

Anyframe Log Manager Plugin



Version 1.1.0

저작권 © 2007-2014 삼성SDS

본 문서의 저작권은 삼성SDS에 있으며 Anyframe 오픈소스 커뮤니티 활동의 목적하에서 자유로운 이용이 가능합니다. 본 문서를 복제, 배포할 경우에는 저작권자를 명시하여 주시기 바라며 본 문서를 변경하실 경우에는 원문과 변경된 내용을 표시하여 주시기 바랍니다. 원문과 변경된 문서에 대한 상업적 용도의 활용은 허용되지 않습니다. 본 문서에 오류가 있다고 판단될 경우 이슈로 등록해 주시면 적절한 조치를 취하도록 하겠습니다.

I. Introduction	1
II. Log Manager	2
1. Dependency	3
2. Configuration	4
2.1. Logging Policy File(log4j.xml)	4
2.2. web.xml	4
3. Resources	8

I.Introduction

Log Manager plugin은 Anyframe Log Manager와 연동하기 위한 Legacy Application에서 샘플 코드와 이를 활용하는데 필요한 참조 라이브러리들로 구성되어 있다.

Installation

Command 창에서 다음과 같이 명령어를 입력하여 logmanager plugin을 설치한다.

```
mvn anyframe:install -Dname=logmanager
```

해당 플러그인 설치 폴더의 src/main/resources 폴더로 이동하여 log4j.xml(or logback.xml) 파일을 연 후, MongoDBAppender 설정 부분에서 ip 정보와 port 정보등 개발환경(또는, 운영환경)에 맞게 수정해 주도록 한다.(기본적으로 hostname=localhost, port=27017로 변경한다.)

```
<appender class="org.anyframe.logmanager.log4j.MongoDbAppender" name="mongodb">
  <param name="hostname" value="localhost"/>
  <param name="port" value="27017"/>
  <param name="userName" value=""/>
  <param name="password" value=""/>
  <param name="databaseName" value="logs"/>
  <param name="collectionName" value="log4jlogs"/>
</appender>
```

```
<appender class="org.anyframe.logmanager.logback.MongoDbAppender" name="mongodb">
  <hostname>localhost</hostname>
  <port>27017</port>
  <userName/>
  <password/>
  <databaseName>logs</databaseName>
  <collectionName>logbacklogs</collectionName>
</appender>
```

installed(mvn anyframe:installed) 혹은 jetty:run(mvn clean jetty:run) command를 이용하여 설치 결과를 확인해볼 수 있다.

Plugin Name	Version Range
Core [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html]	2.0.0 > * > 1.4.0

II.Log Manager

Log Manager plugin은 Anyframe에서 제공하는 Log Manager를 통한 해당 Application의 로그 조회, 로그 분석, 로그 정책 변경 작업을 수행하고자 할 때, 이를 위해 Legacy Application에 정의해야하는 Log Appender와 Log Pattern Layout, ServletFilter등의 클래스를 제공한다. Anyframe의 Log Manager는 기본적으로 Mongo DB를 로그 저장소로 사용한다. 그러므로 이 때, Log4j(또는 Logback)의 Appender로 MongoDbAppender를 사용해야 하며, 보다 다양하고 직관적인 패턴으로 저장하고 보여주기 위해 Anyframe에서 확장한 Appender와 Pattern Layout을 사용하도록 한다. 기본적으로 Application에서 남기는 로그 이외에도 Client IP, Server Name, Application Name 을 남겨주며, 이외에도 남기고 싶은 사용자 정보 또는 입력 값을 Anyframe에서 제공하는 ServletFilter를 통해 간편하게 설정해 줄 수 있다.

1.Dependency

사용자 Application을 Anyframe Log Manager와 연동하기 위해서는 MongoDB-Driver, Anyframe Log Manager등의 Dependency를 추가해주도록 한다.

Name	Version	Required	
mongo-java-driver.jar	2.10.1	Y	
anyframe-logmanager.jar	1.1.0	Y	

Maven 기반의 프로젝트로 생성할 경우 아래와 같이 pom.xml에 Dependency를 추가해 주도록 한다.

```
<dependency>
  <groupId>org.mongodb</groupId>
  <artifactId>mongo-java-driver</artifactId>
  <version>2.10.1</version>
</dependency>
<dependency>
  <groupId>org.anyframe</groupId>
  <artifactId>anyframe-logmanager</artifactId>
  <version>1.1.0</version>
</dependency>
```

2.Configuration

Anyframe Log Manager와 연동하는 작업은 별도의 추가 코딩 없이 간단한 설정만으로 가능하다. Log Manager plugin에서 제공하는 기능 중 로그 조회 기능을 사용하기 위해서는 로그 저장소로 사용되는 MongoDB에 로그를 저장해야 하므로 log4j.xml(또는 logback.xml) 파일에 Appender를 추가 정의해야 한다. 또한, 로그 레벨 변경 기능은 WAS의 재기동없이 log4j.xml 파일을 reload 해야하므로 web.xml 파일에 MDCServletFilter를 정의해 준다.(Logback을 사용하는 경우에는 불필요) 이 밖에도 MDCServletFilter를 통해 프로젝트에서 정의한 Custom Data에 대해서 로그를 남길 수 있다.

