

Vous avez 8 heures pour concevoir les pièces manquantes d'une application permettant aux clients d'un magasin de détail (vins et spiritueux) d'obtenir le produit qui leur convient. Vous serez évalués sur votre talent à produire des solutions performantes et fiables et pour votre créativité à répondre aux besoins de nos capricieux utilisateurs.

Vous aurez à votre disposition

- Cette documentation
- Les judicieux conseils de nos trois clients et leur fiche descriptive
- Deux environnements pour tester vos solutions:
 - Un simplifié pour développer votre solution
 - Un complet pour l'évaluation
- Réseau et connexion internet (svp soyez modéré avec votre utilisation d'Internet)

Vous devrez développer ces trois pièces manquantes

1. **Un extracteur de données** qui vous permettra d'indexer l'information nécessaire pour votre application.
2. **Un service de suggestions et de corrections de requêtes** que vous pourrez utiliser dans votre application pour faciliter la recherche.
3. **Une interface utilisateur** qui permettra à nos clients de rapidement trouver le produit qui leur convient.

Pour gagner vous devrez

Cumuler un maximum de points. Vous pouvez réaliser ceci de trois façons, soit :

1. En tout temps, en soumettant votre solution d'extraction ou de suggestion de requêtes à l'évaluation
2. En présentant une maquette de votre application 1½ heure après le début du défi
3. En soumettant votre application aux juges à la fin du défi

Vous pouvez faire évaluer votre solution d'extraction ou de suggestion à tout moment. Les points ne sont pas cumulatifs, seule votre dernière évaluation sera considérée pour la note finale. Les résultats seront affichés en temps réel à l'écran, question de nourrir votre esprit de compétition. Les résultats du problème interface utilisateur seront présentés à la fin du défi. Le total des points est partagé également entre les trois problèmes.

Quelques règles pour une saine compétition

- L'utilisation de bibliothèques externes doit être autorisée par Coveo.
- Toute utilisation de code produit préalablement au début du défi doit être autorisée par Coveo.
- Pour le problème d'extraction, l'information poussée à l'index ne peut provenir de données persistées sur votre machine. Les données doivent provenir de notre service de données.
- Toute action portant à nuire à une autre équipe est interdite.
- Coveo se garde le droit de regard sur le code source des solutions pour s'assurer que les règles ont été respectées.
- Coveo se garde un droit de regard sur le contenu de l'index utilisé par l'application et pourrait réviser le pointage si le contenu comporte des irrégularités.
- Le non-respect des règles peut mener à l'élimination de l'équipe.
- Un test antidopage sera réalisé à la fin du défi pour s'assurer que le code réalisé est conforme aux règles. :P

Amusez-vous, soyez compétitifs, mais pas antisportifs !



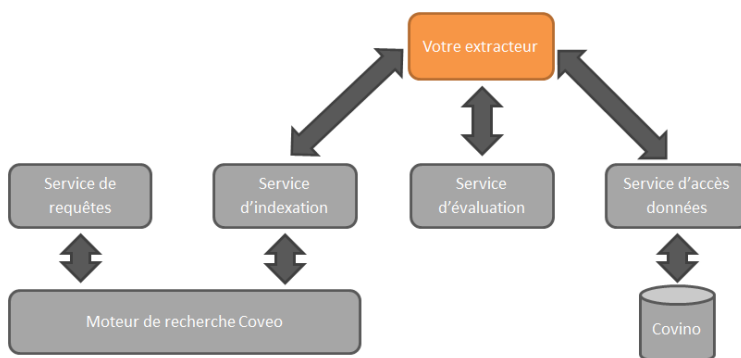
Déroulement du défi

- 7h00 Arrivée des participants et rencontre avec Coveo
- 8h00 Présentation du défi
- 8h30 Début de la compétition
- 10h00 Évaluation des croquis
- 12h00 Dîner
- 16h30 Fermeture des services d'évaluation
Votre dernier résultat est celui qui sera utilisé pour le pointage final, soyez prudents !
- 16h45 Visite libre, présentation des applications et évaluation finale des juges
- 17h00 5 @ 7
- 20h00 Présentation et remise des prix



Extraction des données

La connexion à une source de données est la première étape dans l'implantation d'un engin de recherche. Ce problème fera appel à votre capacité à développer des solutions robustes et performantes. Vous devrez faire plusieurs compromis entre la quantité d'information extraite utile à votre interface utilisateur et la performance de votre extracteur. L'ensemble des données doit être transféré vers un index central fourni par l'équipe Coveo et qui sera réutilisé par votre interface utilisateur.



