

Accéder aux API GOOGLE en PHP

Pour les besoins de son application web (site, blog, portail...), il est parfois nécessaire d'accéder aux ressources stockées sur un service Google tel que Youtube, Picassa, Blogger, Documents, ... Pour répondre à ce besoin, Google offre un certain nombre d'API afin d'exploiter ses services et d'intégrer les résultats à d'autres applications.

Cet article explique :

- Quels sont les services Google qui sont accessibles par les API en PHP supportant le protocole de données Google Data Protocol, et quelle est leur logique commune de programmation.

Ce qu'il faut savoir :

- Le langage PHP.
- Savoir utiliser Zend Framework.

Pour les besoins de son application web (site, blog, portail...), il est parfois nécessaire d'accéder aux ressources stockées sur un service Google tel que Youtube, Picassa, Blogger, Documents, ... Pour répondre à ce besoin, Google offre un certain nombre d'APIs afin d'exploiter ses services et d'intégrer les résultats à d'autres applications.

Certaines de ces APIs s'appuient sur le protocole de données *Google Data Protocol* (<http://code.google.com/intl/fr/apis/gdata>). Celui-ci est largement inspiré de l'architecture REST, conçue pour manipuler les données sur internet. Il supporte actuellement *AtomPub* et JSON pour représenter et véhiculer les informations. Comme tout protocole de données, *Google Data Protocol* s'interface avec divers langages du Web, tels que Java, .NET, Perl, Python ou PHP. C'est à travers ce dernier langage que nous allons découvrir quelques fonctionnalités.

Présentation des Services et API

Il est nécessaire tout d'abord, de s'inscrire sur le service souhaité avec son adresse e-mail. Sachant par ailleurs que l'adresse email est le point d'entrée incontournable pour piloter et gérer les services Google, il est important de bien utiliser son ou ses adresses emails pour regrouper ou dissocier les services. En effet, il sera difficile ultérieurement de migrer ou de fusionner les services avec des adresses emails différentes.

La liste complète des services Google qu'il est possible de piloter en PHP avec le protocole *Google Data*

Protocol se trouve à l'url <http://code.google.com/intl/fr/apis/gdata/docs/directory.html>). En voici quelques uns.

Google Blogger

Google Blogger (<http://www.blogger.com>) est une plateforme complète de publication de blogs. L'*API Blogger Data* (<http://code.blogger.com>) permet d'interroger, de visualiser et mettre à jour le contenu du blog via une application cliente autre.

Youtube

Accessible à l'adresse <http://youtube.com>, Youtube est une plateforme de partage de vidéos et offre en outre toute une panoplie de fonctionnalités pour commenter et annoter les vidéos. L'*API YouTube Data* (<http://code.google.com/intl/fr/apis/youtube>) permet de créer,



Figure 1. Portail de l'api Blogger Data

modifier, visualiser et gérer les vidéos via sa propre application.

Picasa

Picasa (<http://picasaweb.google.com>) est une plateforme de partage d'albums d'images et de photos. L'API *Picasa Web Albums Data* (<http://code.google.com/intl/fr/apis/picasaweb>) autorise la gestion des albums, des images et des photos.

Google Books Search

Google Books (<http://books.google.com>) permet de rechercher spécifiquement dans le contenu des livres mis à disposition. La recherche s'effectue sur tout le contenu de l'ouvrage lorsqu'il s'agit d'un livre dans le domaine public, mis à disposition gratuitement ou dans le cadre d'une licence *creative common*... Pour les ouvrages commercialisés, le contenu partiel ou choisi en partenariat avec l'éditeur sera accessible. Google Books autorise également la création et le partage de bibliothèques de documents, ainsi que de rechercher la disponibilité d'un ouvrage à un endroit localisé (bibliothèque physique ou librairie par exemple).

Toutes ces actions sont réalisables à partir d'une application avec l'API *Google Books Search* (<http://code.google.com/intl/fr/apis/books/>).

Google Calendar

Google Calendar (<http://www.google.com/calendar>) permet de mettre en ligne et de partager son agenda, où un agenda événementiel. L'API *Calendar Data API* (<http://code.google.com/intl/fr/apis/calendar>) autorise toute la gestion et la visualisation de votre agenda dans votre application.

Google Documents

Google Documents (<http://docs.google.com>) autorise la mise en ligne et le partage de documents. Il est possible de partager ces documents avec tout le monde ou de spécifier les emails autorisés. Il est également possible d'autoriser spécifiquement et individuellement soit la visualisation, soit la modification du document. Par ailleurs, les documents peuvent être partagés soit avec leur format natif, soit en tant que pièce attachée.

