

Recherches de sources

Fournisseurs actuels

Fournisseur	Auth requise	Données reçues
Allocine	Oui	XML/JSON
CristalZik	-	-
Deezer	Non	JSON
Electre	Oui	JSON
Gallica	Non	XML
INA	Non	JSON
Izneo	-	-
LastFM	Oui	XML/JSON
Libfly	-	-
MusicMe	Oui	XML
Spotify	Oui	JSON
Wikipédia	Non	JSON

Site référençant toutes les APIs du web pour la musique: <http://musicmachinery.com/music-apis/>

Fournisseurs envisageables

Fournisseur	Auth requise	Données reçues
IMDB	Oui	JSON
ITunes	Non	JSON
OpenLibrary	Non	JSON
WorldCat	Non	XML, HTML, JSON, PYTHON, RUBY, TEXT, CSV
Google Books	Non	JSON

Détails

1. Allocine

1.1 Fonction

- Récupérer des données sur les films

1.2 Utilisation

- Url de base `http://api.allocine.fr/rest/v3/`
- Pour pouvoir récupérer des données depuis l'API d'allociné, il est nécessaire de renseigner un code partenaire qui sert de signature.

Plus de renseignements

1.2.1 Recherche

- Url : `http://api.allocine.fr/rest/v3/search`

Paramètres :

Paramètre	valeur
partner	Code partenaire
q	Chaine à chercher
format(optionnel)	Renvoie le résultat sous format XML ou JSON
filter(optionnel)	Filtrer selon un type de résultat (movie, theater, person, news, tvseries)
count(optionnel)	Nombre de résultats à renvoyer
page(optionnel)	Numéro de la page de résultats à afficher (10 résultats par page par défaut)

Exemple de recherche :

```
http://api.allocine.fr/rest/v3/search?  
partner=QUNXZWItQWxsbn0Npbuk&filter=movie,theater,person,news,tvseries&count=5&page=1&q=  
avatar&format=json
```

1.2.2 Informations sur un film

- Url : `http://api.allocine.fr/rest/v3/movie`

Paramètres :

Paramètre	valeur
partner	Code partenaire
code	Identifiant du film
profile (optionnel)	Degré d'informations renvoyé (small, medium, large)
mediafmt (optionnel)	Format vidéo (flv, mp4-lc, mp4-hip, mp4-archive, mpeg2-theater, mpeg2

	...)
format (optionnel)	Renvoie le résultat en format JSON ou XML
filter (optionnel)	Filtrer selon un type de résultat (movie, theater, person, news, tvseries)
striptags (optionnel)	Supprime les tags HTML des paramètres valeurs passées en paramètre

Exemple de récupération d'information sur un film :

```
http://api.allocine.fr/rest/v3/movie?
partner=QUNXZWItQWxs0Npbuk&code=61282&profile=large&mediafmt=mp4-
lc&format=json&filter=movie&striptags=synopsis,synopsisshort
```

1.2.3 Critiques sur un film

- Url : `http://api.allocine.fr/rest/v3/reviewlist`

Paramètres :

Paramètre	valeur
partner	Code partenaire
type	Type (movie, ...)
code	Identifiant du film
filter	Type de critique (desk-press ou public)
count	Nombre de critiques à renvoyer
page	Numéro de la page des résultats à afficher
format	JSON ou XML

Exemples de récupération de critique(s) sur un film :

```
http://api.allocine.fr/rest/v3/reviewlist?
partner=QUNXZWItQWxs0Npbuk&type=movie&code=61282&filter=public&count=30&format=json
```

```
http://api.allocine.fr/rest/v3/reviewlist?
partner=QUNXZWItQWxs0Npbuk&type=movie&filter=desk-
press&code=61282&count=30&format=json
```

2. CrystalZik

-

3. Deezer

3.1 Fonction

- Récupérer des informations sur des albums de musiques

3.2 Utilisation

- Url de base `https://api.deezer.com/version/service/id/method/?parameters`
- Pas de token d'authentification nécessaire

Plus d'informations

3.2.1 Informations sur un album

Paramètre	valeur
id	L'id de l'album deezer
title	Le titre de l'album
upc	Le numéro UPC de l'album
link	L'url de l'album sur deezer
...	...

