Cours 7 : Pour aller plus loin

Mardi 15 Décembre 2020

Antoine Flotte (aflotte@excilys.com)

Excilys
Développeurs de passion

SOMMAIRE

Java

Spring Security est un des modules de Spring qui vous permet de sécuriser votre application.

- mise en place d'utilisateurs enregistrés en base (authentification)
- protection contre des attaques
- gestion de rôle

Se configure avec des classes @Configuration

Si vous voulez mettre en place un système de Notification il faut utiliser les websockets.

Jusqu'à présent c'est toujours votre front qui questionne votre back, mais pour une notification c'est l'inverse.

Vous pouvez donc questionner votre back tous les x temps (pulling) mais c'est très coûteux si vous avez pas mal d'utilisateurs connectés.

Le principe d'un websocket c'est que le front demande au back d'ouvrir un canal sécurisé, puis le back pourra utiliser ce canal pour parler avec le front.

Pour mettre ça en place, encore une fois Spring nous propose un module, Spring Messaging, et Spring Security Messaging pour sécuriser le canal.

Vous pouvez optimiser votre code avec des threads, il faut alors faire attention à si vos objets sont Thread-safe ou non.

On a parlé des Streams, il existe aussi des ParallelStream qui parallélise les traitements. Au lieu d'appeler .stream() sur votre liste, vous appelez .parallelStream().

Profile Spring

Utiliser des profils Spring ou Maven vous permet d'avoir des fonctionnement spécifiques sur votre prod.

Exemple :

url du front

une API non disponible en local

des configurations particulières

...

Vous pouvez améliorez vos logs ! Suivant la librairie que vous utilisez vous pouvez les configurer avec un fichier de configuration permettant de choisir la sortie, d'afficher la date et l'heure, la classe et même la ligne en question de façon automatisé à chaque appel.

Vous pouvez aussi mettre en place des fichiers de log tournant…

Exemple de configuration

```
# Root logger option
log4j.rootLogger=INFO, stdout

# Direct log messages to stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
%c{1} - %m%n
```

Pour recevoir :

```
2017-03-21 23:31:13 INFO  HelloLog4J - This is how you configure Log4J with SLF4J
```

Lombok est une librairie qui génère pour vous énormément de code de manière propre. Par exemple, les getters/setters les toString(), etc.

Pas toujours apprécié car les gens aiment avoir la main sur ça, mais vous choisissez où vous l'utilisez donc très pratique.

Lombok exemple

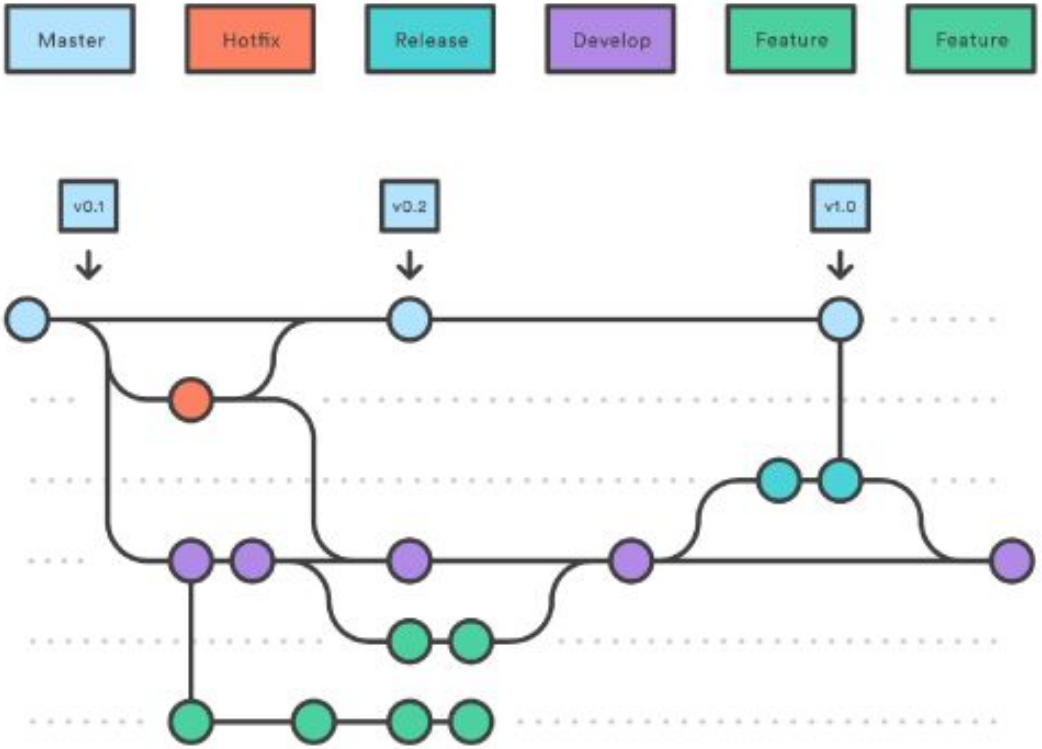
```
@Entity
@Getter
@Setter
@ToString
@EqualsAndHashCode
@NoArgsConstructor
@AllArgsConstructor
public class Film {

    private long id;
    private String titre;
    private int duration;
    private Realisateur realisateur;

}
```

Projet

Gitflow



Docker permet d'avoir des containers, ça se rapproche d'une VM mais on partage le même système d'exploitation que le système hôte, mais aussi ses librairies et binaires quand c'est possible.

C'est donc beaucoup plus léger qu'une VM et permet de s'assurer de déployer constamment sur la même machine.

Le principe de GitLab CI (et de l'Intégration Continue de manière plus générale), c'est d'automatiser les actions à faire quand vous voulez merger votre code.

Tout d'abord, cela permet de déployer automatiquement votre projet quand vous mettez à jours master et preprod.

Sur GitLab on ajoute un fichier gitlab.yml qui va contenir les informations nécessaire.

On a aussi besoin de lancer des Runners, qui vont s'occuper d'effectuer les scripts. Ces Runners peuvent être mis en place avec Docker.

On peut même automatiser le lancement des tests et si ils ne passent pas, alors la merge est refusé. De même avec des outils comme Sonar on peut demander d'avoir un coverage minimum, et un code propre.

OVH - GCP - AWS - Un PC qui sert de serveur

Apache2 - Tomcat - Nginx

Tout ça demande des configurations, c’est le travail du dev-ops ou du tech lead en général (donc vous n’avez pas à vous en soucier sauf pour les projets perso)

Une vraie base de donnée mySQL, base permanente H2, ou autre.

Les bases H2 temporaires sont idéals pour les projets tests, et pour tester les DAOs.

Là encore on peut mettre la base dans un conteneur Docker pour faciliter les déploiements automatisés.

On peut aussi optimiser certaines fonctions Java en mettant en place des fonctions SQL, et des Triggers. Cela permet de diminuer les appels à la base de données.

Rappel : créer une connection à la base de données est négligeable, mais quand on les enchaîne dans une boucle ce n’est plus le cas.

On vous propose aussi un stage de fin d’étude de 6 mois.

Contactez moi par mail si vous voulez plus d’info à ce sujet et je vous mettrais en contact avec un recruteur.

Rendu

Le projet est à rendre pour le Vendredi 8 Janvier par mail à aflotte@excilys.com

Vous recevrez un mail pour confirmer la réception.