L'API *Google Documents List* (<http://code.google.com/intl/fr/apis/documents>) s'interface avec ce service et permet de créer, partager, uploader et modifier les documents. Lorsqu'il s'agit plus spécifiquement de feuilles de calcul, l'API *Google Spreadsheets*



Figure 2. Portail de l'api YouTube Data



Figure 3. Portail de l'api Picasa Web Album Data



Figure 4. Portail de l'api Google Book Search



Figure 5. Portail de l'api Google Calendar

Listing 1.

```
$titre = 'titre de mon billet' ;
$contentu = 'contenu de mon billet' ;
$url = 'http://www.blogger.com/feeds/' . $blogID . '/'
posts/default';

$mon-billet = $mon-blog->newEntry();
$mon-billet->title = $mon-blog->newTitle($titre);
$mon-billet->content = $mon-blog->
newContent($contentu);
$mon-billet->content->setType('text');
$mon-billet-dans-blog = $mon-blog->insertEntry($mon-
billet, $url);
```

Et pour Picasa, ce sera de la manière suivante :

```
$titre = ' titre du nouvel album' ;
$contentu = ' description du nouvel album' ;

$google-client = new Zend_Gdata_Photos($client,
"Google-DevelopersGuide-1.0");
$album = new Zend_Gdata_Photos_AlbumEntry();
$album->setTitle($google-client->newTitle($titre));
$album->setSummary($google-client->
newSummary($contentu));
$nouvel-album = $ google-client->insertAlbumEntry($
album);
```

(<http://code.google.com/intl/fr/apis/spreadsheets>) peut être très intéressante.

Principe de fonctionnement général

Comme nous l'avons vu ci-dessus, les services offrent tous des fonctionnalités, d'authentification, de création de recherche et de visualisation. Ces fonctionnalités sont spécifiques ensuite à un contenu mais ont le même schéma de fonctionnement. Il est donc logique que les apis qui s'appuient sur ce protocole *Google Dat Protocol* s'interfacent naturellement avec la même librairie.

En PHP, ce socle commun est proposé par Google par l'intermédiaire de l'environnement Zend Framework. Cet environnement fournira, après installation, les fonctions transversales d'authentification, de création, de recherche et de visualisation.

Installer la librairie

Pré-requis indispensable et commun à toutes les apis, il faut télécharger la librairie à l'adresse <http://framework.zend.com/download/gdata>. Dans le code PHP, il faudra donc mettre dans l'entête pour charger la librairie :

```
require_once 'Zend/Loader.php';
```

Puis, il faut choisir et charger les classes nécessaires au déroulement de son application. Ainsi, pour tous les services, il faut charger :

```
Zend_Loader::loadClass('Zend_Gdata');
Zend_Loader::loadClass('Zend_Gdata_Query');
Zend_Loader::loadClass('Zend_Gdata_ClientLogin');
```

Si le service nécessite une authentification supplémentaire, il faut charger la classe `Zend_Gdata_AuthSub` par la même commande :

```
Zend_Loader::loadClass('Zend_Gdata_AuthSub');
```

Et pour chaque service particulier, il faut charger la classe spécifique. Ainsi, par exemple, pour le service Picasa, il faut charger la classe `Zend_Gdata_Photos` par l'intermédiaire de :

```
Zend_Loader::loadClass('Zend_Gdata_Photos');
```

S'authentifier

Il faut utiliser le compte google utilisé pour créer et gérer le service souhaité. Ainsi, par exemple, pour le service *GoogleBlogger*, le code sera le suivant :

```
$nom = 'nom-compte-google@fournisseur.com';
$pwd = 'mot-de-passe';
$service = 'blogger';
```

```
$client = Zend_Gdata_ClientLogin::getHttpClient($nom,
$pwd, $service, null,
Zend_Gdata_ClientLogin::DEFAULT_SOURCE,
null, null,
Zend_Gdata_ClientLogin::CLIENTLOGIN_URI,
'GOOGLE');
$mon-blog = new Zend_Gdata($client);
```

Création de contenu

Le formatage des informations dépend du service visé. Ainsi, par exemple, pour le service *GoogleBlogger*, le code pour intégrer un billet est présenté dans le Listing 1.

Interrogation

Pour interroger le service, la classe `Zend_Gdata_Query` peut être utilisée simplement ou prendre les paramètres particuliers de ce service. Ainsi, par exemple, pour