Plus de paramètres

Exemple :

- `http://api.deezer.com/album/302127`

on recevra dans un JSON, l'ensemble des albums des daft punks

- `http://api.deezer.com/album/upc:724384960650`

on recevra comme ci-dessous les information sur l'album Discovery d'upc 724384960650

```
{
  "id": 302127,
  "title": "Discovery",
  "upc": "724384960650",
  "link": "http://www.deezer.com/album/302127",
  "share": "http://www.deezer.com/album/302127?utm_source=deezer&utm_content=alb",
  "cover": "http://api.deezer.com/album/302127/image",
  "cover_small": "http://e-cdn-images.deezer.com/images/cover/2e018122cb5698627",
  "cover_medium": "http://e-cdn-images.deezer.com/images/cover/2e018122cb569862",
  "cover_big": "http://e-cdn-images.deezer.com/images/cover/2e018122cb569862771",
  "genre_id": 113,
  "genres": {
    "data": [{
      "id": 113,
      "name": "Dance",
      "picture": "http://api.deezer.com/genre/113/image",
      "type": "genre"
    }]
  }
}
```

```
    }}
  },
  "label": "Parlophone France",
  "nb_tracks": 14,
  "duration": 3660,
  "fans": 142788,
  "rating": 0,
  "release_date": "2001-03-07",
  "record_type": "album",
  "available": true,
  "tracklist": "http://api.deezer.com/album/302127/tracks",
  "explicit_lyrics": false,
  "contributors": [{
    "id": 27,
    "name": "Daft Punk",
    "link": "http://www.deezer.com/artist/27",
    "share": "http://www.deezer.com/artist/27?utm_source=deezer&utm_content=ar",
    "picture": "http://api.deezer.com/artist/27/image",
    "picture_small": "http://e-cdn-images.deezer.com/images/artist/f2bc007e91",
    "picture_medium": "http://e-cdn-images.deezer.com/images/artist/f2bc007e9",
    "picture_big": "http://e-cdn-images.deezer.com/images/artist/f2bc007e9133",
    "radio": true,
    "tracklist": "http://api.deezer.com/artist/27/top?limit=50",
    "type": "artist",
    "role": "Main"
  }],
  "artist": {
    "id": 27,
    "name": "Daft Punk",
    "picture": "http://api.deezer.com/artist/27/image",
    "picture_small": "http://e-cdn-images.deezer.com/images/artist/f2bc007e91",
    "picture_medium": "http://e-cdn-images.deezer.com/images/artist/f2bc007e9",
    "picture_big": "http://e-cdn-images.deezer.com/images/artist/f2bc007e9133",
    "tracklist": "http://api.deezer.com/artist/27/top?limit=50",
    "type": "artist"
  },
  "type": "album",
  "tracks": {
    "data": [{
      "id": 3135553,
      "readable": true,
      "title": "One More Time",
      "title_short": "One More Time",
      "title_version": "",
      "link": "http://www.deezer.com/track/3135553",
      "duration": 320,
      "rank": 861450,
      "explicit_lyrics": false,
      "preview": "https://e-cdns-preview-4.dzcdn.net/stream/43808a3ac856cc11",
      "artist": {
        "id": 27,
        "name": "Daft Punk",
```

```

        "tracklist": "http:\\\\api.deezer.com\\artist\\27\\top?limit=50",
        "type": "artist"
    },
    "type": "track"
}
...
]
}
}

```

4. Electre

4.1 Fonction

- Récupérer des informations sur un livre ou un auteur

4.2 Utilisation

- Url de base `http://www.electre.com/webservice/search.asmx`
- Authentification nécessaire avec un token d'authentification

[Plus d'informations](#)

