

 eBook Gratuit

APPRENEZ

ibm-watson-cognitive

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#ibm-

watson-

cognitive

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec ibm-watson-cognitive.....	2
Remarques.....	2
Versions.....	2
Exemples.....	2
Obtenir les informations d'identification de l'API.....	2
Appel des API Watson avec curl.....	5
Utilisation des SDK Watson Developer Cloud.....	6
Chapitre 2: AlchemyLanguage.....	8
Remarques.....	8
Limites de taille.....	8
Support linguistique.....	8
Détection de langue.....	8
Nettoyage de texte.....	8
Exemples.....	9
Appel combiné: utilisez plusieurs fonctions dans un seul appel API (Node.js).....	9
Analyse de sentiment: obtenir des informations de sentiment pour des phrases spécifiques d.....	9
Concepts: identifier des concepts à partir d'une page Web (Node.js).....	10
Chapitre 3: Discours à texte.....	12
Remarques.....	12
Exemples.....	12
Reconnaître un fichier audio en utilisant WebSockets en Java.....	12
Transcrire un fichier audio à l'aide de WebSockets (Node.js).....	13
Chapitre 4: Reconnaissance visuelle.....	15
Exemples.....	15
Obtenir une liste de classificateurs personnalisés.....	15
Obtenir des informations sur un classificateur personnalisé spécifique.....	15
Former un classificateur personnalisé.....	15
Supprimer un classificateur personnalisé.....	16
Classifier une image.....	16

Conditions préalables	16
Classer une URL d'image	17
Chapitre 5: Récupérer et classer	19
Remarques.....	19
Exemples.....	19
Recherche et classement à l'aide de la fonction Récupérer et classer en Java.....	19
Crédits	21

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [ibm-watson-cognitive](#)

It is an unofficial and free ibm-watson-cognitive ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ibm-watson-cognitive.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec ibm-watson-cognitive

Remarques

Cette rubrique fournit des instructions de base sur l'obtention des informations d'identification pour les services Watson et fournit des liens pertinents pour chaque service et les SDK Watson Developer Cloud.

Pages d'accueil des services Watson:

- [AlchemyLanguage](#)
- [AlchemyData Nouvelles](#)
- [Conversation](#)
- [Découverte](#)
- [Conversion de documents](#)
- [La traduction de la langue](#)
- [Classificateur de langue naturelle](#)
- [Compréhension du langage naturel](#)
- [Perspectives de personnalité](#)
- [Récupérer et classer](#)
- [Discours à texte](#)
- [Texte pour parler](#)
- [Analyseur de tonalité](#)
- [Analyse des compromis](#)
- [Reconnaissance visuelle](#)

Versions

Version	Date de sortie
1.0.0	2016-05-05

Exemples

Obtenir les informations d'identification de l'API

Pour vous authentifier auprès des services Watson, vous avez besoin d'informations d'identification pour chaque service que vous prévoyez d'utiliser. Selon le service, vous devrez transmettre un nom d'utilisateur et un mot de passe avec l'authentification de base, ou vous devrez transmettre une clé d'API dans un paramètre pour chaque demande que vous effectuez.

Comment obtenir des informations d'identification pour un service Watson:

1. [Inscrivez-vous à Bluemix](#) et connectez-vous.
2. Accédez à la page de service correspondant au service Watson souhaité:
 - [AlchemyLanguage et AlchemyData News](#)
 - [Conversation](#)
 - [Dialogue](#)
 - [Conversion de documents](#)
 - [La traduction de la langue](#)
 - [Classificateur de langue naturelle](#)
 - [Perspectives de personnalité](#)
 - [Récupérer et classer](#)
 - [Discours à texte](#)
 - [Texte pour parler](#)
 - [Analyseur de tonalité](#)
 - [Analyse des compromis](#)
 - [Reconnaissance visuelle](#)
3. Sélectionnez le plan souhaité et cliquez sur CREER:



Conversation

IBM

PUBLISH DATE
08/15/2016

AUTHOR
IBM

TYPE
Service

LOCATION
US South

[VIEW DOCS](#)

Add a natural language interface to your application to automate interactions with your applications include virtual agents and chat bots that can integrate and communicate with any device. Train Watson Conversation service through an easy-to-use web application to quickly build natural conversation flows between your apps and users, and deploy your solutions.



