

# Projet Programmation Orientée Objets 2012-2013

## Informations générales

En programmation orientée objets, nous parlons souvent du concept d'encapsulation des données et de modularité du code. Ceci permet la réutilisabilité du code source d'un programme. Ces morceaux de codes sont généralement rassemblés au sein d'une librairie (de l'anglais *library*). Une API (*Application Programming Interface*) est une interface fournie par une telle librairie. Pour chacune de ces API, les services (méthodes) qu'il est possible d'utiliser depuis l'extérieur sont exposés. De plus, une documentation est généralement fournie (une *javadoc*). Nous vous proposons ici de parachèvement votre apprentissage de Java par un projet de programmation qui a pour but d'acquérir les compétences de bases en Java et de mieux comprendre les notions précédentes.

## Livrables

Il sera demandé aux étudiants de fournir une archive contenant :

- Le code source du programme
- Un rapport de maximum 3 pages qui présente l'architecture de votre programme, la manière dont vous avez implémenté les différentes fonctionnalités, ainsi que les difficultés que vous rencontrées.

Le schéma de nommage de l'archive DOIT être le suivant (pénalité de 1 point en cas de non respect) : pg220-2013-LOGINS.zip, avec LOGINS à remplacer par la liste (classée par ordre alphabétique) des logins des membres du groupe séparés par des underscore (\_).

L'archive est à déposer parmi les fichiers du projet suivant de la forge Enseirb : <https://thor.enseirb-matmeca.fr:9000/projects/pg220> avant le 13 Janvier 2013 à minuit.

## Evaluation

Ce projet est à réaliser par groupe de 2 (3 si besoin) étudiants. Vous serez évalués sur :

- La conception des classes et l'architecture de votre programme.
- La modularité et l'extensibilité du programme (minimiser l'effort pour inclure de nouveaux format de fichier de configurations, de nouveaux types de calendriers, etc.).
- La robustesse de la façon dont vous avez implémenté les fonctionnalités
- La qualité de l'utilisation des librairies externes relatives à ce projet

- La gestion des erreurs

Vous devez absolument RESPECTER les contraintes suivantes. Une pénalité de 2 points sera appliquée sur chaque projet qui ne les respecte pas.

- Votre projet doit contenir un ou plusieurs packages
- Les noms de packages sont intégralement en minuscules
- Les noms de classes et interfaces doivent commencer par une majuscule
- Les noms de méthodes, d'attributs et de variables doivent commencer par une minuscule
- Utilisez le camel case pour les noms des classes, interfaces, méthodes et attributs.
- Les attributs doivent avoir une visibilité. Par défaut, les attributs sont *private*. Des méthodes de type *get/set* doivent permettre de gérer l'accès depuis l'extérieur à ces variables. Attention, définissez ces méthodes pour chaque attribut, seulement si elles sont utiles.
- Définissez des constructeurs que lorsqu'ils sont nécessaires.
- Commentez au maximum votre code. Expliquez ce que font vos méthodes quand elles sont complexes. N'expliquez pas tout et n'importe quoi. Soyez concis et allez à l'essentiel, indiquez ce qui est réellement utile à la compréhension.

## Description du sujet

L'objectif de ce projet est la réalisation d'un notificateur d'évènement Twitter. Ce programme s'exécute en arrière plan et surveille des calendriers au format iCal dans lesquels sont spécifiés des évènements. Quand la date en cours est inférieure d'un délai fixé d'un évènement, le programme va alors twitter un message comportant la description de l'évènement. Le notificateur sera capable de gérer deux types de calendriers : les emplois du temps ENSEIRB et un calendrier contenant la météo sur Bordeaux. Les différents calendriers qui seront gérés par le programme seront décrits dans un fichier de configuration au format JSON. Enfin le programme maintiendra la liste des actions qui ont déjà été effectuées de sorte que, si le programme est redémarré, il ne répètera aucun message qui a déjà été envoyé.

## Librairies à utiliser

Afin de parvenir à implémenter ce projet, nous allons utiliser les librairies ci-dessous afin de bénéficier des services spécifiques qu'elles offrent. Pour chacune d'elles, ayez le réflexe de consulter les informations disponibles sur la page web, telle que le javadoc, tutoriaux ou diverses documentations qui vous aideront à prendre en main les libraires. Bien entendu la documentation de meilleure qualité se trouve en général en anglais sur la page officielle.

