

Projet 4A

Design Objet

Année 2015

Nicolas André
nicolas.andre@gleetr.net

Objectifs

- L'objectif de ce projet est de concevoir et développer un framework de logging.
- Le principe d'un framework de logging est de mettre à disposition un système simple pour gérer les messages d'erreur et comprendre le flux d'exécution d'un programme.
- Un même message peut donner lieu à plusieurs « cibles » :
 - une trace dans la console
 - une ligne dans un fichier
 - un enregistrement dans une base de données
 - un mail à destination d'une équipe support
 - etc ...

Les 3 composantes

- Un framework de logging se base généralement sur les 3 composants suivants :
 - Un Logger pour émettre le message
 - Un formateur pour personnaliser le contenu du message
 - Une ou plusieurs cibles pour stocker le message

Architecture des cibles

Cibles

```
logger.error("message");
```

Logger



Console

Fichier

Base

▪
▪
▪

Cible n

1 message : plusieurs cibles de stockage possibles

API de loggings

```
// DEBUG < INFO < ERROR
Logger logger = LoggerFactory.getLogger(MaClasse.class);
logger.debug("Test log ESIEA");
logger.info("Test log ESIEA");
logger.error("Test log ESIEA");
```

Trace dans la console

```
2015-01-16 18:05:16.227 [NAME=FR.ESIA.TP LEVEL=DEBUG MESSAGE= TEST LOG ESIEA]
2015-01-16 18:05:16.234 [NAME=FR.ESIA.TP LEVEL=INFO MESSAGE= TEST LOG ESIEA]
2015-01-16 18:05:16.235 [NAME=FR.ESIA.TP LEVEL=ERROR MESSAGE= TEST LOG ESIEA]
```

// Formateur des messages

```
logger.debug("Test log ESIEA");
```

```
2015-01-16 18:05:16.227 [NAME=FR.ESIA.TP LEVEL=DEBUG MESSAGE= TEST LOG ESIEA]
```

Ajout date/heure + infos

Règles de priorités des logs

DEBUG < INFO < ERROR

```
// Si configuration du logger pour MaClasse = INFO
Logger logger = LoggerFactory.getLogger(MaClasse.class);
logger.debug("Test log ESIEA");
logger.info("Test log ESIEA");
logger.error("Test log ESIEA");
```

Trace dans la console pour une configuration INFO

```
2015-01-16 18:05:16.227 [NAME=FR.ESIA.TP LEVEL=INFO MESSAGE= TEST LOG ESIEA]
2015-01-16 18:05:16.234 [NAME=FR.ESIA.TP LEVEL=ERROR MESSAGE= TEST LOG ESIEA]
```

Trace dans la console pour une configuration ERROR

```
2015-01-16 18:05:16.227 [NAME=FR.ESIA.TP LEVEL=ERROR MESSAGE= TEST LOG ESIEA]
```

Identification d'un Logger

- Un logger est identifié par un nom de classe. Cet identifiant de logger est ensuite utilisé par la configuration du framework.

```
public class MaClasse {  
  
    private Logger logger = LoggerFactory.getLogger(MaClasse.class);  
  
    public void init() {  
        logger.info("message");  
    }  
}
```

Configuration du framework

- Pour chaque logger, il est possible de spécifier le niveau de la priorité filtrée du message, la ou les cibles du message, le formateur associé.

Exemple de configuration en Java

```
configuration.ajouterCible(MaClasse.class, new ConsoleCible());  
configuration.ajouterCible(MaClasse.class, new FileCible("/path/  
file.txt"));  
configuration.setLayout(MaClasse.class, new Formateur());  
configuration.setLevel(MaClasse.class, DEBUG);
```


Spécification du framework de logging

- Possibilité de configurer votre framework en Java et dans un format textuel (fichier Properties Java).
- Possibilité de définir des nouvelles cibles de messages.
 - Fournir les cibles suivantes : console, fichier et fichier rotatif (en fonction de la taille)
- Possibilité de déclarer ses propres formateur de message.
 - Fournir un formateur affichant la date du message, le niveau de log et le nom du logger.

Exemple de configuration Properties

Exemple de configuration en fichier properties.

```
logger.fr.esiea.MaClasse.level = DEBUG  
logger.fr.esiea.MaClasse.formater = fr.esiea.formater  
logger.fr.esiea.MaClasse.cible1 = fr.esia.ConsoleCible  
logger.fr.esiea.MaClasse.cible2 = fr.esia.FileCible  
logger.fr.esiea.MaClasse.cible2.path = "/path/log.txt"
```

Règles de Design Objet

- Votre framework doit respecter le principe OCP : Open Close Principle. Il doit être en effet possible de configurer le framework avec de nouvelles cibles sans modifier le code existant de votre framework.

Livrables

- Les codes sources de votre framework de logs doivent être livrés dans un repository github.
- La documentation de l'architecture ainsi qu'une documentation d'installation et d'utilisation de votre framework doivent être fournies dans le repository github du projet.

Délai de réalisation

- Les équipes projet doivent être composées au maximum de 3 personnes.
- Date de remise du projet : 27/02/2015 à 11 AM, chaque équipe devra communiquer la liste des noms des étudiants avant le 23/01/2015 à 11 AM.
- Contact Nicolas ANDRE : nicolas.andre@gleetr.net