Pick a plan

Monthly prices shown are for

Plan	Features	
✓ Free	1000 API queries per month Up to 3 Workspaces Up to 25 Intents Shared Public Cloud	Free
Free		
Standard	Unlimited API queries per month Up to 20 Workspaces Up to 2000 Intents Shared Public Cloud	\$0.00

4. Cliquez sur le bouton "Informations d'identification du service" de la page de votre tableau de bord de service pour afficher vos informations d'identification. Si vous n'êtes pas amené au tableau de bord de service automatiquement, accédez à votre [tableau de bord Bluemix](#) et cliquez sur l'instance de service de votre choix.

The screenshot shows the IBM Bluemix console interface. At the top, there's a navigation bar with the Bluemix logo and text: "IBM Bluemix Ready? Try the new Bluemix | New! Try OpenWhisk". Below this is a dark sidebar with a "Back to Dashboard..." button and a "Conversation-rw" header. The sidebar contains a "Manage" section with "Service Credentials" highlighted in a red box, and "Service Access Authorization". Below that is a section titled "APPS USING SERVICE". The main content area is titled "Service Credentials" and contains a table with one entry: "NAME Credentials-1". Below the table, there's a section titled "SERVICE CREDENTIALS" which displays a JSON object in a dark-themed code editor. The JSON object is:

```
{  "credentials": {    "url": "https://gateway.watsonplatform.net",    "password": "XXXXXXXXXX",    "username": "XXXXXXXXXX"  }}
```

Appel des API Watson avec curl

En fonction du service, vous devrez soit utiliser l'authentification de base avec un `username` et un `password` soit passer un `apikey` tant que paramètre dans chaque demande.

Certains services prennent également en charge l' [authentification par jeton](#) .

GET using Tone Analyzer:

```
curl -X GET \  
-u "username":"password" \  
-d "version=2016-05-19" \  
-d "text=Hey! Welcome to Watson Tone Analyzer!" \  
"https://gateway.watsonplatform.net/tone-analyzer/api/v3/tone"
```


POST utilisant AlchemyLanguage:

```
curl -X POST \  
-d "apikey=YOUR_API_KEY" \  
-d "url=www.ibm.com" \  
"https://gateway-a.watsonplatform.net/calls/url/URLGetRankedKeywords"
```

Utilisation des SDK Watson Developer Cloud

Le moyen le plus rapide de commencer avec les services Watson consiste à utiliser les SDK Watson Developer Cloud. Les référentiels GitHub suivants contiennent des instructions d'installation et des exemples d'utilisation de base:

- [Android](#)
- [iOS](#)
- [Java](#)
- [Node.js](#)
- [Python](#)
- [Unité](#)

Par exemple, voici comment créer un appel d'API AlchemyLanguage avec le SDK Node.js:

Installez le SDK:

```
$ npm install watson-developer-cloud
```

Enregistrez le code suivant dans un fichier (nous l'appellerons *app.js*). Assurez-vous de remplacer `API_KEY` par votre clé API.

```
// Instantiate the service  
var AlchemyLanguageV1= require('watson-developer-cloud/alchemy-language/v1');  
var alchemy_language = AlchemyLanguageV1({  
  api_key: 'API_KEY'  
})  
  
var parameters = {  
  extract: [  
    'entities',  
    'keywords'  
  ]  
  url: 'https://www.ibm.com/us-en/'  
};  
  
alchemy_language.combined(parameters, function (err, response) {  
  if (err)  
    console.log('error:', err);  
  else  
    console.log(JSON.stringify(response, null, 2));  
});
```