Les critères évalués

- Le nombre de bouteilles extraites de Covino
- Le temps total de l'extraction

L'évaluation

1. Obtenez votre code d'évaluation de l'administrateur Coveo [<runid>](#) et [<tokenid>](#).
2. Débutez une session avec:
<http://<crawlsvr>:8123/blitzservice/start?team=<name>&run=<runid>&token=<tokenid>>.
Aucun autre service n'est disponible tant que [/start](#) n'a pas été appelé.
3. Réalisez l'extraction et l'indexation. Attention, le serveur est lent. Chaque appel prend au moins 500 ms peut-être plus! Il est donc grandement conseillé de paralléliser les appels.
4. Terminez votre session avec (le chrono s'arrête seulement après cet appel):
<http://<crawlsvr>:8123/blitzservice/end?team=<name>&run=<runid>&token=<tokenid>>
5. Obtenez votre résultat. Le détail est disponible à cette adresse:
<http://<monitorsvr>/teamdetails.aspx?team=<teamname>>

Pour débiter

Connectez-vous à l'API REST de Covino. Le seul service qui est documenté est [/masterservice](#). Tous les services possèdent la méthode [/def](#) qui retourne la liste des méthodes valides pour un service donné. Vous pouvez donc faire une requête auprès du [/masterservice](#) pour obtenir cette liste via <http://<crawlsvr>:8123/masterservice/def>.

API Covino

L'API de Covino est un API REST. Il est séparé en de nombreux services contenant différentes informations. Vous devrez utiliser des requêtes HTTP de type GET pour pouvoir extraire les informations contenues dans le système. Voici un exemple de requête:

[GET http://<crawlsrv>:8123/masterservice/def](http://<crawlsrv>:8123/masterservice/def)

La commande GET est utilisée pour tous les appels de Covino. Les paramètres sont tous passés via l'URL demandé. Voici un exemple de réponse:

```
HTTP/1.1 200 OK
Content-Length: 62
Content-Type: application/json; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
Date: Fri, 16 Dec 2011 22:21:59 GMT

[{"Url":"/def","Method":"GET"}, {"Url":"/list","Method":"GET"}]
```

La réponse contient seulement deux informations pertinentes : le code de statut et le corps du message. Le code de statut vous indique si la réponse est un succès (200) ou une erreur (404, 500, ...). La réponse contient l'objet demandé sous format JSON. Les services peuvent répondre à plusieurs requêtes simultanément. Pour une question de performance, il est donc grandement conseillé de paralléliser les tâches d'extraction.

Dans le cadre de l'évaluation, les services de Covino doivent être appelés avec la query string [?team=<name>&run=<rundi>&token=<tokenid>](#) pour tous les appels. Tout appel utilisant une session invalide sera refusé. Une session devient valide lorsque [blitzservice/start](#) est appelé et devient invalide lorsque [blitzservice/end](#) est appelé. Si vous avez des questions spécifiques techniques quant à la communication avec Covino, n'hésitez pas à demander aux gens de Coveo.



API d'indexation

Pour être en mesure d'accéder au contenu via le moteur de recherche Coveo, vous devez ajouter les documents à l'index via l'API d'indexation. Voici en trois étapes comment faire:

1. Préparer votre index à recevoir des documents
[GET http://<indexsvr>:8124/Index/Clear?Team=<team>](http://<indexsvr>:8124/Index/Clear?Team=<team>)
Détruit tous les documents de votre index et le prépare pour une nouvelle indexation.
2. Ajouter vos documents à votre index
[POST http://<indexsvr>:8124/Index?Team=<team>](http://<indexsvr>:8124/Index?Team=<team>)
Le corps du message doit être présenté sous le format JSON et respecter la structure suivante:

```
{
  "Id": "nn",
  "Name": "nom du vin",
  "Info": "texte ou html",
  "Fields": [
    {
      "Name": "field name", // utiliser BlitzFieldxx où xx est entre 01 et 50
      "Value": "simple or complex"
    }
  ] // vous pouvez avoir jusqu'à 50 fields
}
```

3. Terminer l'indexation
[GET http://<indexsvr>:8124/Index/Commit?Team=<team>](http://<indexsvr>:8124/Index/Commit?Team=<team>)
Informe le serveur que l'indexation est complète et rend les documents disponibles pour les recherches.

Exemple d'un appel pour indexer un document

Message envoyé:

POST http://Blitz01:8124/Index?Team=blitz
Content-Type: application/json; charset=utf-8
Host: Blitz01:8124
Content-Length: 115
Expect: 100-continue

```
{"Id":0,"Name":"Cabernet Ridge Monte Bello  
2008","Info":null,"Fields":[{"Name":"BlitzField01","Value":"Bordeaux"}]}
```

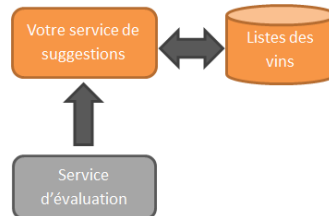
Réponse:

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
Nancy-Version: 0.9.0.0
Date: Fri, 16 Dec 2011 22:18:20 GMT
"OK"



Suggestion et correction de requêtes

Cette fonctionnalité assiste l'utilisateur en lui suggérant des mots à utiliser lors de la rédaction de sa requête. Il permet également de corriger l'orthographe de certains mots et empêche donc l'utilisateur de partir sur une fausse piste. Pour votre application, vous devrez donc fournir un service capable de proposer des suggestions et corrections à des requêtes utilisateur. Dans le contexte de Covino, vous aurez à votre disposition la liste des noms de tous les vins contenus dans notre base de données.



