generator version='1.0' uri='http://www.google.com/calendar'>Google Calendar</generator>
<entry>
<id>http://www.google.com/calendar/feeds/default/private/basic/r5vgs23iigbum19em3j00icaajo</id>
<published>2010-11-30T13:24:57.000Z</published>
<updated>2010-11-30T13:24:57.000Z</updated>
<category scheme='http://schemas.google.com/g/2005#kind' term='http://schemas.google.com/g/2005#event'/>
<title type='html'>予定タイトル</title>
<summary type='html'>期間: 2010/12/04 (&#22303;) 11:00 ~ 12:00</summary>
<content type='html'>期間: 2010/12/04 (土) 11:00 ~ 12:00</content>
<author><name>Hiroyuki Yamamoto</name><email>hiroyuki.yamamoto@gmail.com</email></author>
</entry>
</feed>
```

カレンダーデータの構造

<feed>

フィードに関する情報...

<entry>

<id>エントリのID</id>

エントリに関する情報...

</entry>

<entry>

...

</entry>

</feed>

ID (UID) について

`<id>http://www.google.com/calendar/feeds/default/private/basic/r5vgs23iigbum19em3j00icajo</id>`

- 特定の予定を識別するためのID
- 予定の取得、更新、削除のときに使用する

カレンダーに予定を追加

1. 以下の内容のXMLファイルを**UTF-8**で作成する

```
<entry xmlns='http://www.w3.org/2005/Atom'  
  xmlns:gd='http://schemas.google.com/g/2005'  
  <category scheme='http://schemas.google.com/g/2005#kind'  
    term='http://schemas.google.com/g/2005#event' />  
  <title type='text'>予定の名前</title>  
  <content type='text'>予定の詳細</content>  
  <gd:transparency value='http://schemas.google.com/g/2005#event.opaque'>  
  </gd:transparency>  
  <gd:eventStatus value='http://schemas.google.com/g/2005#event.confirmed'>  
  </gd:eventStatus>  
  <gd:where valueString='予定の場所'></gd:where>  
  <gd:when startTime='2010-12-14T09:00:00.000Z'  
    endTime='2010-12-14T11:00:00.000Z'></gd:when>  
</entry>
```

入力が面倒な方は以下から取ってください: <http://sylpheed.sraoss.jp/tmp/study-entry.xml>

各要素の意味

<entry>

<**title** type='text'>予定の名前</title>

<**content** type='text'>予定の詳細</content>

<**gd:transparency** 予定ありとして見えるかどうか />

<**gd:visibility** 公開するかどうか(public/private) />

<**gd:where** valueString='予定の場所' />

<**gd:who** email='...' valueString='参加者' />

<**gd:when** startTime='開始時刻' endTime='終了時刻' />

(<**gd:recurrence**>繰り返し予定</gd:recurrence>)

</entry>

カレンダーに予定を追加

2. POSTメソッドでHTTPリクエストを投げる

```
curl --request POST \  
  --header "Expect:" \  
  --header "Content-Type: application/atom+xml" \  
  --header \  
  "Authorization: GoogleLogin auth=<トークン>" \  
  --data-binary "@path/to/study-entry.xml" \  
  http://www.google.com/calendar/feeds/default/private/full
```

その他の操作

- 予定の更新

- ◆ IDを指定して予定を取得する
 - ◆ <http://www.google.com/calendar/feeds/calendarID/events/entryID>
- ◆ 取得したXMLの内容を編集する
- ◆ 予定のURLに**PUT**リクエストでXMLを送信する

- 予定の削除

- ◆ 上記と同じURLに**DELETE**リクエストを送信する

その他

- JSON-C 形式
 - ◆ Atom XML 形式の代わりに JSON-C という形式が使える
 - ◆ XMLに比べて人間が読みやすい
 - ◆ URLに alt=jsonc クエリパラメータを追加する

```
{
  "data": {
    "title": "Tennis with Beth",
    "details": "Meet for a quick lesson.",
    "transparency": "opaque",
    "status": "confirmed",
    "location": "Rolling Lawn Courts",
    "when": [
      {
        "start": "2010-04-17T15:00:00.000Z",
        "end": "2010-04-17T17:00:00.000Z"
      }
    ]
  }
}
```

その他

- バッチ処理

- ◆ 一回のリクエストで複数の処理をまとめて高速に行える
 - ◆ クエリ、追加、更新、削除
- ◆ XMLのみで利用可能 (JSON-C では使えない)

```
<feed>
  <entry>
    <batch:id>Insert01</batch:id>
    <batch:operation type='insert' />
    ...
  </entry>
  <entry>
    <batch:id>Update01</batch:id>
    <batch operation type='update' />
    ...
  </entry>
</feed>
```

参考URL

- **Google カレンダーのAPIとツール**
 - ◆ <http://code.google.com/intl/ja/apis/calendar/>
- **Data API Developer's Guide**
 - ◆ http://code.google.com/intl/ja/apis/calendar/data/2.0/developers_guide.html
 - ◆ Protocol Guide, Atom Reference
- **Google Data API**
 - ◆ <http://code.google.com/intl/ja/apis/gdata/index.html>
- **Using cURL to interact with Google Data services**
 - ◆ http://code.google.com/intl/ja/apis/gdata/articles/using_cURL.html

参考URL

- **Sylpheed**
 - ◆ <http://sylpheed.sraoss.jp/>
- **Sylpheed Pro**
 - ◆ <http://www.sraoss.co.jp/sylpheed-pro/>
- **Twitter (個人)**
 - ◆ http://twitter.com/hiro_sylpheed
- **Sylpheed Pro 公式 Twitter**
 - ◆ http://twitter.com/sylpheed_pro

本資料のURL

http://sylpheed.sraoss.jp/tmp/Google_API_study.pdf