Exécutez l'application:

```
$ node app.js
```

Lire Démarrer avec ibm-watson-cognitive en ligne: <https://riptutorial.com/fr/ibm-watson-cognitive/topic/607/demarrer-avec-ibm-watson-cognitive>

Chapitre 2: AlchemyLanguage

Remarques

AlchemyLanguage est un ensemble de méthodes d'analyse de texte qui permettent de mieux comprendre votre contenu texte ou HTML. Consultez la rubrique [Démarrage](#) pour apprendre à utiliser AlchemyLanguage et d'autres services Watson. Pour plus de détails sur AlchemyLanguage et des exemples, consultez la [référence](#) et la [documentation de l' API](#) .

Limites de taille

- Contenu HTML avant le nettoyage du texte: **600 Ko**
- Texte source, après nettoyage du texte: **50 Ko**
- Appels utilisant des modèles personnalisés: **5 Ko**

Support linguistique

Pour voir quelles langues sont prises en charge pour chaque fonction, reportez-vous à l'entrée de chaque fonction dans la [référence de l' API](#) .

Détection de langue

Par défaut, AlchemyLanguage détecte automatiquement la langue de votre texte source. Vous pouvez spécifier manuellement la langue de votre contenu avec le paramètre de requête de `language` . (par exemple `language=spanish`)

Nettoyage de texte

Lorsque vous utilisez une fonction HTML ou URL de l'API, AlchemyLanguage nettoie le contenu pour préparer le texte source pour l'analyse. Le paramètre `sourceText` vous permet de personnaliser le processus de nettoyage avec les options suivantes:

- `cleaned_or_raw` (default) - Supprime les éléments du site tels que les liens, les annonces, etc. Si le nettoyage échoue, le texte brut de la page Web est utilisé
- `cleaned` - Supprime les éléments du site tels que les liens, les annonces, etc.
- `raw` - Utilise du texte brut de page Web sans nettoyage
- `cquery` - Utilise la requête de contraintes visuelles que vous spécifiez dans le paramètre `cquery` . Voir la [documentation](#) pour plus de détails sur les requêtes de contraintes visuelles.
- `xpath` - Utilise la requête XPath que vous spécifiez dans le paramètre `xpath`
- `xpath_or_raw` - Utilise les résultats d'une requête XPath pour revenir au texte brut si la

requête XPath ne renvoie rien

- `cleaned_and_xpath` - Utilise les résultats d'une requête XPath sur du texte de page Web nettoyé

Exemples

Appel combiné: utilisez plusieurs fonctions dans un seul appel API (Node.js)

La méthode d'appel combiné vous permet d'utiliser plusieurs fonctions `AlchemyLanguage` dans une requête. Cet exemple utilise un appel combiné pour extraire des entités et des mots-clés du site Web IBM et renvoie des informations de sentiment pour chaque résultat.

Cet exemple nécessite [les informations d'identification du service AlchemyLanguage](#) et [Node.js](#).

1. Utilisez une interface de ligne de commande pour installer le [SDK Watson Developer Cloud Node.js](#) :

```
$ npm install watson-developer-cloud
```

2. Enregistrez le code suivant dans un fichier `app.js` dans le même répertoire. Assurez-vous de remplacer `API_KEY` par votre clé `AlchemyAPI`:

```
var AlchemyLanguageV1 = require('watson-developer-cloud/alchemy-language/v1');
var alchemy_language = AlchemyLanguageV1({
  api_key: 'API_KEY'
});

var parameters = {
  extract: 'entities,keywords',
  sentiment: 1,
  url: 'https://www.ibm.com/us-en/'
};

alchemy_language.combined(parameters, function (err, response) {
  if (err)
    console.log('error:', err);
  else
    console.log(JSON.stringify(response, null, 2));
});
```

3. Exécutez l'application:

```
$ node app.js
```

Analyse de sentiment: obtenir des informations de sentiment pour des phrases spécifiques dans le texte (Node.js)

La fonctionnalité «Sentiment ciblé» d'`AlchemyLanguage` peut rechercher des phrases cibles dans votre contenu et renvoyer des informations de sentiment pour chaque résultat.

