|  |
| --- |
| Criminalité sur Internet |
| Sous-projet A: Clear Web – Scripts d’acquisition |

[Moteurs de recherche 1](#_Toc416790066)

[Remarque préliminaire 1](#_Toc416790067)

[Google 1](#_Toc416790068)

[API - Custom Search 1](#_Toc416790069)

[Prix et limitations 1](#_Toc416790070)

[Prérequis au programme 2](#_Toc416790071)

[Fonctions du programme google.js 3](#_Toc416790072)

[Données obtenues 4](#_Toc416790073)

[Bing 5](#_Toc416790074)

[API – Bing Search API 5](#_Toc416790075)

[Prix et limitations 5](#_Toc416790076)

[Prérequis au programme 5](#_Toc416790077)

[Fonctions du programme bing.js 6](#_Toc416790078)

[Données obtenues 7](#_Toc416790079)

[Yahoo 8](#_Toc416790080)

[API – BOSS Search API 8](#_Toc416790081)

[Prix et limitations 8](#_Toc416790082)

[Prérequis au programme 8](#_Toc416790083)

[Fonctions du programme yahoo.js 9](#_Toc416790084)

[Données obtenues 10](#_Toc416790085)

Les scripts fonctionnent avec NodeJS dans sa version 0.12.2[[1]](#footnote-1).

# Moteurs de recherche

## Google

### API - Custom Search

Google propose une interface de programmation (API), nommée « Custom Search »[[2]](#footnote-2) intéressante dans le cadre de l’acquisition de résultats de recherche par mots-clefs. Initialement cette API permet d’intégrer la fonction de recherche de Google à un site Internet pour les afficher directement sur la page. Elle permet donc d’acquérir les données de recherches parmi les sites Internet et les images.

### Prix et limitations

Il est possible d’effectuer 100 requêtes/jour au maximum. Avec un maximum de 10 résultats par requête. Pour dépasser cette limite il faut entrer ses données de facturation.

Le coût est de 5$ pour 1000 requêtes supplémentaires avec un maximum de 10’000/jour.

De plus la cadence des requêtes ne doit pas être supérieure à une par seconde.

### Prérequis au programme

1. Création d’un compte [Google](https://accounts.google.com/SignUp?continue).
2. Création d’un [projet API](https://code.google.com/apis/console/?pli=1) :

* Activer l’API  (Onglet : API et authentification > API): **Custom Search API** .
* Créer une clé d’accès à l’API publique (API et authentification > Identifiants) : **Clé de navigateur**, puis cliquer sur créer.

La clé obtenue est à insérer dans le fichier *keys.js*dans la variable exports.googleKey (exports.googleKey= " clef\_à\_copier" ;)

1. Création du [moteur de recherche](https://www.google.com/cse/create/new) virtuel, **Google Search Engine**:

* Entrer un nom de site quelconque (par ex : *www.example.com*) et un nom de moteur de recherche, puis cliquer sur créer.
* Modifier ensuite le moteur de recherche (panneau de configuration) :
* **Activer** la recherche d’image.
* **Désactiver** la recherche vocale.
* Sélectionner **Chercher sur l’ensemble du Web en privilégiant les sites inclus** dans la liste déroulante pour les sites sur lesquels effectuer des recherches.
* **Supprimer** l’URL du site entrer lors de la création (il ne doit pas y avoir de site inclus).
* Cliquer sur **mettre à jour.**

Dans le panneau de configuration cliquer sur **Identifiant du moteur de recherche**. L’identifiant obtenu est à insérer dans le fichier *keys.js*dans la variable exports.googleSearchEngine

(exports.googleSearchEngine= "identifiant\_à\_copier" ;).

1. Installation du package json (Terminal: « npm install »).

### Fonctions du programme *google.js*

Deux fonctions, webSearch et imagesSearch permettent respectivement la recherche d’URL et la recherche d’images parmi les résultats obtenus via Google.

* **webSearch** (keyword, num, opt\_args, callback)
* **searchImages**(keyword, num, opt\_args, callback)

Avec les arguments :

* **keyword** : mots-clefs utilisés pour la recherche.
* **num** : nombre de résultats à retourner, avec une requête utilisée tous les 10 résultats (100 au maximum).
* **opt\_args**: arguments optionnels sous forme de JSON.

Consultable sur : <https://developers.google.com/custom-search/json-api/v1/reference/cse/list?hl=FR> (14.04.2015).

* **callback**: fonction de callback sous la forme : function(error, result){ … }.

