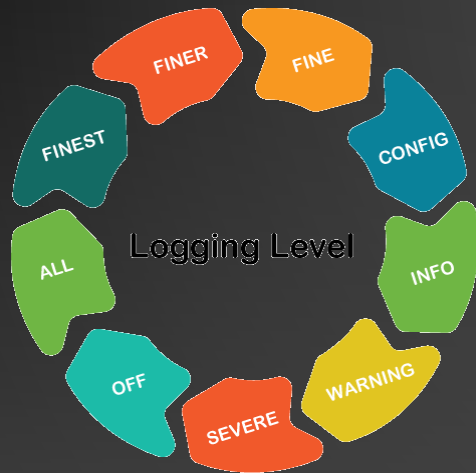


LOGGING

SLF4J & SPRING BOOT



Spring1's - 2023

PLAN

1 – Logging, qu'es-ce ?

2 – Bibliothèques de logging dans
Spring Boot

3 – Les outils : SLF4J & Spring
Boot Logging

4 – Ressources



LOGGING, QU'ES-CE ?

Le Logging permet d'enregistrer dans des fichiers ou de visualiser dans la console erreurs, avertissements, les problèmes...

Par défaut, Spring Boot utilise la librairie Logback pour sa journalisation, vous pouvez personnaliser les paramètres.

LOGBACK

Qui est Logback :

Loggin logback est une librairie intégrée à java.

Il est conçu comme un successeur du populaire log4j.

Le projet de logback est sous double licence LGPL et l'EPL (licence libre).



LIBRAIRIES DE LOGGING DANS SPRING BOOT

- Spring Boot par default permet d'utiliser la JAVA Util Logging, Log4j2 and Logback.
- D'autres librairies ou couches d'abstraction peuvent aussi être importées, comme SLF4J.
- Spring Boot Log produit un log format de base avec :
 - Date et Time
 - Log level
 - Process ID
 - Thread name
 - Logger Name (classe qui a déclenché le log)
 - Le message de log

```
2016-04-21 08:44:33.381 DEBUG 4500 --- [nio-8080-exec-1] g.s.controllers.IndexController : This is a debug message
2016-04-21 08:44:33.383 INFO 4500 --- [nio-8080-exec-1] g.s.controllers.IndexController : This is an info message
2016-04-21 08:44:33.386 WARN 4500 --- [nio-8080-exec-1] g.s.controllers.IndexController : This is a warn message
2016-04-21 08:44:33.387 ERROR 4500 --- [nio-8080-exec-1] g.s.controllers.IndexController : This is an error message
2016-04-21 08:44:33.411 INFO 4500 --- [nio-8080-exec-1] g.s.helpers.SpringLoggingHelper : This is an info message
2016-04-21 08:44:33.413 WARN 4500 --- [nio-8080-exec-1] g.s.helpers.SpringLoggingHelper : This is a warn message
2016-04-21 08:44:33.413 ERROR 4500 --- [nio-8080-exec-1] g.s.helpers.SpringLoggingHelper : This is an error message
```


LOG LEVELS

Tous les niveaux de logs sont supportés par Spring Boot :

“TRACE”, “DEBUG”, “INFO”, “WARN”, “ERROR”, “FATAL”, “OFF”.



3 - LES OUTILS : SLF4J

Le SLF4J ou **S**imple **L**ogging **F**açade **f**or **J**ava : une couche d'abstraction pour divers frameworks de logging.

Permet de connecter différents frameworks de logging au déploiement sans toucher au code.



AJOUTER SLF4J LOGGER

```
package com.tutorialspoint.demo;

import org.slf4j.Logger; 1
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoApplication { 2
    private static final Logger logger = LoggerFactory.getLogger(DemoApplication.class);

    public static void main(String[] args) {
        logger.info("this is a info message"); 3
        logger.warn("this is a warn message");
        logger.error("this is a error message");
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

1. Importer la librairie
2. Créer un objet logger à partir de la logger factory, en passant le nom de la classe en paramètre.
3. Appel de l'objet logger afin de créer des logs avec un niveau de sévérité et un message.

RESSOURCES

- https://www.tutorialspoint.com/spring_boot/spring_boot_logging.htm
- https://www.tutorialspoint.com/spring_boot/spring_boot_logging.htm
- <https://sematext.com/blog/slf4j-tutorial/>
- <https://devstory.net/11753/utiliser-logging-dans-spring-boot>