Cet exemple nécessite [les informations d'identification du service AlchemyLanguage](#) et [Node.js](#)

1. Utilisez une interface de ligne de commande pour installer le [SDK Watson Developer Cloud](#) [Node.js](#) :

```
$ npm install watson-developer-cloud
```

2. Enregistrez le code suivant dans un fichier *app.js* dans le même répertoire. Assurez-vous de remplacer `API_KEY` par votre clé AlchemyAPI:

```
var AlchemyLanguageV1 = require('watson-developer-cloud/alchemy-language/v1');
var alchemy_language = new AlchemyLanguageV1({
  api_key: 'API_KEY'
})

var parameters = {
  text: 'Grapes are the best! I hate peaches.',
  targets: [
    'grapes',
    'peaches'
  ]
};

alchemy_language.sentiment(parameters, function (err, response) {
  if (err)
    console.log('error:', err);
  else
    console.log(JSON.stringify(response, null, 2));
});
```

3. Exécutez l'application:

```
$ node app.js
```

Concepts: identifier des concepts à partir d'une page Web (Node.js)

AlchemyLanguage peut détecter des concepts généraux référencés dans votre contenu. Le service renvoie [les liens de données liées](#) pour chaque concept et une URL vers un site Web pertinent lorsque cela est possible.

Cet exemple nécessite [les informations d'identification du service AlchemyLanguage](#) et [Node.js](#)

1. Utilisez une interface de ligne de commande pour installer le [SDK Watson Developer Cloud](#) [Node.js](#) :

```
$ npm install watson-developer-cloud
```

2. Enregistrez le code suivant dans un fichier *app.js* dans le même répertoire. Assurez-vous de remplacer `API_KEY` par votre clé AlchemyAPI.

```
var AlchemyLanguageV1 = require('watson-developer-cloud/alchemy-language/v1');
var alchemy_language = new AlchemyLanguageV1({
  api_key: 'API_KEY'
})

var parameters = {
  url: 'http://www.cnn.com'
};

alchemy_language.concepts(parameters, function (err, response) {
  if (err)
    console.log('error:', err);
  else
    console.log(JSON.stringify(response, null, 2));
});
```

3. Exécutez l'application:

```
$ node app.js
```

Lire AlchemyLanguage en ligne: <https://riptutorial.com/fr/ibm-watson-cognitive/topic/6817/alchemylanguage>

Chapitre 3: Discours à texte

Remarques

IBM Watson Speech to Text offre une variété d'options pour la transcription audio en plusieurs langues et formats:

- **WebSockets** - établissez une connexion persistante sur le protocole WebSocket pour une transcription continue
- **Sans session** - transcrire l'audio sans avoir à créer et à maintenir une session
- **Sessions** - créez de longs échanges multi-tours avec le service ou établissez plusieurs conversations parallèles avec une instance particulière du service
- **Asynchrone** - fournit une interface HTTP non bloquante pour la transcription audio. Vous pouvez enregistrer une URL de rappel pour être informé de l'état du travail et des résultats, ou vous pouvez interroger le service pour connaître l'état du travail et récupérer les résultats manuellement.

Reportez-vous à la rubrique [Mise en route](#) pour apprendre à utiliser Speech to Text et d'autres services Watson. Pour plus de détails sur la parole en texte et des exemples, consultez la [référence de l' API](#) et la [documentation](#) .