4.2.1 Informations sur un auteur

Paramètre	valeur
sessionToken	Token de la session
auteurId	Id de l'auteur

[Plus de paramètres](#)

Exemple :

- On envoie `/webservice/search.asmx/findAuteur?sessionToken=string&auteurId=string` et on recevra :

```

<?xml version="1.0" encoding="utf-8"?>
<AuteurInfo xmlns="http://electre.com/ElectreNET/literalTypes">
  <nom>string</nom>
  <id>string</id>
  <auteurType>AUTMANIF or AUTEURPP or AUTCOLLE</auteurType>
  <typParti>
    <string>string</string>
    <string>string</string>
  </typParti>
</AuteurInfo>

```

5. Gallica

5.1 Fonction

- Le projet Gallica de la BNF donne accès à de nombreux fonds documentaires issus des bibliothèques françaises sous forme électronique.

5.2 Utilisation

- Pas de token d'authentification nécessaire

Plus d'informations

5.2.1 Recherche par mot clé

Il n'y a pas de documentation disponible pour cette API, mais voici un exemple de recherche par mot clé :

Exemple :

- On envoie `http://gallica.bnf.fr/SRU?operation=searchRetrieve&version=1.2&maximumRecords=1&startRecord=1&query=text%20any%20%22aujourd%27hui%22 any "aujourd%27hui"`

on recevra dans un XML le premier document de la base contenant le mot "aujourd'hui" sous format XML

```
<?xml version="1.0" encoding="UTF-8"?>
<srw:searchRetrieveResponse xmlns:srw="http://www.loc.gov/zing/srw/" xmlns="http://gal
  <srw:version>1.2</srw:version>
  <srw:echoedSearchRetrieveRequest>
    <srw:query>text any "aujourd'hui"</srw:query>
    <srw:version>1.2</srw:version>
  </srw:echoedSearchRetrieveRequest>
  <srw:numberOfRecords>157117</srw:numberOfRecords>
  <srw:records>
    <srw:record>
      <srw:recordSchema>http://www.openarchives.org/OAI/2.0/OAIdc.xsd</srw:recordSc
      <srw:recordPacking>xml</srw:recordPacking>
      <srw:recordData>
        <oai_dc:dc>
          <dc:creator>Conseil international des musées</dc:creator>
          <dc:date>19??</dc:date>
          <dc:format>application/pdf</dc:format>
          <dc:identifier>http://gallica.bnf.fr/ark:/12148/cb34348702t/date</dc:id
          <dc:identifier>ISSN 10206426</dc:identifier>
```

```

    <dc:language>eng</dc:language>
    <dc:publisher>ICOM (Paris)</dc:publisher>
    <dc:relation>Notice du catalogue : http://catalogue.bnf.fr/ark:/12148/c
    <dc:source>ICOM (Conseil International des musées), 2013-311990</dc:sou
    <dc:title>Nouvelles de l'ICOM</dc:title>
    <dc:type>text</dc:type>
    <dc:type>text</dc:type>
    <dc:type>publication en série imprimée</dc:type>
    <dc:type>printed serial</dc:type>
  </oai_dc:dc>
</srw:recordData>
<srw:recordIdentifier>ark</srw:recordIdentifier>
<srw:recordPosition>0</srw:recordPosition>
<srw:extraRecordData>
  <epubFile />
  <infoSupModifiable />
  <link>http://gallica.bnf.fr/ark:/12148/cb34348702t/date</link>
  <nqamoyen>99.93</nqamoyen>
  <thumbnail />
  <typedoc>periodiques</typedoc>
</srw:extraRecordData>
</srw:record>
</srw:records>
<srw:nextRecordPosition>2</srw:nextRecordPosition>
</srw:searchRetrieveResponse>

```

6. INA

6.1 Fonction

-L'INA donne accès à du contenu audiovisuel et multimédia gratuitement

6.2 Utilisation

- Url de base `http://fresques.ina.fr/jalons/api`
- Pas de token d'authentification nécessaire

