

Code challenge : récupération des NOTAM françaises

L'objectif

Dans le cadre de nos offres de stages il est important de pouvoir évaluer les candidats sur des aspects humains et des aspects techniques. Pour la première partie rien ne vaut une rencontre IRL mais pour la partie technique rien ne vaut l'analyse d'un bout de code produit pour résoudre un problème précis et mettant en valeur le niveau de créativité, de logique et de rigueur du candidat à un stage dans nos équipes.

C'est pourquoi nous avons imaginé ce « code challenge » qui nous permettra d'apprécier au mieux votre motivation, votre patte de développeur et votre sens pratique.

Le code produit sera évalué sur les points suivants (par ordre d'importance) :

- Respect des spécifications données
- Clarté du code
- Robustesse du code (gestion des exceptions et logging associé)
- Performance du code
- Éléance de la solution retenue

Bien entendu votre code fera l'objet de discussion lors de notre prochains entretien IRL.

Le contexte

Les NOTAM (Notice To Air Men) sont des messages produits par les services de la navigation aérienne d'un état à destination des utilisateurs de son espace aérien.

En France, ces messages sont publiés sur le site officiel suivant : <http://notamweb.aviation-civile.gouv.fr/>

Le site officiel est plutôt ancien et ne propose aucune API de récupération des données. Pour récupérer les NOTAMs il est donc nécessaire de « crawler » le site pour en extraire les données.

Un formulaire permet de récupérer les NOTAM par FIR (Flight Information Region, il y en a 5 en France : LFBB, LFEE, LFFF, LFMM et LFRR) en précisant certain critères (durée, IFR/VFR, Flight Level min et max, FIR concernées, ...). Le but est d'utiliser ce formulaire pour récupérer les NOTAM (version complète) de manière programmatique.

La compréhension des notions aéronautiques utilisées dans les NOTAM n'est pas nécessaire pour la réalisation du code.

Il n'y a pas de piège et l'ensemble des recettes nécessaires à la réalisation de challenge sont disponibles sur Google. A vous de savoir trouver, comprendre et adapter en comprenant ce que vous faite.

Le livrable

Vous devrez implémenter le programme en Java 11 et nous fournir le code sous la forme d'un dépôt Github clonable qui sera un fork du dépôt : <https://github.com/innovatm/java-notam-challenge>

Votre code sera récupéré et exécuté dans l'environnement cible :

- Linux Ubuntu ou MacOSX
- Java 11 installé et JAVA_HOME correctement renseignée

Le code et tous les fichiers produits doivent être en anglais (noms des variables et fonctions, documentation, ...).

Pour exécuter le code nous exécuterons les commandes suivantes :

```
$ git clone https://github.com/VOTE_USERNAME/java-notam-challenge
```

Puis :

```
$ javac src/main/java/*.java
$ java -cp src/main/java/ NotamCrawler
```

Le dépôt Github à forker contient le code nécessaire à l'exécution des commandes ci-dessus :

```
$ git clone https://github.com/innovatm/java-notam-challenge
$ cd java-notam-challenge
$ javac src/main/java/*.java
$ java -cp src/main/java/ NotamCrawler
> Ready to implement!
```

La réalisation

A vous d'implémenter le programme à partir du fichier `src/main/java/NotamCrawler.java`

Le programme doit alors récupérer les NOTAM des 96 prochaines heures pour l'ensemble de la France depuis NOTAM-Web et afficher le code du NOTAM et le champ Q dans la console puis se stopper proprement.

Chaque NOTAM ne devra être affiché qu'une seule fois par exécution.

Pour récupérer les NOTAM vous devrez réaliser des requêtes HTTP POST de la forme suivante :

Le service REST à appeler pour soumettre le formulaire est le suivant (verbe http POST) :

```
POST http://notamweb.aviation-civile.gouv.fr/Script/IHM/Bul_FIR.php?FIR_Langue=FR
bResultat true
ModeAffichage COMPLET
FIR_Date_DATE 2020/09/29
FIR_Date_HEURE 14:00
FIR_Langue FR
```

```
FIR_Duree 12
FIR_CM_REGLE 1 // IFR/VFR
FIR_CM_GPS 2 // non
FIR_CM_INFO_COMP 2 // oui
FIR_NivMin 0
FIR_Niv_Max 20
FIR_Tab_Fir[0] LFRR
FIR_Tab_Fir[1 à 9] // optionnel
```