### Données obtenues

Les résultats obtenus à l’aide des fonctions sont en format JSON. Un tableau contenant les résultats avec pour chaque objet obtenu les données suivantes :

{

"**url**": "http://www.elitefitness.com/reports/secrets/",

"**image**": "http://www.elitefitness.com/images/guy/13.jpg",

"**request**": {

"**date**": " Tue Apr 14 2015 15:49:39 GMT+0200",

"**platform**": "Google",

"**type**": "image",

"**keyword**": "buy steroids",

"**ranking**": 9

"**opt\_args**": {hl : "fr ", gl : "ch", googlehost : " google.ch "}

}

}

Avec :

* **url** : l’URL de la page internet (pour les images : celle qui contient l’image)
* **image**: l’URL de l’image, uniquement pour imagesSearch
* **request**: les informations sur la requête effectuée :
  + **date** : la date de la requête
  + **platform** : la plateforme utilisée
  + **type**: le type de recherche effectuée (web ou image)
  + **keyword** : le/les mot(s)-clef(s) utilisé(s)
  + **ranking** : le rang d’apparition du résultat avec les paramètres utilisés pour la requête
  + **opt\_args**: un objet contenant les paramètres optionnels utilisés pour la requête.

## Bing

### API – Bing Search API

Microsoft propose une interface de programmation (API), nommée « Bing Search API »[[3]](#footnote-3) intéressante dans le cadre de l’acquisition de résultats de recherche par mots-clefs. Elle permet d’acquérir des données de recherches parmi les sites internet, les images, les vidéos ou les actualités retournés par le moteur de recherche.

### Prix et limitations

Il est possible d’effectuer **5000 requêtes/mois** au maximum gratuitement.

Avec un maximum de 50 résultats par requête. Pour dépasser cette limite il faut souscrire un abonnement payant.

Tarifs :

* 5'000 requêtes/mois : gratuit
* 10'000 requêtes/mois : 15.31 €
* 20'000 requêtes/mois : 30.61 €
* 50'000 requêtes/mois : 76.53 €
* et d’autres.

### Prérequis au programme

1. Création d’un compte [Microsoft](https://signup.live.com/).
2. Création d’un compte [Microsoft Azure Marketplace](https://datamarket.azure.com/register).
3. Connexion à [Bing Search API gratuit](https://datamarket.azure.com/dataset/bing/search) (5000 requêtes/mois).

* Connexion et acceptation des conditions.

1. Récupération de la clef du compte :
   * Accéder à : mon compte > [Clés de compte](https://datamarket.azure.com/account/keys) .
   * Copier la valeur de la clef.
   * Insérer la dans le fichier *keys.js*dans la variable exports.bingKey

(exports.bingKey= "clef\_à\_copié" ;).

1. Installation du package json (Terminal: « npm install »).

### Fonctions du programme *bing.js*

Le programme comprend les fonctions suivantes : webSearch, imagesSearch et videosSearch respectivement les recherches de site Internet, d’images ou de vidéos parmi les résultats obtenus via le moteur de recherche Bing.

* **webSearch** (keyword, num, opt\_args, callback)
* **imagesSearch** (keyword, num, opt\_args, callback)
* **videosSearch**(keyword, num, opt\_args, callback)

Avec les arguments :

* **keyword** : mots-clefs utilisés pour la recherche
* **num** : le nombre de résultats à retourner, avec une requête utilisée tous les 50 résultats.
* **opt\_args**: arguments optionnels sous forme de JSON.

Consultable sur : <https://datamarket.azure.com/dataset/bing/search#schema> (14.04.2015).

* **callback**: fonction de callback sous la forme : function(error, result){ … }.

### Données obtenues

Les résultats obtenus à l’aide des fonctions sont en format JSON. Un tableau contenant les résultats avec pour chaque objet obtenu les données suivantes :

{

"**url**": " http://bulking-cutting.com/buy-steroids.php",

"**image**": " http://bulking-cutting.com/images/steroidgroup.jpg",

"**request**": {

"**date**": "Tue Apr 14 2015 15:49:39 GMT+0200",

"**platform**": "Bing",

"**type**": "image",

"**keyword**": "buy steroids",

"**ranking**": 6

"**opt\_args**": null

}

}

Avec :

* **url** : l’URL de la page internet (pour les images et vidéos : celle qui contient le média)
* **image**: l’URL de l’image, uniquement pour imagesSearch
* **request**: les informations sur la requête effectuée :
  + **date** : la date de la requête
  + **platform** : la plateforme utilisée
  + **type**: le type de recherche effectuée (web, image ou video)
  + **keyword** : le/les mot(s)-clef(s) utilisé(s)
  + **ranking** : le rang d’apparition du résultat avec les paramètres utilisés
  + **opt\_args**: un objet contenant les paramètres optionnels utilisés pour la requête.

## Yahoo

### API – BOSS Search API

Yahoo propose une interface de programmation (API), nommée « BOSS Search API »[[4]](#footnote-4) intéressante dans le cadre de l’acquisition de résultats de recherche par mots-clefs. Elle permet d’acquérir des données de recherches parmi les sites internet et les images retournés par le moteur de recherche.