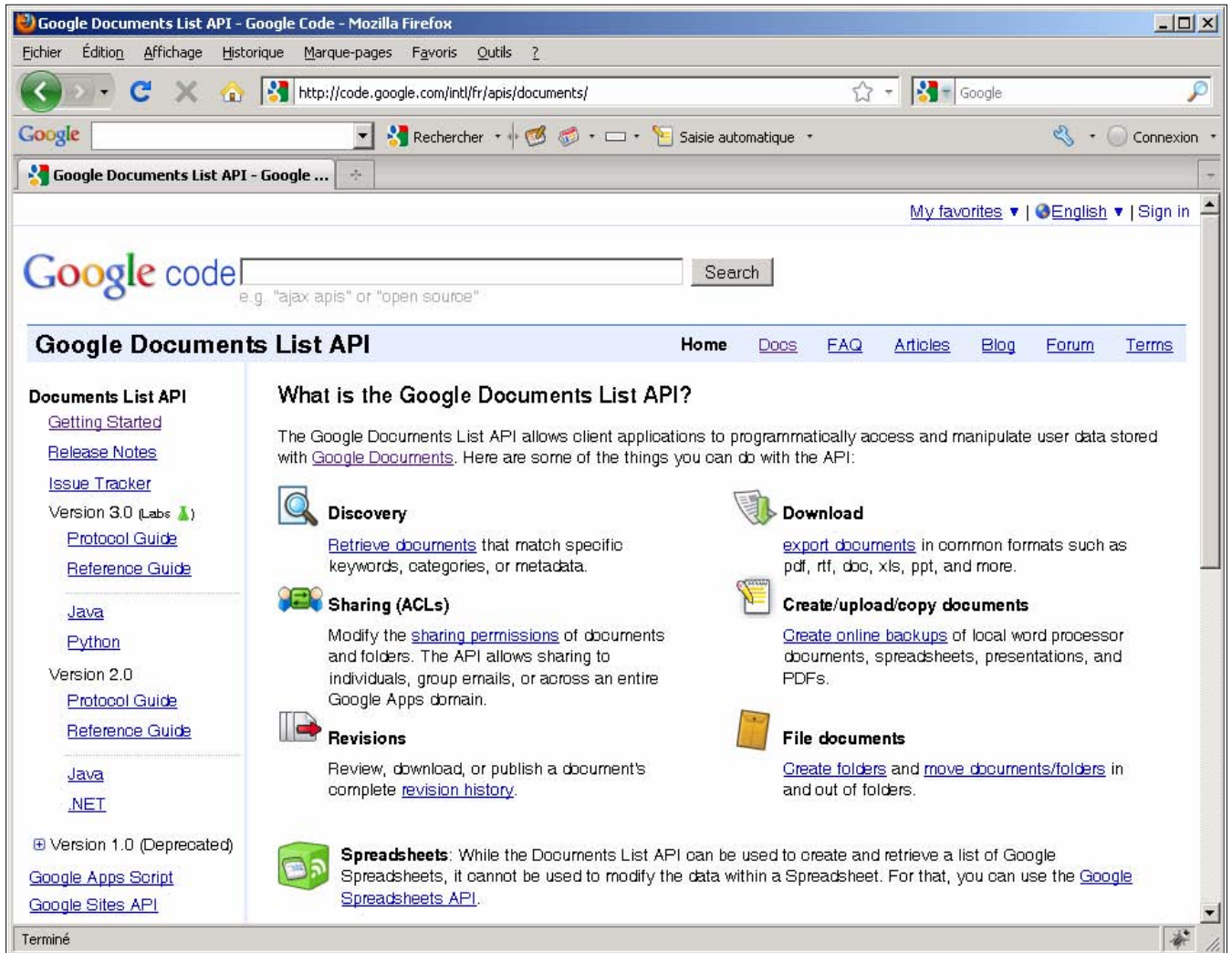


Figure 6. Portail des api Google Documents

recupérer l'ensemble des billets de son compte sur *Google Blogger*, la requête aura la forme suivante :

```
$quest=new Zend_Gdata_Query('http://www.blogger.com/feeds/default/blogs');
$resultats = $gdClient->getFeed($quest);
```

Pour Picassa, ce sera :

```
$quest = $google-client->newAlbumQuery();
$quest->setUser("default");
$quest->setAlbumName("album à rechercher");
$resultats = $ google-client->getAlbumFeed($quest);
```

Affichage de résultats

Les résultats de même nature peuvent ainsi être affichés au sein d'un même type de boucle. Ainsi, par exemple, pour les deux services interrogés ci-dessus, la boucle d'affichage sera la suivante :

```
foreach($resultats->entries as $document) {
    print $document->title . "\n";
}
```

Conclusion

Les services Google sont complètement pilotables à l'aide des APIs, quel que soit le langage utilisé. En outre, si l'API n'existe pas dans le langage souhaité, il est toujours possible de la développer et de la proposer ensuite à Google. C'est un concept assez intéressant dans la mesure où il permet de s'affranchir des contraintes de programmation multiple et de tout piloter à partir d'un seul langage.

ISABELLE LUPI

Spécialisée dans les domaines de la gestion documentaire, du traitement du langage naturel et du référencement. Travaille depuis plus de 10 ans dans ce dernier domaine et a créé, début 2007, b-find, sa propre société spécialisée dans le domaine du référencement.