[Plus d'informations](#)

6.2.1 Récupérer toutes les ressources

La syntaxe est la suivante : `/api/search ? query={QUERY} & limit={LIMIT} & offset={OFFSET} & mime={MIME}`

Paramètre	valeur
SORT(optionnel)	Propriété sur laquelle trier les ressources
DIRECTION(optionnel)	Direction de tri (asc/desc)

LIMIT(optionnel)	Nombre maximal de ressources renvoyées (par défaut : 50)
OFFSET(optionnel)	Offset de la première ressource renvoyée (par défaut : 0)
MIME(optionnel)	Format de retour (xml/json/csv)

6.2.2 Récupérer une ressource cible

La syntaxe est la suivante : `/api/resource/{ID} ? mime={MIME}`

Paramètre	valeur
ID(obligatoire)	Identifiant de ressource
MIME(optionnel)	Format de retour (XML/JSON/CSV)

Exemple : On envoie `http://fresques.ina.fr/jalons/api/search?query=président&limit=1` pour récupérer le premier article contenant le mot "président"

On reçoit sous forme de JSON :

```
{
  "version": "1.0",
  "provider_name": "Ina.fr",
  "provider_url": "http://fresques.ina.fr",
  "author_name": "Fresque hypermedia",
  "author_url": "",
  "action": "search",
  "return": {
    "total": 49,
    "offset": 0,
    "limit": 1,
    "resources": [{
      "id": "InaEdu00015",
      "title": "L'éllection du premier président de la IVe République",
      "description": "<p>Selon la nouvelle Constitution, le président de la",
      "duration": 113.8,
      "date": "1947-01-23",
      "locale": "fra",
      "type": "video",
      "width": 512,
      "height": 384,
      "html": "<iframe allowfullscreen=\"true\" frameborder=\"0\" marginheight=\"",
      "url": "http://fresques.ina.fr/jalons/fiche-media/InaEdu00015/1-elec",
      "api": "http://fresques.ina.fr/jalons/api/resource/InaEdu00015",
      "thumbnail": "http://fresques.ina.fr/jalons/media/imagette/80x60/In",
      "thumbnail_width": 80,
      "thumbnail_height": 60
    }]
  }
}
```

7. Izneo

-

8. LastFM

8.1 Fonction

- LastFM donne accès à des informations sur du contenu musical

8.2 Utilisation

- Url de base `http://ws.audioscrobbler.com/2.0/`
- Une clé d'API nécessaire

8.2.1 Récupérer des informations sur un album

Paramètre	valeur
artist(obligatoire)	Le nom de l'artiste
album(obligatoire)	Le nom de l'album
mbid(obligatoire)	Le numéro mbid de l'album
autocorrect[0/1] (optionnel)	Transforme les mots mal orthographiés par le nom correct de l'artiste à la place. Le nom correct de l'artiste est retourné dans la réponse
username(optionnel)	Le nom d'utilisateur
lang(optionnel)	La langue de retour
api_key(obligatoire)	Une clé LastFM pour l'API
format(optionnel)	format des données reçues(json/xml)

Exemple 1 :

On envoie : `http://ws.audioscrobbler.com/2.0/?`

`method=album.getinfo&api_key=YOUR_API_KEY&artist=Cher&album=Believe` pour récupérer les informations sur un album (ici l'album Believe de Cher).

On reçoit des données XML de ce type :

```
<album>
  <name>Believe</name>
  <artist>Cher</artist>
  <id>2026126</id>
  <mbid>61bf0388-b8a9-48f4-81d1-7eb02706dfb0</mbid>
```

```

<url>http://www.last.fm/music/Cher/Believe</url>
<releasedate>6 Apr 1999, 00:00</releasedate>
<image size="small">...</image>
<image size="medium">...</image>
<image size="large">...</image>
<listeners>47602</listeners>
<playcount>212991</playcount>
<toptags>
  <tag>
    <name>pop</name>
    <url>http://www.last.fm/tag/pop</url>
  </tag>
  ...
</toptags>
<tracks>
  <track rank="1">
    <name>Believe</name>
    <duration>239</duration>
    <mbid/>
    <url>http://www.last.fm/music/Cher/\_/Believe</url>
    <streamable fulltrack="0">1</streamable>
    <artist>
      <name>Cher</name>
      <mbid>bfcc6d75-a6a5-4bc6-8282-47aec8531818</mbid>
      <url>http://www.last.fm/music/Cher</url>
    </artist>
  </track>
  ...
</tracks>
</album>