Vous devrez retourner des suggestions au mot à compléter (seulement et non tous les termes de la requête). Vous devrez supporter de nous retourner une liste de suggestions de taille variable. Si le dernier mot de la requête est complet, vous devez nous proposer le mot suivant. Pour vous éclairer, voici quelques exemples que vous devriez supporter :

Requête : "cabernet"	Suggestions : cabernet, cabernetkoonunga, cabernets
Requête : "é"	Suggestions : émilien, érable, éden, édition, église
Requête : "e"	Suggestions : emilien, estate, eastern, estèphe
Requête : "cabernet f"	Suggestions : franc, fuzion, wakefield
Requête : "cabernet "	Suggestions : sauvignon, merlot, shiraz
Requête : "moutn"	Suggestions : mouton, mount, mout, moulin, south, mont

Les critères évalués

- La capacité à donner des suggestions pertinentes pour l'utilisateur
- La capacité à détecter des erreurs de frappe et donner des suggestions en conséquence

L'évaluation

1. Dès le début du défi, donnez l'adresse IP de la machine qui hébergera le service à notre administrateur Coveo
2. Débutez une session en le demandant verbalement à l'administrateur Coveo
3. Votre service de suggestion sera appelé par notre service d'évaluation (les requêtes changeront dans le temps)
4. Obtenez votre résultat. Le détail sera disponible à cette adresse
<http://<monitorsvr>/teamdetails.aspx?team=<teamname>>



Pour débiter

Obtenez les listes des vins en utilisant l'API de Covino <http://<testsvr>:8123/BottleService/names>. Pour cette liste seulement et pour le problème de suggestion seulement, il vous est permis de réutiliser la liste sauvegardée sur disque.



API de suggestion

Vous devez fournir un web service pour le problème de suggestion. Celui-ci sera appelé par notre service d'évaluation pour vous donner un résultat. Il devra supporter la syntaxe suivante:

[GET http://<host>:8125/Suggest?Query=<query>&MaxNb=<Count>](http://<host>:8125/Suggest?Query=<query>&MaxNb=<Count>)

[<Query>](#): C'est la requête sur laquelle vous devez faire des suggestions

[<Count>](#): C'est le nombre de suggestions que l'on veut en retour

Le résultat attendu est une liste (array) de mots en format JSON.

Exemple d'un appel de suggestion

Message envoyé:

GET http://Blitz01:8127/Suggest?Query=shirez&MaxNb=10 HTTP/1.1

Content-Type: application/json; charset=utf-8

Host: Blitz01:8124

Content-Length: 115

Expect: 100-continue

Réponse:

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

Server: Microsoft-HTTPAPI/2.0

Nancy-Version: 0.9.0.0

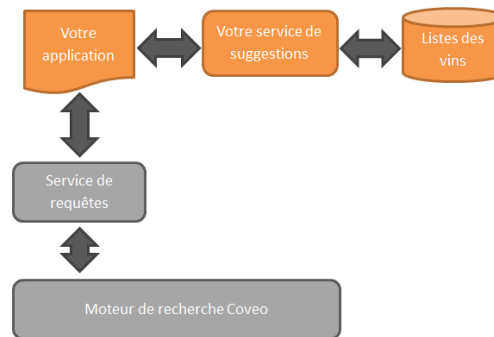
Date: Fri, 16 Dec 2011 22:30:55 GMT

["shiraz","shore","shine","shirah","share","spiritueux","chorey","poire","white","spinetta"]



Interface utilisateur

La finalité de ce concours est évidemment de faciliter la vie de nos clients. Ceux-ci requièrent une interface utilisateur qui leur permet de trouver le produit qui leur convient le mieux. Votre application devra mettre à profit l'information que vous avez extraite de Covino et stockée dans l'index. Vous pouvez également mettre votre service de suggestions à l'œuvre. N'hésitez pas à poser des questions à nos trois clients typiques. Ils seront également vos juges pour l'évaluation finale. Vraiment, rien ne vaut une bonne discussion pour bien comprendre ce que les gens recherchent.