Vous pouvez tester votre requête avec un client tel que curl (documentation simple des options ici : <https://gist.github.com/subfuzion/08c5d85437d5d4f00e58>) :

```
$ curl -F 'bResultat=true' -F 'ModeAffichage=COMPLET' -F
'FIR_Date_DATE=2020/10/05' -F 'FIR_Date_HEURE=19:00' -F
'FIR_Duree=12' -F 'FIR_CM_REGLE=1' -F 'FIR_CM_GPS=2' -F
'FIR_CM_INFO_COMP=2' -F 'FIR_NivMin=0' -F 'FIR_NivMax=5' -F
'FIR_Tab_Fir[0]=LFRR' -X POST http://notamweb.aviation-civile.gouv.fr/Script/IHM/Bul\_FIR.php?FIR\_Langue=FR
```

```
> <HTML><HEAD><TITLE>(R&eacute;sultat) BULLETIN FIR
(Fran&ccedil;ais)</TITLE>
```

```
...
```

Ou bien dans un client lourd tel que Postman (<https://www.postman.com/product/api-client/>) :

POST

http://notamweb.aviation-civile.gouv.fr/Script/IHM/Bul_FIR....

Params

Send

Save

Authorization

Headers

Body

Pre-request Script

Tests

Code

form-data

x-www-form-urlencoded

raw

binary

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	bResultat	true			
<input checked="" type="checkbox"/>	ModeAffichage	COMPLET			
<input checked="" type="checkbox"/>	FIR_Date_DATE	2020/09/29			
<input checked="" type="checkbox"/>	FIR_Date_HEURE	Text 14:00			X
<input checked="" type="checkbox"/>	FIR_Langue	EN			
<input checked="" type="checkbox"/>	FIR_Duree	Text 12			X
<input checked="" type="checkbox"/>	FIR_CM_REGLE	1			
<input checked="" type="checkbox"/>	FIR_CM_GPS	2			
<input checked="" type="checkbox"/>	FIR_CM_INFO_COMP	1			
<input checked="" type="checkbox"/>	FIR_NivMin	0			
<input checked="" type="checkbox"/>	FIR_NivMax	5			
<input checked="" type="checkbox"/>	FIR_Tab_Fir[0]	LFRR			
<input checked="" type="checkbox"/>	FIR_Tab_Fir[1]				
	New key	Value	Description		

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 1781 ms

Pretty

Raw

Preview

HTML

527

528

529

530

531

532

533

534

535

536

537

538

539

540

LFPA-F0840/20

Q)

LFXX/QAFXX/ I/NBO/ E/000/999/4412

A)

LFBB LFEE LFFF LFMM LFRR

B)

2020 Jun 30 23:59

 C)

2020 Sep 30 23:59

E)

FRANCE FIR/UIR ARE PROHIBITED TO

ION CAN BE GRANTED BY DGAC FOR NON-COMMERCIAL

TTED BY EASA AD 2019-0051-R1 AND NECESSARY IN THE

Vous récupérez alors un fichier HTML que vous pouvez parser pour récupérer les NOTAM par manipulation de chaînes.

Pour être sûr de récupérer toutes les NOTAM, il faut utiliser les critères suivants :

- Date et heure de validité (champs FIR_Date_DATE et FIR_Date_HEURE dans le formulaire) : moment actuel
- Durée (FIR_Duree) : 36h
- Règle de vol (FIR_CM_REGLE) : IFR/VFR (valeur 1 dans le formulaire)
- Sélection des NOTAM GPS (FIR_CM_GPS) : non (valeur 2)

- Type NOTAM : Général et divers
- NOTAM sur les aérodromes des FIR sélectionnées (FIR_CM_INFO_COMP) : oui (valeur 2)
- FL min (FIR_NivMin) : 0
- FL max (FIR_NivMax) : 20
- FIR (FIR_Tab_Fir[i]) : les 5 FIR françaises

Attention, une requête retournant trop de NOTAM pourra échouer si le nombre de NOTAM demandé est trop long. Vous devrez donc requêter le site pour chaque FIR et tranches de 6 heures pour couvrir les 96h.

Tout en affichant chaque NOTAM une seule fois.

Bonus

Si vous en avez le temps et la motivation vous pouvez aller un peu plus loin en réalisant un ou plusieurs objectifs bonus :

- Décoder les champs B et C en timestamp et les afficher dans la console avec le code du NOTAM
- Ajouter des options à la ligne de commande : durée, FL min et max, liste des FIR, ...
- Enregistrer pour chaque NOTAM son code et tous ses champs (Q, A, B, D, E, F, G) dans un fichier CSV

Pour finir

Bon courage ! En effet en tant que futur développeur professionnel ce cas pratique est très représentatif des problèmes à résoudre que vous rencontrerez dans votre futur travail. Votre capacité à rechercher, analyser et appliquer les solutions trouvées sur Internet seront la clé.