

Développement de Batches

Artifact Projet Batch avec Spring 3.2 et Java 6

Logging

On utilisera la librairie SLF4J pour logger les traces des classes java.

La configuration est contenu dans un fichier log4j.properties à déposer dans le répertoire

- src/main/resources

Déclaration du logger dans les classes java

```
Logger loggerSLF4J = LoggerFactory.getLogger(ArtifactJUnitTest.class);
```

Exemple de fichier log4j.properties

```
// Sortie standard vers la console

log4j.rootLogger=TRACE, 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'T'HH:mm:ss.SSS} %-5p [%c] - %m%n
```

Niveaux de logs

Les niveaux de logs du moins au plus important :

- TRACE
- DEBUG
- INFO
- WARN
- ERROR
- FATAL
- ALL

log4j.rootLogger=TRACE ou ALL = On trace tous les niveaux

Rolling File (à prévoir)

Les Tests Unitaires

Chaque méthode de Test sont annotées avec l'annotation @Test

- Configuration Spring

Les classes de Test basées sur Spring doivent porter la configuration suivante en entête de classe :

Entête de classe

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = { "classpath:spring/applicationContext.xml" })
public class ArtifactSpringTest {
    .....
}
```

Configuration Base de données

Configuration Ressources JMS

Les Mocks

Mémento Spring

- Injection d'une propriété système, null si pas trouvé

```
@Value("#{systemProperties['my-property'] ?: null}")
private String myPropertySystem;
```

- Injection d'une propriété provenant d'un fichier properties avec valeur par défaut si pas trouvée

```
@Value("${my.property.fonc: Valeur par défaut}")
private String myPropertyFonc;
```

- Injection de plusieurs propriétés sous forme de tableau de String

```
my.property.array.fonc = A, B, C
.....

@Value("${my.property.array.fonc:}")
private String[] myPropertyArrayFonc;
```

- Injection de plusieurs propriétés sous forme de liste de String

```
my.property.array.fonc = A, B, C
.....

@Value("#{${my.property.array.fonc}'.split(',')}")
private List<String> myPropertyListFonc;
```

Redmines

<http://qlprod.orleanssrv.domsma/issues/36179>