```

Exemple 2 :

On envoie `http://ws.audioscrobbler.com/2.0/?`

`method=album.gettags&artist=cher&album=believe&api_key=YOUR_API_KEY&format=json` pour récupérer la liste des tags d'un album

On reçoit :

```

<tags artist="Sally Shapiro" album="Disco Romance">
  <tag>
    <name>swedish</name>
    <url>http://www.last.fm/tag/swedish</url>
  </tag>
  ...
</tags>

```

9. Libfly

Liens morts :

- http://www.libfly.com/api_groupe.html
- <http://www.libfly.com/Webservice/api/index.php>

10. MusicMe

11. Spotify

11.1 Fonction

- Spotify donne accès à des informations sur du contenu musical

11.2 Utilisation

- Url de base `https://api.spotify.com/v1/`
- Aucune clé d'identification nécessaire

[Plus d'informations](#)

8.2.1 Récupérer des informations sur un album

La syntaxe est la suivante : `https://api.spotify.com/v1/albums/{id}` , id étant l'id de l'album spotify.

Exemple :

On envoie `https://api.spotify.com/v1/albums/0sNOF9WDwhWunNAHPD3Baj`

Les données reçues sont de ce type :

```
{
  "album_type" : "album",
  "artists" : [ {
    "external_urls" : {
      "spotify" : "https://open.spotify.com/artist/2BTZIQw0ntH9MvilQ3ewNY"
    },
    "href" : "https://api.spotify.com/v1/artists/2BTZIQw0ntH9MvilQ3ewNY",
    "id" : "2BTZIQw0ntH9MvilQ3ewNY",
    "name" : "Cyndi Lauper",
    "type" : "artist",
    "uri" : "spotify:artist:2BTZIQw0ntH9MvilQ3ewNY"
  } ],
  "available_markets" : [ "AD", "AR", "AT", "AU", "BE", "BG", "BO", "BR", "CA", "CH",
  "copyrights" : [ {
    "text" : "(P) 2000 Sony Music Entertainment Inc.",
    "type" : "P"
  } ]
}
```

```
} ],
"external_ids" : {
  "upc" : "5099749994324"
},
"external_urls" : {
  "spotify" : "https://open.spotify.com/album/0sNOF9WDwhWunNAHPD3Baj"
},
"genres" : [ ],
"href" : "https://api.spotify.com/v1/albums/0sNOF9WDwhWunNAHPD3Baj",
"id" : "0sNOF9WDwhWunNAHPD3Baj",
"images" : [ {
  "height" : 640,
  "url" : "https://i.scdn.co/image/07c323340e03e25a8e5dd5b9a8ec72b69c50089d",
  "width" : 640
}, {
  "height" : 300,
  "url" : "https://i.scdn.co/image/8b662d81966a0ec40dc10563807696a8479cd48b",
  "width" : 300
}, {
  "height" : 64,
  "url" : "https://i.scdn.co/image/54b3222c8aaa77890d1ac37b3aaaa1fc9ba630ae",
  "width" : 64
} ],
"name" : "She's So Unusual",
"popularity" : 39,
"release_date" : "1983",
"release_date_precision" : "year",
"tracks" : {
  "href" : "https://api.spotify.com/v1/albums/0sNOF9WDwhWunNAHPD3Baj/tracks?offset=0",
  "items" : [ {
    "artists" : [ {
      "external_urls" : {
        "spotify" : "https://open.spotify.com/artist/2BTZIQw0nth9MvilQ3ewNY"
      },
      "href" : "https://api.spotify.com/v1/artists/2BTZIQw0nth9MvilQ3ewNY",
      "id" : "2BTZIQw0nth9MvilQ3ewNY",
      "name" : "Cyndi Lauper",
      "type" : "artist",
      "uri" : "spotify:artist:2BTZIQw0nth9MvilQ3ewNY"
    } ],
    "available_markets" : [ "AD", "AR", "AT", "AU", "BE", "BG", "BO", "BR", "CA", "C",
    "disc_number" : 1,
    "duration_ms" : 305560,
    "explicit" : false,
    "external_urls" : {
      "spotify" : "https://open.spotify.com/track/3f9zqUnrnIq0LANhmnaF0V"
    },
    "href" : "https://api.spotify.com/v1/tracks/3f9zqUnrnIq0LANhmnaF0V",
    "id" : "3f9zqUnrnIq0LANhmnaF0V",
    "name" : "Money Changes Everything",
    "preview_url" : "https://p.scdn.co/mp3-preview/01bb2a6c9a89c05a4300aea427241b171",
    "track_number" : 1,
```

```

    "type" : "track",
    "uri" : "spotify:track:3f9zqUnrnIq0LANhmnaF0V"
  }, {
    ...
  } ],
  "limit" : 50,
  "next" : null,
  "offset" : 0,
  "previous" : null,
  "total" : 13
},
"type" : "album",
"uri" : "spotify:album:0sNOF9WDwhWunNAHPD3Baj"
}

