

Recherche infos livre

Parsing réponse JSON

Cédric Dumoulin

2016

Plan

- Problème
- Recherche des infos sur les livres
 - Google book API
 - Amazon ?
 - Envoie de requêtes HTTP
 - Attente de la réponse
- Analyse (Parsing) de la réponse JSON
- Scan de Code barres
 - Principe
 - API Google
 - API XZing
- Exemple
 - Tuto complet
 - Comment remplacer apache httpREquest

Problème

Problème

- Comment rechercher des informations sur un livre, à partir de :
 - Son ISBN ?
 - Une partie du titre ?
 - Un nom d'auteur ?
 - ...

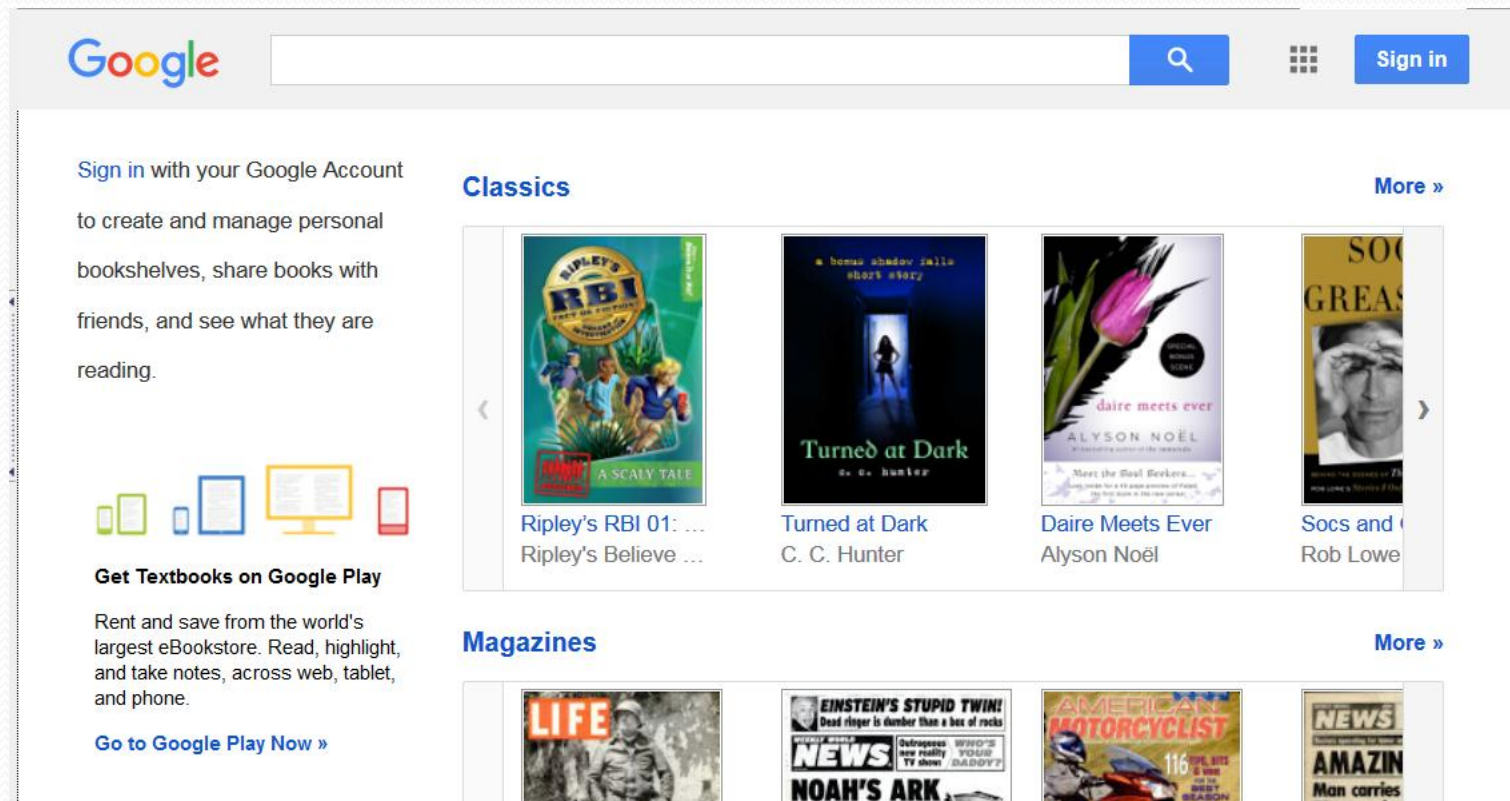
Solution

- Utiliser des sites spécialisés comme :
 - Google Book
 - Amazon
- Demander des infos à l'aide d'une requête HTTP
- Attendre de la réponse
- Extraire les informations de la réponse
 - ➔ format JSON