1. *Jackson* pour le traitement de fichiers [JSON](http://jackson.codehaus.org/) : <http://jackson.codehaus.org/>
2. *iCal4J* qui permet de manipuler des documents au format [iCalendar](http://ical4j.sourceforge.net/index.html) : <http://ical4j.sourceforge.net/index.html>
3. *Twitter4J* qui permet d'accéder aux services de [Twitter](http://twitter4j.org/) via Java : <http://twitter4j.org/>

## Configuration de Twitter

Pour accéder aux services de Twitter par l'intermédiaire d'une application Java, il est nécessaire de passer par une procédure d'enregistrement (plus précisément, Twitter utilise le modèle d'authentification [OAuth](#)). La première étape consiste à créer un compte twitter qui sera dédié à votre application. Utilisez le nom de compte `eirb_pg220_gID` avec `ID` le numéro de votre groupe. Ensuite enregistrez cet utilisateur Twitter comme développeur en utilisant le site <https://dev.twitter.com>. Une fois enregistré et connecté sur ce site utilisez le lien <https://dev.twitter.com/apps/new> pour créer une nouvelle application. Pour cela vous devez remplir les champs Nom, Description et Website (mettez votre adresse de site perso Enseirb pour le champ Website). A l'issue de cette procédure vous pourrez récupérer le `consumerKey` et `consumerSecret` de l'application. Dans settings, paramétrer l'application pour qu'elle ait le droit d'envoyer des messages (Access : read, write and direct messages). Pour finaliser la procédure et pouvoir twitter en utilisant l'API `Twitter4J` vous devez autoriser votre application à utiliser le compte twitter de l'utilisateur que vous venez de créer. Pour cela, indiquez votre `consumerKey` et `consumerSecret` dans le code de la classe `fr.enseirb.twitter.GetAccessToken` (dispo ici <http://www.labri.fr/perso/cteyton/Twitter/GetAccessToken.java>), puis exécutez cette classe pour finaliser l'enregistrement (attention, il faut préalablement compiler la classe en mettant la librairie `Twitter4J` dans le classpath). Le programme vous fournira une URL sur laquelle vous devrez vous rendre pour récupérer le code Pin. Indiquez ce code Pin au programme en cours (via la console), puis notez vos nouvelles données `Access Token` et `Access Token Secret`. Gardez bien ces 4 valeurs, qu'il vous faudra mettre dans le fichier de configuration JSON et fournir à `Twitter4J` pour que vous puissiez twitter. Ce processus est détaillé ici en cas de problème : <http://consultingblogs.emc.com/nileeshabojjawar/archive/2010/03/18/twitter4j-oauth-generating-the-access-token.aspx>

## Liste des fonctionnalités

### Fichier de configuration

Voici un exemple de fichier de configuration. Il est défini en utilisant le format [JSON](#).

```
{
  "twitter_consumer_key" : "8v1eT4LGdNouIE3vvd8gg",
  "twitter_consumer_secret" : "8A2USuMEGpVVUFusvcIqBfRBxV1JsVsPJozLMszcQ1",
  "twitter_access_token" : "563530639-Jc6AzByW2akQdm9zjsKz11MruTCQDSYvs8p344rd",
  "twitter_access_token_secret" : "4TbMegTx6BF80Lzu6u9f0XWfyaiHrjDqsXmFQYdgv",
  "period" : 60,
  "calendars" : [
    {
      "url" : "/home/user/t1.ics",
      "kind" : "EIRB",
      "expire" : -1,
      "delay" : 60
    },
    {
      "url" : "http://ical.wunderground.com/auto/ical/global/stations/07510.ics?units=metric",
      "kind" : "WEATHER",
      "expire" : 1440,
      "delay" : 1440
    }
  ]
}
```

```
}  
 ]  
}
```

La structure de ce fichier est la suivante. Tout d'abord, il faut fournir les 4 informations personnelles pour accéder aux services Twitter. La section précédente explique comment obtenir ces données. En outre, un attribut `period` définit la périodicité d'activation du notificateur (en minutes). A chaque activation, les actions relatives à la gestion des calendriers et des twits doivent être déclenchées. Une fois achevée, la notification est mise en sommeil pendant une nouvelle période de temps.

Enfin, les différents calendriers sources sont définis dans un élément `calendars`. Cet élément est un tableau dont chaque entrée représente un calendrier source. Pour chaque entrée, il est spécifié les informations suivantes :

- `url`: l'url du calendrier, unique dans le fichier de configuration
- `kind`: le type de calendrier, qui est soit `EIRB` soit `WEATHER`
- `expire`: la période (en minutes) à l'issue de laquelle le calendrier doit être rechargé à partir de son url.
- `delay`: le délai (en minutes) à partir duquel les événements sont notifiés

La première étape consiste à programmer un composant qui est capable de lire les informations contenues dans un fichier de configuration.