```

12. Wikipédia

12.1 Fonction

- Wikipédia sert à récupérer des informations dans notre cas sur les auteurs(biographie etc ...)

12.2 Utilisation

- Url de base `https://fr.wikipedia.org/w/api.php`
- Aucune clé d'identification nécessaire

[Plus d'informations](#)

12.2.1 Récupérer des informations sur un auteur

[Paramètres et options](#)

Exemple :

On envoie la requête suivante : `https://fr.wikipedia.org/w/api.php?format=json&action=query&titles=Muse_(groupe)&prop=extracts`

On reçoit les données suivantes :

```

{
  "batchcomplete": "",
  "query": {
    "normalized": [{
      "from": "Muse_(groupe)",
      "to": "Muse (groupe)"
    }],
    "pages": {

```

```
    "62841": {
      "pageid": 62841,
      "ns": 0,
      "title": "Muse (groupe)",
      "extract": "<p><b>Muse</b> est un groupe de rock britannique originair
    }
  }
}
```

13. IMDB

13.1 Fonction

- IMDB sert à récupérer des informations sur les films et les acteurs

13.2 Utilisation

- Le nombre de requêtes maximum par jour est limité au nombre de 2000
- Le nombre de requêtes effectuées toutes les 10 secondes est de 15
- Une authentification est requise pour pouvoir effectuer des requêtes
- Url de base `http://www.myapifilms.com/imdb/`

[Plus d'informations](#)

14. iTunes

14.1 Fonction

- Itunes sert à rechercher des musiques, artistes, films, livres, auteurs ...

14.2 Utilisation

- L'API d'Itunes ne requiert pas de token pour s'authentifier
- Url de base `https://itunes.apple.com/search`

[Plus d'informations](#)

14.2.1 Récupérer des informations sur un groupe de musique

[Liste des paramètres](#)

Exemple 1 :

On envoie `https://itunes.apple.com/search?term=shaka%20ponk&attribute=musicArtist` pour

rechercher des informations sur Shaka Ponk. L'attribut `attribute=musicArtist` indique bien que l'on recherche un artiste du domaine musical.