Exemples

Reconnaître un fichier audio en utilisant WebSockets en Java

Utilisation du [Java-SDK 3.0.1](#)

```
CountDownLatch lock = new CountDownLatch(1);

SpeechToText service = new SpeechToText();
service.setUsernameAndPassword("<username>", "<password>");

FileInputStream audio = new FileInputStream("filename.wav");

RecognizeOptions options = new RecognizeOptions.Builder()
    .continuous(true)
    .interimResults(true)
    .contentType(HttpMediaType.AUDIO_WAV)
    .build();

service.recognizeUsingWebSocket(audio, options, new BaseRecognizeCallback() {
    @Override
    public void onTranscription(SpeechResults speechResults) {
        System.out.println(speechResults);
        if (speechResults.isFinal())
            lock.countDown();
    }
})
```

```
});  
  
lock.await(1, TimeUnit.MINUTES);
```

Transcrire un fichier audio à l'aide de WebSockets (Node.js)

Cet exemple montre comment utiliser le service IBM Watson Speech to Text pour reconnaître le type d'un fichier audio et produire une transcription du texte prononcé dans ce fichier.

Cet exemple nécessite [des informations d'identification de service Speech to Text](#) et [Node.js](#)

1. Installez le module npm pour le [SDK Watson Developer Cloud Node.js](#) :

```
$ npm install watson-developer-cloud
```

2. Créez un fichier JavaScript (par exemple, *app.js*) et copiez-y le code suivant. Assurez-vous de saisir le `username` et le `password` de votre instance de service Speech to Text.

```
var SpeechToTextV1 = require('watson-developer-cloud/speech-to-text/v1');  
var fs = require('fs');  
  
var speech_to_text = new SpeechToTextV1({  
  username: 'INSERT YOUR USERNAME FOR THE SERVICE HERE',  
  password: 'INSERT YOUR PASSWORD FOR THE SERVICE HERE',  
  url: 'https://stream.watsonplatform.net/speech-to-text/api'  
});  
  
var params = {  
  content_type: 'audio/flac'  
};  
  
// Create the stream,  
var recognizeStream = speech_to_text.createRecognizeStream(params);  
  
// pipe in some audio,  
fs.createReadStream('0001.flac').pipe(recognizeStream);  
  
// and pipe out the transcription.  
recognizeStream.pipe(fs.createWriteStream('transcription.txt'));  
  
// To get strings instead of Buffers from received `data` events:  
recognizeStream.setEncoding('utf8');  
  
// Listen for 'data' events for just the final text.  
// Listen for 'results' events to get the raw JSON with interim results, timings, etc.  
['data', 'results', 'error', 'connection-close'].forEach(function(eventName) {  
  recognizeStream.on(eventName, console.log.bind(console, eventName + ' event: '));  
});
```

3. Enregistrez le fichier audio exemple *0001.flac* dans le même répertoire. Cet exemple de code est configuré pour traiter les fichiers **FLAC**, mais vous pouvez modifier la section `params` du code exemple pour obtenir des transcriptions à partir de fichiers audio dans d'autres formats. Les formats pris en charge incluent **WAV** (type `audio/wav`), **OGG** (type `audio/ogg`)

) et autres. Voir la [référence de l'API Speech to Text](#) pour une liste complète.

4. Exécutez l'application (utilisez le nom du fichier contenant le code exemple)

```
$ node app.js
```

Après avoir exécuté l'application, vous trouverez le texte transcrit de votre fichier audio dans le fichier *transcription.txt* du répertoire à partir duquel vous avez exécuté l'application.

Lire Discours à texte en ligne: <https://riptutorial.com/fr/ibm-watson-cognitive/topic/675/discours-a-texte>

Chapitre 4: Reconnaissance visuelle

Exemples

Obtenir une liste de classificateurs personnalisés

Cela répertorie tous les classificateurs personnalisés que vous avez formés.

```
'use strict';

let watson = require('watson-developer-cloud');

var visualRecognition = watson.visual_recognition({
  version: 'v3',
  api_key: process.env['API_KEY'],
  version_date: '2016-05-19'
});

let url =
'https://upload.wikimedia.org/wikipedia/commons/1/1c/Chris_Evans_filming_Captain_America_in_DC_cropped

visualRecognition.classify({url: url}, function(error, results) {
  console.log(JSON.stringify(results, null, 2));
});
```