2.1.Logging Policy File(log4j.xml)

먼저, Anyframe Log Manager는 로그 저장소로 MongoDB를 사용하므로 Legacy Application에서도 마찬가지로 로그 저장소로 MongoDB를 사용해야 한다. MongoDB와의 연결 정보를 정의하기 위해 아래와 같은 Parameter를 제공하며 정의하지 않을 시 Default Value로 등록된다.

Name	Default Value		
hostname	localhost		
port	27017		
databaseName	logs		
collectionName	log_repo		
userName	<input type="checkbox"/>		
password	<input type="checkbox"/>		

다음은 Anyframe Log Manager Plugin에서 정의한 log4j.xml 파일의 MongoDBAppender 정의 부분의 일부이다.

```
<appender class="org.anyframe.logmanager.log4j.MongoDbAppender" name="mongodb">
  <param name="hostname" value="localhost"/>
  <param name="port" value="27017"/>
  <param name="databaseName" value="logs"/>
  <param name="collectionName" value="moviefinderlogs"/>
  <param name="userName" value="admin"/>
  <param name="password" value="admin00"/>
</appender>
```

log4j.xml에 정의되어 있는 logger의 appender 값을 위에서 정의한 appender의 name인 "mongodb"로 지정해 주면 Property의 연결 정보를 기반으로 해당 MongoDB에 로그를 남길 수 있다.

```
<logger additivity="false" name="org.anyframe.plugin.logmanager">
  <level value="DEBUG"/>
  <appender-ref ref="mongodb"/>
</logger>
```

2.2.web.xml

Anyframe Log Manager에서는 로그 조회 기능 뿐만 아니라 로그 정책을 변경 할 수 있는 기능을 제공한다. 다시 말해서 Legacy Application의 log4j.xml 파일을 읽어와 로그 레벨을 변경 할 수 있다. 이 때, WAS 재기동 없이 변경된 로그 정책에 대해 적용시킬 수 있다. 이 또한, MDCServletFilter를 통해 해당 기능을 수행할 수 있다. 설정할 수 있는 속성들은 아래와 같다.

Name	Description	Default Value	
log4jFile	log4j 설정 파일의 이름	log4j.xml	

Name	Description	Default Value	
log4jDynamicReload	log4j.xml 파일의 Dynamic Reload 여부	true	
log4jDynamicReloadPeriod	log4j.xml 파일의 변경을 체크하는 주기 (milliseconds)	60000	

다음은 위에서 설명한 log4j.xml 파일의 Dynamic Reloading을 위해 설정한 MDCServletFilter의 정의 부분이다.

```
<filter>
  <filter-name>MDCServletFilter</filter-name>
  <filter-class>org.anyframe.logmanager.web.filter.MDCServletFilter</filter-class>
  <init-param>
    <param-name>log4jFile</param-name>
    <param-value>log4j.xml</param-value>
  </init-param>
  <init-param>
    <param-name>log4jDynamicReload</param-name>
    <param-value>true</param-value>
  </init-param>
  <init-param>
    <param-name>log4jDynamicReloadPeriod</param-name>
    <param-value>60000</param-value>
  </init-param>
</filter>
```

위와 같이 정의 하면 log4j.xml이라는 이름의 파일에 대해 1분 주기로 변경을 체크하게 된다.



Spring의 Log4jConfigListener

Spring에서는 log4j.xml 파일의 Dynamic Reloading 기능을 제공하기 위해 Log4jConfigListener를 제공한다. 위에서 설명한 MDCServletFilter를 통한 Dynamic Reloading도 가능하지만 Spring 기반의 Application일 경우 Spring의 Log4jConfigListener를 정의하여 적용하도록 한다.

```
<web-app...>
  <listener>
    <listener-class>org.springframework.web.util.Log4jConfigListener</
  listener-class>
  </listener>
  <context-param>
    <param-name>log4jRefreshInterval</param-name>
    <param-value>60000</param-value>
  </context-param>
  <context-param>
    <param-name>log4jConfigLocation</param-name>
    <param-value>classpath: log4j.xml</param-value>
  </context-param>
</web-app>
```

listener-class를 Log4jConfigListener로 정의하고 log4jConfigLocation으로 log4j.xml파일의 위치를, log4jRefreshInterval 속성을 정의함으로써 변경 체크 주기를 설정할 수 있다. 위의 코드처럼 설정할 경우 classpath의 log4j.xml에 대해 60초 주기로 변경을 체크하게 될 것이다.