On reçoit :

```
{
  "resultCount": 1,
  "results": [{
    "wrapperType": "artist",
    "artistType": "Artist",
    "artistName": "Shaka Ponk",
    "artistLinkUrl": "https://itunes.apple.com/us/artist/shaka-ponk/id90118544?uo=",
    "artistId": 90118544,
    "amgArtistId": 995943,
    "primaryGenreName": "Rock",
    "primaryGenreId": 21
  }]
}
```

Exemple 2 :

On envoie `http://itunes.apple.com/search?term=deadpool&country=FR&entity=movie` pour rechercher des informations sur Deadpool

On obtient :

```
{
  "resultCount": 1,
  "results": [{
    "wrapperType": "track",
    "kind": "feature-movie",
    "trackId": 1078113028,
    "artistName": "Tim Miller",
    "trackName": "Deadpool",
    "trackCensoredName": "Deadpool",
    "trackViewUrl": "https://itunes.apple.com/fr/movie/deadpool/id1078113028?uo=4",
    "artworkUrl30": "http://is2.mzstatic.com/image/thumb/Video49/v4/2d/9d/63/2d9d6",
    "artworkUrl60": "http://is2.mzstatic.com/image/thumb/Video49/v4/2d/9d/63/2d9d6",
    "artworkUrl100": "http://is2.mzstatic.com/image/thumb/Video49/v4/2d/9d/63/2d9d",
    "releaseDate": "2016-02-10T08:00:00Z",
    "collectionExplicitness": "notExplicit",
    "trackExplicitness": "notExplicit",
    "country": "FRA",
    "currency": "EUR",
    "primaryGenreName": "Action et aventure",
    "contentAdvisoryRating": "-12",
    "shortDescription": "DEADPOOL est l'antihéros le plus atypique de l'univers Ma",
    "longDescription": "DEADPOOL est l'antihéros le plus atypique de l'univers Mar
```



```
}]
}
```

15. OpenLibrary

15.1 Fonction

- OpenLibrary est un site web regroupant des informations sur des milliers de livres fourni par des contributeurs.

15.2 Utilisation

- L'API ne requiert pas d'authentification.
- Url de base `https://openlibrary.org/dev/docs/api/`

[Plus d'infomations](#)

15.2.1 Récupérer des informations sur un livre à partir de son ISBN

[Liste des paramètres.](#)

Exemple 1 :

On recherche les informations sur le livre "Le seigneur des anneaux" à l'aide de son ISBN :

```
http://openlibrary.org/api/books?bibkeys=ISBN:9782070612888&format=json
```

On obtient ce résultat :

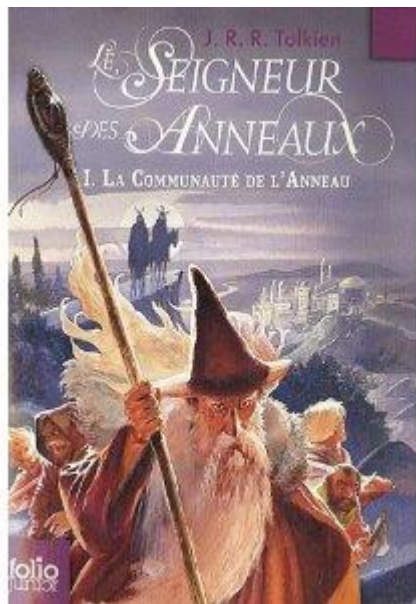
```
{
  "ISBN:9782070612888": {
    "bib_key": "ISBN:9782070612888",
    "preview": "noview",
    "thumbnail_url": "https://covers.openlibrary.org/b/id/6425390-S.jpg",
    "preview_url": "http://openlibrary.org/books/OL24298430M/Le_Seigneur_des_Annea",
    "info_url": "http://openlibrary.org/books/OL24298430M/Le_Seigneur_des_Anneaux_"
  }
}
```

Exemple 2 :

On recherche une image de couverture pour le livre "seigneur des anneaux" à partir de son ISBN :

```
http://covers.openlibrary.org/b/isbn/9782070612888-L.jpg La syntaxe est la suivante :
http://covers.openlibrary.org/b/isbn/<ISBN>-<SIZE>.jpg
```

On obtient ce résultat :