Obtenir des informations sur un classificateur personnalisé spécifique

Cela renvoie des informations sur un ID de classificateur spécifique que vous avez formé. Cela inclut des informations sur son statut actuel (c.-à-d. S'il est prêt ou non).

```
'use strict';

let watson = require('watson-developer-cloud');

var visualRecognition = watson.visual_recognition({
  version: 'v3',
  api_key: process.env.API_KEY,
  version_date: '2016-05-19'
});

visualRecognition.getClassifier({classifier_id: 'DogBreeds_1162972348'}, function(error,
results) {
  console.log(JSON.stringify(results, null, 2));
});
```

Former un classificateur personnalisé

Former un classificateur personnalisé nécessite un corpus d'images organisé en groupes. Dans cet exemple, j'ai un tas d'images de pommes dans un fichier ZIP, un groupe d'images de bananes

dans un autre fichier ZIP et un troisième groupe d'images de choses qui *ne sont pas des fruits* pour un *jeu négatif*. Une fois qu'un classificateur personnalisé est créé, il sera en état de `training` et vous devrez utiliser l'ID de classificateur pour vérifier si elle est prêt (en utilisant le « Obtenir des informations sur un classificateur spécifique personnalisé » exemple).

```
'use strict';

let watson = require('watson-developer-cloud');
let fs = require('fs');

var visualRecognition = watson.visual_recognition({
  version: 'v3',
  api_key: process.env.API_KEY,
  version_date: '2016-05-19'
});

let custom_classifier = {
  apple_positive_examples: fs.createReadStream('./apples.zip'),
  banana_positive_examples: fs.createReadStream('./bananas.zip'),
  negative_examples: fs.createReadStream('./non-fruits.zip'),
  name: 'The Name of My Classifier'
}

visualRecognition.createClassifer(custom_classifier, function(error, results) {
  console.log(JSON.stringify(results, null, 2));
});
```

Supprimer un classificateur personnalisé

```
'use strict';

let watson = require('watson-developer-cloud');
let fs = require('fs');

var visualRecognition = watson.visual_recognition({
  version: 'v3',
  api_key: process.env.API_KEY,
  version_date: '2016-05-19'
});

let classifier_id_to_delete = 'TheNameofMyClassifier_485506080';

visualRecognition.deleteClassifier({classifier_id: classifier_id_to_delete}, function(error, results) {
  console.log(JSON.stringify(results, null, 2));
});
```

Classer une image

Conditions préalables

Tout d'abord, vous devez installer le SDK `watson-developer-cloud`.

```
$ npm install watson-developer-cloud
```

Classer une URL d'image



Nous utiliserons une image de Captain America de Wikipedia.

```
'use strict';

let watson = require('watson-developer-cloud');

var visualRecognition = watson.visual_recognition({
  version: 'v3',
  api_key: "<YOUR API KEY GOES HERE>",
  version_date: '2016-05-19'
});

let url =
"https://upload.wikimedia.org/wikipedia/commons/1/1c/Chris_Evans_filming_Captain_America_in_DC_cropped";

visualRecognition.classify({url: url}, function(error, results) {
  console.log(JSON.stringify(results, null, 2));
});
```

Lire Reconnaissance visuelle en ligne: <https://riptutorial.com/fr/ibm-watson->

Chapitre 5: Récupérer et classer

Remarques

Le client Solrj et le SDK Java sont indépendants, vous pouvez donc les mettre à jour individuellement. Assurez-vous toujours d'utiliser la dernière version du SDK Java.