### Prix et limitations

L’API est exclusivement payante, tarifs :

* 1'000 requêtes pour les pages Web: 1.50 $
* 1000 requêtes pour les images : 0.30 $

Une requête retourne 50 et 35 résultats respectivement pour les pages Web et les images.

### Prérequis au programme

1. Création d’un compte [Yahoo](https://edit.yahoo.com/registration).
2. Création d’une [application](https://developer.apps.yahoo.com/dashboard/createKey.html?api=boss), avec les paramètres suivants :
   * Application Type : Mobile/Desktop.
   * Home Page URL : http://www.example.com
   * Access Scopes : public APIs.
   * Premium API services : coché.
3. Accéder à l’[application créée](https://developer.apps.yahoo.com/projects) et accéder au BOSS Setup afin d’y entrer les coordonnées bancaires nécessaire au fonctionnement de l’application.
4. Récupération des clefs « Consumer Key » et « Consumer Secret » de l’application créée :
   * Accéder à votre application.
   * Copier la valeur du « Consumer Key » et insérer la dans le fichier *keys.js*dans la variable exports.yahooKey (exports.yahooKey= "clef\_à\_copié" ;).
   * Copier la valeur du « Consumer Secret » et insérer la dans la variable exports.yahooSecret (exports.yahooSecret = "clef\_à\_copié" ;).
5. Installation du package json (Terminal: « npm install »).

### Fonctions du programme *yahoo.js*

Deux fonctions, webSearch et imagesSearch permettent respectivement la recherche d’URL et la recherche d’images parmi les résultats obtenus via Yahoo.

* **webSearch**(keyword, num, opt\_args, callback)
* **imagesSearch**(keyword, num, opt\_args, callback)

Avec les arguments :

* **keyword** : mots-clefs utilisés pour la recherche
* **num** : le nombre de résultats à retourner, avec une requête utilisée tous les 50 (web) ou 35 (images) résultats (1000 au max).
* **opt\_args**: arguments optionnels sous forme de JSON.

Consultable sur : <https://developer.yahoo.com/boss/search/boss_api_guide/web.html> (14.04.2015).

* **callback**: fonction de callback sous la forme : function(error, result){ … }.

### Données obtenues

Les résultats obtenus à l’aide des fonctions sont en format JSON. Un tableau contenant les résultats avec pour chaque objet obtenu les données suivantes :

{

"**url**": " http://goldenmuscles.com/steroids/should-you-buy-steroids/",

"**image**": " http://goldenmuscles.com/wp-content/uploads/2010/06/buy-steroids.jpg",

"**request**": {

"**date**": " Tue Apr 14 2015 15:49:39 GMT+0200",

"**platform**": "Yahoo",

"**type**": "image",

"**keyword**": "buy steroids",

"**ranking**": 1

"**opt\_args**": null

}

}

Avec :

* **url** : l’URL de la page internet (pour les images : celle qui contient l’image)
* **image**: l’URL de l’image, uniquement pour imagesSearch
* **request**: les informations sur la requête effectuée :
  + **date** : la date de la requête
  + **platform** : la plateforme utilisée
  + **type**: le type de recherche effectuée (web ou image)
  + **keyword** : le/les mot(s)-clef(s) utilisé(s)
  + **ranking** : le rang d’apparition du résultat avec les paramètres utilisés pour la requête
  + **opt\_args**: un objet contenant les paramètres optionnels utilisés pour la requête

# Forum

## Reddit

### API

### Prix et limitations

### Prérequis au programme

1. Uniquement l’installation du package json (Terminal: « npm install »).

### Fonctions du programme *reddit.js*

Deux fonctions, redditSearch et subredditSearch permettent respectivement la recherche de postes sur Reddit dans toute les sections ou dans une section (subreddit) particulière.

* **webSearch**(keyword, num, opt\_args, callback)
* **imagesSearch**(keyword, num, opt\_args, callback)

Avec les arguments :

* **keyword** : mots-clefs utilisés pour la recherche
* **num** : le nombre de résultats à retourner, avec une requête utilisée tous les 50 (web) ou 35 (images) résultats (1000 au max).
* **opt\_args**: arguments optionnels sous forme de JSON.

Consultable sur : <https://developer.yahoo.com/boss/search/boss_api_guide/web.html> (14.04.2015).

* **callback**: fonction de callback sous la forme : function(error, result){ … }.

### Données obtenues

1. <https://nodejs.org/download/> (consulté le 14.04.2015) [↑](#footnote-ref-1)
2. <https://developers.google.com/custom-search/json-api/v1/reference/cse/list> (consulté le 11.03.2015) [↑](#footnote-ref-2)
3. <http://datamarket.azure.com/dataset/bing/search> (consulté le 12.03.2015) [↑](#footnote-ref-3)
4. <https://developer.yahoo.com/boss/search/> (consulté le 12.03.2015) [↑](#footnote-ref-4)