

Auteur : Anna Kostrikova		21/10/16
Type : Compte rendu de réunion		Equipe : <Anna Kostrikova, Julien Margarido, Félix Lima Gorito>
Chemin : <github.com/JulienMrgrd/lab-bot/docs >		Statut : <input type="text" value="Initial"/>
Destinataire(s) : <Jonathan Alimi, Mikael Gibert>		Page(s) 3
<Suricats - Bots et interactions réseaux sociaux>		
Compte rendu de réunion n°3 du 21 octobre 2016		

Version	Date	Modifications	Rédacteur
1	20/10/2016	-	Anna Kostrikova

Objectif :

Définir le Sprint 3

Ordre du jour

- Démonstration du Sprint 2
- Définition du Sprint 3

Participants

- Mikhael Gibert

Déroulement

Discussion avec Mikael Gibert (quelques passages de la discussion sont complétés avec des informations internet).

Ce qui c'est bien passé

- communication
- bot connecteur
- github pull request

Ce qui c'est mal passé

- info archi oublié

- gestion de sockets (à faire répertoire *Doc* dans repo)
- accès sudo

Démonstration

Réalisation du bot avec Microsoft Bot Connector.

Apache Maven

Un outil pour la gestion et l'automatisation de projets java. L'objectif comparable à *make*. Semblable à l'outil *Ant*, mais fournit des moyens de configuration plus simples. Au lieu de créer dossier *Lib* et y mettre *jar*, maven cherche directement APIs qui sont disponibles quelque part.

Project Object Model, POM, se matérialisant par *pom.xml* à la racine, contient les informations nécessaires à Maven pour traiter le projet : nom du projet, numéro de version, dépendances vers d'autres projets, bibliothèques nécessaires à la compilation, noms des contributeurs etc.

L'héritage des propriétés du projet parent.

Aptitude à fonctionner en réseau : peut dynamiquement télécharger du matériel sur des dépôts logiciels connus, la synchronisation transparente de modules nécessaires.

Sous eclipse : run as / maven ... - compilation du projet.

Travis

Travis a deux repos source : un chez suricats et un chez les étudiants, pour travis c'est 2 repos differentes. Chaqu'un a *travis.yml*.

Deployment = lancement d'une application sur serveur.

Travis peut (ou pas) lancer des testes avant deployment (et eventuellement, selon les resultat de testes, peut bloquer deployment), puis fait deploymenet automatique.

Travis rajoute les variables d'environmnet. On peut les mettre en claire sur travis (pas visibles).

On peut placer mots de passe, etc dans *travis.yml*. Command de travis : *secure mot-de-passe*, il retourne un truc sécurisé => on peut mettre le mot de passe en claire dans *travis.yml*. <https://docs.travis-ci.com/user/deployment/script/> .

Convention plutôt que configuration

Une pratique informatique qui tend à faire décroître le nombre de décisions qu'un développeur doit prendre. Par exemple, s'il existe un type *Vente* dans le modèle, la table correspondante dans la base de données se nommera implicitement *ventes*. La convention s'est mise en place par l'outil de développement correspond.

Spring framework

Construire et définir l'infrastructure d'une application java3, faciliter le développement et les tests.

Spring boot

Fait partie de Spring framework.

Convention plutôt que configuration solution.

Crée applications autonome spring-basées, lesquels on peut "just run".

Met en place Tomcat ou Jetty directement, pas besoin de deployer war files.

Fournit 'starter' *pom.xml*.

Automatiquement configure Spring dans la mesure de possible.

Fournit instruments d'analyse comme metrics, health checks et externalized configuration.

Aucune generation de code.

Aucune contrainte pour configuration XML.

Spring boot : un seul process qui tourne (par exemple, tomcat : deux processus, n'est pas adapté pour bot).

<https://projects.spring.io/spring-boot/>

pom.xml :

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.4.1.RELEASE</version>
</parent>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
</dependencies>
```

DropWizard

Un framework complet et éprouvé pour faciliter le développement de services web RESTful.

```
<dependencies>
  <dependency>
    <groupId>io.dropwizard</groupId>
    <artifactId>dropwizard-core</artifactId>
    <version>0.7.1</version>
  </dependency>
</dependencies>
```

API

Un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web, le plus souvent accompagnée d'une description qui spécifie comment des programmes *consommateurs* peuvent se servir des fonctionnalités du programme *fournisseur*.

pynetlib / build

A ajouter *pynetlib / build*

This library is a toolbox to manipulate network configurations on Linux.

<https://github.com/migibert/pynetlib>

Remarques sur l'organisation du travail

- utiliser port 12345
- ne changer pas le langage, le langage n'est pas un problème
- dire si on a des problèmes
- un seul bot microsoft pour toute l'équipe, mot de passe à déposer sur slack

Ordre reporté

Tache à effectuer :

1) Nom de la tâche : (Sprint 3) transformer le bot du sprint 1 en API.

Pour rappel :

sprint 1 : slack ↔ bot java

sprint 2 : slack ↔ (microsoft bot connector + son faux bot)

sprint 3 : slack ↔ (microsoft bot connector + son faux bot) ↔ bot java

S'aider par Sprint Boot / dropWizard, ajouter dépendances dans *pom.xml*.

Qui doit la faire : le groupe.

Quand doit-elle être effectuée : avant la prochaine réunion le 28/10/2016.

2) Maveniser notre truc.

Qui doit la faire : le groupe.

Quand doit-elle être effectuée : avant la prochaine réunion le 28/10/2016.

3) Ecrire *travis.yml*.

Qui doit la faire : le groupe.

Quand doit-elle être effectuée : avant la prochaine réunion le 28/10/2016.

Prochain greffier

Félix Lima Gorito

Ce compte-rendu sera validé automatiquement au bout de 7 jours.