Recherche sur API Google book

Google Book

- <https://books.google.com/>
- Permet d'obtenir des renseignements sur les livres



The screenshot shows the Google Books homepage. At the top is the Google logo, a search bar, and a 'Sign in' button. Below the search bar, there's a section for signing in with a Google Account to manage bookshelves. To the right, under 'Classics', there are four book covers: 'Ripley's RBI 01: A SCALY TALE' by Ripley's Believe..., 'Turned at Dark' by C. C. Hunter, 'Daire Meets Ever' by Alyson Noël, and 'Socs and Greasers' by Rob Lowe. Below this, under 'Magazines', there are four magazine covers: 'LIFE', 'EINSTEIN'S STUPID TWIN!', 'AMERICAN MOTORCYCLIST', and 'NEWS'. Navigation arrows are visible on the left and right sides of the book and magazine rows.

Google

Sign in with your Google Account to create and manage personal bookshelves, share books with friends, and see what they are reading.

Get Textbooks on Google Play

Rent and save from the world's largest eBookstore. Read, highlight, and take notes, across web, tablet, and phone.

Go to Google Play Now »

Classics

More »

Ripley's RBI 01: A SCALY TALE
Ripley's Believe ...

Turned at Dark
C. C. Hunter

Daire Meets Ever
Alyson Noël

Socs and Greasers
Rob Lowe

Magazines

More »

LIFE

EINSTEIN'S STUPID TWIN!
Dead ringer is dumber than a box of rocks

AMERICAN MOTORCYCLIST

NEWS
AMAZING
Man carries

API

- Google Book propose une API pour faire des requêtes
- <https://developers.google.com/books/>
- Nécessite un compte google
 - Et une authentification
 - Forte si accès aux données privées
 - Par clé si accès aux données public ←
 - Ne stockez pas vos clés dans votre code source !!
- 2016
 - L'API peut être utilisée sans compte ni clé !

Terminologie

- **Volume** – Représente un **livre** ou un **magazine**. C'est l'unité de base
- **Bookshelf** (Bibliothèque, étagère) – Un **ensemble de volumes**.
- Un Bookshelf par défaut – contient tous les volumes connu de Google
- Des Bookshelves 'privés' – permet de créer des ensembles de volumes

Faire une recherche

- `https://www.googleapis.com/books/v1/volumes?q=search+terms`
- `q=` Recherche des volumes contenant les mots spécifiés
- `terms` – permet de spécifier des zones de recherche
 - `intitle:value+inauthor:value+isbn:value`
- GET
`https://www.googleapis.com/books/v1/volumes?q=flowers+inauthor:keys&key=yourAPIKey`

Terms

- **intitle:**
 - Returns results where the text following this keyword is found in the title.
- **inauthor:**
 - Returns results where the text following this keyword is found in the author.
- **inpublisher:**
 - Returns results where the text following this keyword is found in the publisher.
- **subject:**
 - Returns results where the text following this keyword is listed in the category list of the volume.
- **isbn:**
 - Returns results where the text following this keyword is the ISBN number.
- **lccn:**
 - Returns results where the text following this keyword is the Library of Congress Control Number.
- **oclc:**
 - Returns results where the text following this keyword is the Online Computer Library Center number.

Format du résultat

- Si la requête réussie :
 - statut 200 OK
- Réponse de type JSON

```
{
  "kind": "books#volumes",
  "items": [
    {
      "kind": "books#volume",
      "id": "_ojXNuzgHRcC",
      "etag": "OTD2tB19qn4",
      "selfLink":
        "https://www.googleapis.com/books/v1/volumes/_ojXNuzgHRcC",
      "volumeInfo": {
        "title": "Flowers",
        "authors": [
          "Vijaya Khisty Bodach"
        ],
        ...
      },
    },
    {
      "kind": "books#volume",
      "id": "RJxWlQOvoZUC",
      "etag": "NsxMT6kCCVs",
      "selfLink":
        "https://www.googleapis.com/books/v1/volumes/RJxWlQOvoZUC",
      "volumeInfo": {
        "title": "Flowers",
        "authors": [
          "Gail Saunders-Smith"
        ],
        ...
      },
    },
    ...
  ],
  "totalItems": 3
}
```

Autres fonctions de l'API

- Créer des étagères
- Spécifier le format de la réponse
- ...
- Voir :
 - <https://developers.google.com/books/docs/v1/using>

Recherche sur API Amazon

API Amazon

- Similaire a Google Book
- Plus d'infos :
 - sur le web
- Avantage
 - Réponse plus fournit que Google Book
 - Plus de livre que Google Book

Faire une requête http
A partir d'une Activity

Problème

- L'utilisation des API Books se fait par envoi d'une requête HTTP, et attente de la réponse
- Comment faire une requête HTTP à partir d'une Activity ?
- Comment attendre la réponse ?

