

TP Logging

PREMIERS PAS AVEC SLF4J / LOGBACK

Créer un projet Maven demo-logging.

Ajouter la dépendance vers SLF4J :

```
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.7.18</version>
</dependency>
```

Ajouter la dépendance vers LogBack :

```
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
  <version>1.2.2</version>
</dependency>
```

Ajouter un fichier de configuration des logs dans **src/main/resources** :

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <!-- redirection des logs vers la sortie Console Standard -->
  <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} -
        %msg%n</pattern>
    </encoder>
  </appender>
  <!-- règle de logs par défaut. Ici : niveau INFO vers la sortie standard -->
  <root level="INFO">
    <appender-ref ref="STDOUT" />
  </root>
</configuration>
```

Créer la classe **dev.console.App** :

```
public class App {  
    private static final Logger LOG = LoggerFactory.getLogger(App.class);  
    public static void main(String[] args) {  
        LOG.info("Bonjour SLf4J !");  
        LOG.info("Implementation Logback");  
    }  
}
```

Tester la classe ci-dessus.

Modifier la configuration pour avoir la sortie console suivante :

```
Bonjour SLf4J !  
Implementation Logback
```

APPENDERS LOGBACK

Logback propose de nombreux *appenders* (<https://logback.qos.ch/manual/appenders.html>) :

- OutputStreamAppender
- ConsoleAppender
- FileAppender
- RollingFileAppender
- SocketAppender et SSLSocketAppender
- SMTPAppender
- DBAppender
- et bien d'autres...

NOUS ALLONS METTRE EN OEUVRE LE FILEAPPENDER.

Ajouter la classe **dev.service.AppService** :

```
public class AppService {  
    private static final Logger LOG =  
        LoggerFactory.getLogger(AppService.class);  
  
    public void executer(String param) {  
        LOG.debug("Traitement 1 : param = {}", param);  
    }  
}
```

Modifier la classe **dev.console.App** :

- Invoquez la méthode `executer` de la classe `AppService` avec un paramètre de votre choix.
- Testez
 - la trace de la classe `AppService` n'apparaît pas. Modifiez la configuration `LogBack` afin qu'elle apparaisse dans la console `STDOUT`.

Modifiez la configuration **LogBack** :

- Ajoutez un **appender** de type fichier à la configuration:

```
<appender name="FILE" class="ch.qos.logback.core.FileAppender">
  <file>app.log</file>
  <append>true</append>
  <immediateFlush>true</immediateFlush>
  <encoder>
    <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} -
      %msg%n</pattern>
  </encoder>
</appender>
```

- Ajoutez également le logger suivant utilisant cet appender :

```
<logger name="dev.console" level="INFO">
  <appender-ref ref="FILE" />
</logger>
```

- Testez
 - le fichier **app.log** devrait apparaître à la racine de votre projet STS. Faites un F5 sur votre projet pour le faire apparaître.
- Modifiez la configuration afin que la trace de la classe `AppService` s'inscrive dans le fichier **app.log** mais **pas dans la console**.

[COMMITEZ SUR GITHUB](#)