16. WorldCat

16.1 Fonction

- WorldCat est un site regroupant un grand nombre de livres

16.2 Utilisation

- L'API requiert une clé pour soumettre sa requête
- L'API met à disposition un [évaluateur](#) pour construire ses requêtes (seulement pour la recherche par termes)

Plus d'infomations

Paramètres	Valeur
method (obligatoire)	to10, to13, fixChecksum, getMetadata, getEditions
format (obligatoire)	xml, html, json, python, ruby, text, csv
library (optionnel)	Limite la recherche à une librairie. Valeurs possibles: ebook, bookmooch, wikipedia, paperbackswap, oca, freeebook, etc...
fl (optionnel)	Le parametre "fl" précise les types de champs à afficher. Exemple: fl=* retourne tous les champs, fl=ed, retourne les champs d'edition et de langage.
startIndex (optionnel)	L'index de la première recherche effectuée par le client.
count (optionnel)	Le nombre de résultats à afficher
callback (optionnel)	fonction callback JSON

ai (optionnel)	
token (optionnel)	abonnement
hash (optionnel)	abonnement

16.2.1 Récupérer des informations sur un livre à partir de son ISBN

Exemple 1 : On envoie la requête suivante :

`http://xisbn.worldcat.org/webservices/xid/isbn/0596002815` sans préciser de méthode (avec celle par défaut)

On reçoit le XML suivant :

```
<rsp xmlns="http://worldcat.org/xid/isbn/" stat="ok">
  <isbn>0596002815</isbn>
  <isbn>0596551932</isbn>
  <isbn>0596158068</isbn>
  <isbn>1565928938</isbn>
  <isbn>1449355730</isbn>
  <isbn>0596516800</isbn>
  <isbn>0596513984</isbn>
  <isbn>0596516606</isbn>
  <isbn>1449355692</isbn>
  <isbn>7564112409</isbn>
  <isbn>1449391753</isbn>
  <isbn>1449355714</isbn>
  <isbn>144937932X</isbn>
  <isbn>8184048262</isbn>
  <isbn>0596554494</isbn>
  <isbn>1565924649</isbn>
  <isbn>059680539X</isbn>
</rsp>
```

On a donc récupéré une liste des éditions correspondantes à l'ISBN 0596002815

17. Google Books

17.1 Fonction

- Google books sert à obtenir des informations sur certains livres

17.2 Utilisation

- Pas de clé requise
- [Doc officielle de Google](#)

- [Plus d'informations](#)

17.2.1 Récupérer des informations sur un livre à partir de son ISBN

Exemple : On envoie la requête suivante : `https://www.googleapis.com/books/v1/volumes?q=isbn:1449355730` pour récupérer des informations sur le livre passé en paramètre grâce à son ISBN.

On obtient ceci :

```
{
  "kind": "books#volumes",
  "totalItems": 1,
  "items": [
    {
      "kind": "books#volume",
      "id": "VCINBQAAQBAJ",
      "etag": "7ERBV/Qv4lU",
      "selfLink": "https://www.googleapis.com/books/v1/volumes/VCINBQAAQBAJ",
      "volumeInfo": {
        "title": "Learning Python",
        "subtitle": "5th Edition",
        "authors": [
          "Mark Lutz"
        ],
        "publisher": "O'Reilly Media; Fifth Edition edition (July 6, 2013)",
        "publishedDate": "2013-07-06",
        "description": "Get a comprehensive, in-depth introduction to the core Python lang",
        "industryIdentifiers": [
          {
            "type": "ISBN_13",
            "identifier": "9781449355739"
          },
          {
            "type": "ISBN_10",
            "identifier": "1449355730"
          }
        ],
        "readingModes": {
          "text": false,
          "image": false
        },
        "pageCount": 1594,
        "printType": "BOOK",
        "categories": [
          "Computers"
        ],
        "averageRating": 3.5,
        "ratingsCount": 53,
        "maturityRating": "NOT_MATURE",
        "allowAnonLogging": false,
```