Solution

- Utiliser une librairie spécialisée :
 - Pour envoyer la requête
 - Pour attendre la réponse
- Apache HttpRequest
 - Deprecated ☹
- Volley
 - Préconisé par Google
 - Pour les petites requêtes (pas le streaming)

Principe général

- Créer un client HTTP
- Créer une requête
 - Contient l'URL et ses arguments
- Demander au client d'exécuter la requête
 - Le client attend la réponse
 - Quand le client a reçu la réponse, il appelle un méthode (*listener*) de votre programme.
- Traiter la réponse dans la méthode *listener*

Exemple avec Volley

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    final TextView mTextView = (TextView) findViewById(R.id.text);

    // Instantiate the RequestQueue.
    RequestQueue queue = Volley.newRequestQueue(this);
    String url = "https://www.googleapis.com/books/v1/volumes?q=flowers+inauthor:keyes";

    // Request a string response from the provided URL.
    StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                // Display the first 500 characters of the response string.
                mTextView.setText("Response is: " + response.substring(0,500));
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                mTextView.setText("That didn't work!");
            }
        });

    // Add the request to the RequestQueue.
    queue.add(stringRequest);
}
```

Utiliser Volley

- Ajouter une dépendance dans build.gradle :

```
dependencies {  
    compile 'com.android.volley:volley:1.0.0'  
}
```

- Ou cloner le code de volley et le positionner en tant que 'library project'

- `git clone https://android.googlesource.com/platform/frameworks/volley`

- Ajouter une permission (dans le MANIFEST):

- `android.permission.INTERNET`

- Voir

- <https://developer.android.com/training/volley/index.html>
 - <https://developer.android.com/training/volley/simple.html>

Ajouter une permission dans le Manifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="...">

    <application
        ...
    </application>

    <uses-permission android:name="android.permission.INTERNET" />
</manifest>
```

Parser la réponse (JSON)

JSON - JavaScript Object Notation

- <http://www.json.org/>
- {} : object
- [] : array of objects
- "String": "value",
- Exemple : Google Book

```
{  
  "kind": "books#volumes",  
  "totalItems": 100,  
  
  "items": [  
  
    {  
      "kind": "books#volume",  
      "id": "8Pr_kLFxcYc",  
      ...  
    },  
  
    {  
      "kind": "books#volume",  
      "id": "pIs9Em38dAoC",  
      ...  
    },  
  
  ],  
}
```


JSON – structure d'un livre

```
{
  "kind": "books#volume",
  "id": "8Pr_kLFxcYc",
  "volumeInfo": {
    "title": "Flowers For Algernon",
    "authors": [
      "Daniel Keyes"
    ],
    "publisher": "Hachette UK",
    "publishedDate": "2012-11-15",
    "description": "The classic novel ...",
    "industryIdentifiers": [
      {
        "type": "ISBN_13",
        "identifier": "9780575088498"
      },
      {
        "type": "ISBN_10",
        "identifier": "0575088494"
      }
    ],
    "imageLinks": {
      "smallThumbnail": "http://books.google.com/books/content?...",
      "thumbnail": "http://books.google.com/books/content?...i"
    }
  }
}
```

Analyser la réponse JSON

```
{
  "kind": "books#volumes",
  "items": [
    {
      "kind": "books#volume",
      "id": "_ojXNuzgHRcC",
      "etag": "OTD2tB19qn4",
      "selfLink":
"https://www.googleapis.com/books
/v1/volumes/_ojXNuzgHRcC",
      "volumeInfo": {
        "title": "Flowers",
        "authors": [
          "Vijaya Khisty Bodach"
        ],
        ...
      },
      . . .
    }
  ],
  "totalItems": 1
}
```

```
import org.json.JSONArray;
import org.json.JSONObject;
```

```
JSONObject resultObject =
    new JSONObject(result);

JSONArray bookArray =
    resultObject.getJSONArray("items");

JSONObject bookObject =
    bookArray.getJSONObject(0);

JSONObject volumeObject =
    bookObject.getJSONObject("volumeInfo");

//try for title
String title = volumeObject.getString("title");
```

Document

- Exemple Book Scanner
 - <http://code.tutsplus.com/tutorials/android-sdk-create-a-book-scanning-app-interface-book-search--mobile-17790>
- **Android JSON Parsing Tutorial**
 - <http://www.androidhive.info/2012/01/android-json-parsing-tutorial/>

Bibliographie

- **Android SDK: Create a Book Scanning App**
 - <http://code.tutsplus.com/tutorials/android-sdk-create-a-book-scanning-app--mobile-17634>
 - <http://code.tutsplus.com/tutorials/android-sdk-create-a-book-scanning-app-interface-book-search--mobile-17790>
 - <http://code.tutsplus.com/tutorials/android-sdk-create-a-book-scanning-app-part-3--mobile-17880>