Les critères évalués

- Le respect des besoins de nos clients
- La qualité du design
- L'ergonomie et l'expérience utilisateur
- La créativité du design

L'évaluation

- 1½ heure après le début du défi, vous devrez fournir des maquettes et une présentation de votre design. Vous aurez 5 minutes pour nous vendre votre solution, pas plus.
- À la fin du défi, vous serez évalués par trois juges. Vous aurez 10 minutes pour leur démontrer la force de votre application. Soyez prêts dès leur arrivée.

Bien que 1½ heure soit prévue pour la réalisation des croquis, il est fortement conseillé de commencer la réalisation de l'interface utilisateur dès le début du défi.



API de recherche

Pour accéder aux données que vous avez indexées lors de l'extraction, vous devez utiliser l'interface de recherche hébergée par le moteur de recherche de Coveo. Cette interface vous fournit deux méthodes soit:

1 - Exécuter une recherche

Cette méthode est votre principal moyen d'obtenir l'information que vous allez afficher sur votre portail. Sa syntaxe est la suivante: [POST http://<indexsvr>:8124/Search?Team=<team>](http://<indexsvr>:8124/Search?Team=<team>). Le corps du message doit être présenté sous le format JSON et respecter la structure suivante:

```
{
  "Expression": "le texte de la requête",
  "FirstResult": nn,
  "NbResults": nn,
  "SortByAscending": "BlitzFieldnn",
  "SortByDescending": "BlitzFieldnn",
  "Filters": [
    {
      "Name": "le nom du field",
      "Value": "valeur désirée"
    }
  ]
}
```

Expression: Le texte de la requête (les wildcards sont acceptées '*' et '?')

FirstResult: Le premier résultat à retourner (pour la pagination)(0-based)

SortByAscending: À spécifier si on désire des résultats triés selon ce champ.

SortByDescending: À spécifier si on désire des résultats triés selon ce champ, mais en ordre décroissant. À noter que SortByAscending et SortByDescending sont exclusifs.

Filters: Les filtres qui doivent être appliqués sur la requête. C'est une liste nom\ valeur qui peut contenir plusieurs éléments.

Name: Le nom du champ dans l'index (BlitzFieldxx)

Value: La valeur que doit avoir le champ pour que le document soit retourné. Si la valeur est vide, tous les documents dont le champ est défini seront retournés.

L'objet retourné par la requête est sous la forme suivante:

```
{
  "TotalCount": nn,
  "wines": [
    {
      "Id": nn,
      "Name": "wine name",
      "Info": "text body",
      "Fields": [
        {
          "Name": "field name1",
          "Value": "field value1"
        },
        ...
      ]
    },
    ...
  ]
}
```

TotalCount: Le nombre d'éléments satisfaisant la requête. Le nombre retourné ne dépassera pas NbResults.

wines: La liste de tous les documents retournés par la requête. Cette liste contient des éléments tels que d'écrit dans l'API d'indexation.



2 – Obtenir la liste des valeurs possibles pour un champ

Cette méthode vous permet d'obtenir la liste de toutes les valeurs possibles pour un champ. Avec cette information, il est possible de créer une mécanique de recherche par facettes ou par filtres. Sa syntaxe est la suivante: [GET http://<indexsvr>:8124/Search/ListValues?Team=<team>&Field=<fieldname>](http://<indexsvr>:8124/Search/ListValues?Team=<team>&Field=<fieldname>). La valeur de retour est présentée dans une liste sous la forme suivante:

```
[{"value":"valeur du champ",  
  "NbDocs":nn  
}]  
  
value: La valeur du champ  
NbDocs: Le nombre de documents qui possèdent cette valeur
```

Exemple d'un appel pour une recherche

Message envoyé:

```
POST http://Blitz01:8124/Search?Team=blitz HTTP/1.1  
Content-Type: application/json; charset=utf-8  
Host: Blitz01:8124  
Content-Length: 108  
Expect: 100-continue  
Connection: Keep-Alive
```

```
{"Expression":"cabernet","Filters":[{"Name":"BlitzField01","Value":"bord*"}],"FirstResult":0,"NbResults":12}
```

Réponse:

```
HTTP/1.1 200 OK  
Content-Type: application/json; charset=utf-8  
Server: Microsoft-HTTPAPI/2.0  
Nancy-Version: 0.9.0.0  
Date: Fri, 16 Dec 2011 22:30:55 GMT
```

```
{"TotalCount":2,"Wines":[{"Id":0,"Name":"Cabernet Ridge Monte Bello  
2008","Info":null,"Fields":[{"Name":"blitzfield01","Value":"Bordeaux"}, {"Name":"blitzid","Value":"0"}, {"Name":"blitzteam","Value":"blitz"}]}, {"Id":1,"Name":"Cabernet Riserva Alois Laged Alto Adige  
2007","Info":null,"Fields":[{"Name":"blitzfield01","Value":"Bordeaux"}, {"Name":"blitzid","Value":"1"}, {"Name":"blitzteam","Value":"blitz"}]}]}
```