Voir la page de publication de GitHub pour les mises à jour <https://github.com/watson-developer-cloud/java-sdk/releases>

Exemples

Recherche et classement à l'aide de la fonction Récupérer et classer en Java

Installez les dépendances requises:

```
'org.apache.solr:solr-solrj:5.5.1'  
'org.apache.httpcomponents:httpclient:4.3.6'  
'com.ibm.watson.developer_cloud:java-sdk:3.2.0'
```

Le code ci-dessous suppose que vous avez une collection Solr avec des documents et que vous avez formé un classeur, sinon suivez ce [tutoriel](#)

```
public class RetrieveAndRankSolrJExample {  
  
    private static HttpSolrClient solrClient;  
    private static RetrieveAndRank service;  
  
    private static String USERNAME = "<username>";  
    private static String PASSWORD = "<password>";  
    private static String SOLR_CLUSTER_ID = "<your-solr-cluster-id>";  
    private static String SOLR_COLLECTION_NAME = "<your-collection-name>";  
    private static String RANKER_ID = "<ranker-id>";  
  
    public static void main(String[] args) throws SolrServerException, IOException {  
  
        // create the retrieve and rank instance  
        service = new RetrieveAndRank();  
        service.setUsernameAndPassword(USERNAME, PASSWORD);  
  
        // create the solr client  
        String solrUrl = service.getSolrUrl(SOLR_CLUSTER_ID);  
        solrClient = new HttpSolrClient(solrUrl, createHttpClient(solrUrl, USERNAME, PASSWORD));  
  
        // build the query  
        SolrQuery query = new SolrQuery("*:*");  
        query.setRequestHandler("/fcselect");  
        query.set("ranker_id", RANKER_ID);  
  
        // execute the query  
        QueryResponse response = solrClient.query(SOLR_COLLECTION_NAME, query);  
        System.out.println("Found " + response.getResults().size() + " documents!");  
    }  
}
```

```

    System.out.println(response);
}

private static HttpClient createHttpClient(String uri, String username, String password) {
    final URI scopeUri = URI.create(uri);

    final BasicCredentialsProvider credentialsProvider = new BasicCredentialsProvider();
    credentialsProvider.setCredentials(new AuthScope(scopeUri.getHost(), scopeUri.getPort()),
        new UsernamePasswordCredentials(username, password));

    final HttpClientBuilder builder = HttpClientBuilder.create()
        .setMaxConnTotal(128)
        .setMaxConnPerRoute(32)

    .setDefaultRequestConfig(RequestConfig.copy(RequestConfig.DEFAULT).setRedirectsEnabled(true).build())

        .setDefaultCredentialsProvider(credentialsProvider)
        .addInterceptorFirst(new PreemptiveAuthInterceptor());
    return builder.build();
}

private static class PreemptiveAuthInterceptor implements HttpRequestInterceptor {
    public void process(final HttpRequest request, final HttpContext context) throws
HttpException {
        final AuthState authState = (AuthState)
context.getAttribute(HttpClientContext.TARGET_AUTH_STATE);

        if (authState.getAuthScheme() == null) {
            final CredentialsProvider credsProvider = (CredentialsProvider) context
                .getAttribute(HttpClientContext.CREDS_PROVIDER);
            final HttpHost targetHost = (HttpHost)
context.getAttribute(HttpCoreContext.HTTP_TARGET_HOST);
            final Credentials creds = credsProvider.getCredentials(new
AuthScope(targetHost.getHostName(),
                targetHost.getPort()));
            if (creds == null) {
                throw new HttpException("No creds provided for preemptive auth.");
            }
            authState.update(new BasicScheme(), creds);
        }
    }
}
}
}
}

```

Lire Récupérer et classer en ligne: <https://riptutorial.com/fr/ibm-watson-cognitive/topic/6053/recuperer-et-classer>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec ibm-watson-cognitive	Community , Garrett M , German Attanasio
2	AlchemyLanguage	Garrett M
3	Discours à texte	Garrett M , German Attanasio , seh , WvH
4	Reconnaissance visuelle	German Attanasio , Joshua Smith , seh
5	Récupérer et classer	German Attanasio