## Programme de notification

Le programme de notification est le programme qui va permettre d'envoyer les tweets en fonction du contenu des calendriers. Ce programme fonctionne de la manière suivante :

1. Il charge le fichier de configuration en utilisant le composant qui a été développé précédemment
2. Il charge en mémoire tous les calendriers spécifiés dans le fichier de configuration
3. Il charge le fichier de reprise sur arrêt, qui permet de savoir quels sont les twits qui ont déjà été envoyés
4. Ensuite il va périodiquement s'activer pour effectuer les actions requises. La durée qui s'écoule entre chaque activation est définie par l'attribut `period` du fichier de configuration.

## Fonctionnement de l'activation du programme

A chaque activation du programme de notification, celui-ci vérifie les actions qui doivent être effectuées sur les différents calendriers. Voici la liste des différentes actions possibles :

- `SEND_TWEET(CAL,EVT)`. Lorsque que dans un calendrier `CAL`, un événement `EVT` se trouve dans la situation suivante :
  - a. `actual_time`  $\geq$  `start_time` - `delay` avec `actual_time` le temps actuel, `start_time` la date de départ de l'évènement `EVT`, `delay` pour le calendrier `CAL` est spécifié dans le fichier de configuration.
  - b. `EVT` n'a jamais été twitté

alors l'évènement `EVT` doit être twitté. La manière dont un événement doit être twitté dépend du `kind` de l'évènement (voir les détails dans les section dédiées).

- REFRESH\_CALENDAR(CAL). Si un calendrier se trouve dans la situation suivante :
  - a.  $\text{download\_time} + \text{expire} > \text{actual\_time}$  avec  $\text{download\_time}$  la dernière date de téléchargement du calendrier CAL,  $\text{actual\_time}$  le temps actuel,  $\text{expire}$  pour le calendrier CAL est spécifié dans le fichier de conf
  - b.  $\text{expire} \geq 0$alors le calendrier doit être rechargé en mémoire. La nouvelle date de téléchargement doit bien entendu être mémorisée. Si  $\text{expire}$  est négatif, le calendrier n'est jamais rechargé.

Il est à noter que l'action REFRESH\_CALENDAR(CAL) d'un calendrier donné, si elle est effectuée, doit être déclenchée avant de vérifier si il y a des actions de type SEND\_TWEET(CAL,EVT) à effectuer.

### Reprise sur arrêt

Si jamais à cause d'une erreur le programme s'arrête, il doit être possible de le relancer. Dans ce cas, le programme ne doit pas retwitter les événements qui avaient déjà été twittés. Vous devez proposer une solution qui permet de s'assurer de ce comportement.

### Calendrier EIRB

Pour ce type de calendrier, le programme de notification va afficher un message contenant la description et la salle du cours sur un hashtag qui est inclus dans le summary de l'événement. Exemple : si le summary est :

```
SUMMARY:Cours - IF 103 Structure des ordinateurs\, Barthou\, Denis\, T1\, T1
```

alors le tweet doit avoir la forme suivante :

```
#EIRB_T1 SUMMARY (LOCATION)
```

Avec SUMMARY et LOCATION les champs correspondants dans l'événement. L'objectif est de publier les messages sur des hashtags correspondant aux différentes promotions et groupes de l'Enseirb : #EIRB\_T2, #EIRB\_T1\_G1, etc... Il est à noter que les calendriers des cours de l'Enseirb seront téléchargés sur le disque dur.

### Calendrier WEATHER

Pour ce type de calendrier, le programme de notification va afficher un message standardisé dont le contenu est

```
#EIRB_WEATHER Tomorrow's weather: SUMMARY
```

SUMMARY étant à remplacer par la valeur de ce champ dans l'événement concerné. L'URL du calendrier iCal contenant la météo de bordeaux est la suivante : <http://ical.wunderground.com/auto/ical/global/stations/07510.ics?units=metric>.

## Du temps libre ?

Dans le cas où vous avez accompli le projet en intégralité, vous pouvez ajouter des sources d'informations supplémentaires qui peuvent intéresser les étudiants de l'Enseirb. Chaque ajout donnera lieu à des points bonus. Pour chaque type de calendrier, vous devrez donc définir son kind, et en fonction de celui-ci la façon de construire le twit en fonction de l'évènement. Pour ne pas polluer *Twitter*, vous posterez systématiquement les twits sur des hashtags préfixés par #EIRB\_.

Voici des exemples de calendriers intéressants :

- Les matchs de foot des girondins de Bordeaux : <http://www.lfp.fr/iCalendar/250.ics>
- Les concerts sur Bordeaux : <http://www.5gig.fr/feed/ville/3031582/ical>