

Padrão de LOGs

Este documento apresenta todos as informações necessárias para o uso de Logs nos sistemas da GETIC.



Configurações

Alteração no POM.XML

Adicione a dependência do *Log4j* no pom.xml.

```
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-log4j12</artifactId>
  <version>1.8.0-alpha1</version>
  <scope>compile</scope>
</dependency>
```

Arquivo log4j.properties

Adicione o arquivo *log4j.properties* na para de recursos da aplicação.

```
# Root logger option
log4j.rootLogger=${log4j.level}, file, stdout

# Redirect log messages to console
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}:%L - %m%n

# Redirect log messages to a log file, support file rolling.
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=${jboss.home.dir}/standalone/log/<SISTEMA>
/<SISTEMA>.log
log4j.appender.file.MaxFileSize=5MB
log4j.appender.file.MaxBackupIndex=3
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd
HH:mm:ss} %-5p %c{1}:%L - %m%n
```

Utilizando LOGGER

Importe a classe logger.

```
import org.apache.log4j.Logger;
```

Crie uma propriedade estática que represente o Logger;

```
public class NomeDaClasse {

    static final Logger logger =
Logger.getLogger(MethodHandles.lookup()

.lookupClass());

}
```

Substitua: `e.printStackTrace();` por `logger.error(e.getMessage(), e);`

Antes:

```
try {  
  
} catch (Exception e) {  
    FacesUtils.addMensagemErro(MENSAGEM_ACESSO_BANCO);  
    e.printStackTrace();  
}
```

Depois:

```
try {  
  
} catch (Exception e) {  
    FacesUtils.addMensagemErro(MENSAGEM_ACESSO_BANCO);  
    logger.error(e.getMessage(), e)  
}
```