프로젝트에서 로그로 남기고 싶은 정보가 있다면 web.xml 파일에 Anyframe의 MDCServletFilter의 sessionAttributes로 해당 데이터의 key 값을 정의하여 적용시킬 수 있다.

```
<web-app...>
```

```

<filter>
  <filter-name>MDCServletFilter</filter-name>
  <filter-class>org.anyframe.logmanager.web.filter.MDCServletFilter</filter-class>
  <init-param>
    <param-name>sessionAttributes</param-name>
    <param-value>userId</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>MDCServletFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
</web-app>

```

위와 같이 Filter를 정의하면 기본적으로 Application Name, Client IP, Server Name을 로그로 남길 수 있다. 이 때, log4j의 ConversionPattern에 %X{key}를 사용하여 출력할 수 있으며 각각의 key값은 appName, clientIp, serverName이 된다.

MDCServletFilter에서는 내부적으로 MDC를 통해 로그로 출력해줄 요소를 key, value쌍으로 put, remove하는 로직을 수행하고 있다. 이는 log4j(또는 logback)의 ConversionPattern에 %X{key}의 형태로 정의함으로써 MDC를 통해 ThreadLocal에 저장된 값을 로그로 출력할 수 있다. 이미 MDCServletFilter에 의해 MDC에 저장된 appName, clientIp, serverName이외에 출력해야할 정보가 있다면 아래와 같이 설정해줄 수 있다.

- Http Session에 객체 저장

로그로 출력해줄 데이터를 HttpSession에 저장한다.

```

@RequestMapping("/logmanagerLogin.do")
public String login(@Valid User user, BindingResult results, HttpSession session){
    if (results.hasErrors())
        return "/logmanager/login";
    session.setAttribute("userId", user.getUserId());
    return "redirect:/logmanagerMovieFinder.do?method=list";
}

```

- web.xml 파일을 열어 MDCServletFilter의 "sessionAttributes" parameter value로 MDC에 put 해줄 객체의 key 값을 정의해 준다.

```

<web-app...>
  <filter>
    <filter-name>MDCServletFilter</filter-name>
    <filter-class>org.anyframe.logmanager.web.filter.MDCServletFilter</filter-class>
    <init-param>
      <param-name>sessionAttributes</param-name>
      <param-value>userId</param-value>
    </init-param>
  </filter>
</web-app>

```

HttpSession에 저장되어있는 객체 중 여러개의 객체를 MDC에 put해줘야 한다면 구분자로 ","를 사용한다. 또한, 객체가 String이 아닌 멤버 변수의 getter와 setter를 가지는 도메인 객체이면 "."을 사용하여 해당 멤버 변수를 지정해 줄 수 있다.

```

<web-app...>
  <filter>
    <filter-name>MDCServletFilter</filter-name>
    <filter-class>org.anyframe.logmanager.web.filter.MDCServletFilter</filter-class>
    <init-param>
      <param-name>sessionAttributes</param-name>

```



```
<param-value>user.userId,user.userName</param-value>
</init-param>
</filter>
</web-app>
```



Logging Framework으로 Logback을 사용하는 경우

Logging Framework으로 logback [<http://logback.qos.ch>]을 사용하는 경우에는 위에서 언급한 로그 정책의 자동재반영을 기본적으로 지원하고 있기 때문에 log4jFile, log4jDynamicReload, log4jDynamicReloadPeriod의 세 가지 항목을 설정할 필요가 없다

Logback에서 로그 정책을 자동으로 재반영하고 싶은 경우에는 아래와 같이 logback.xml의 configuration 항목에 scan과 scanPeriod 속성을 정의하면 된다.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration scan="true" scanPeriod="60 seconds">

  <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <pattern>[%-5level] %d{yyyy-MM-dd HH:mm:ss} %logger %n%msg%n
    </pattern>
    </encoder>
  </appender>

  ...

  <root level="ERROR">
    <appender-ref ref="console"/>
  </root>

</configuration>
```

Anyframe에서는 Anyframe Logback plugin [<http://dev.anyframejava.org/docs/anyframe/plugin/optional/logback/1.1.0/reference/htmlsingle/logback.html>]을 통해 보다 상세한 사용법과 Library Dependency를 제공하고 있다.

Logback에 관한 보다 자세한 내용은 <http://logback.qos.ch>에서 확인할 수 있다.



Logging Framework 설치에 따른 Log4j와 Logback 지원 문제

Anyframe Log Manager plugin은 Log4j와 Logback에 대한 지원만을 담당할 뿐 해당 Logging Framework에 대한 Library는 제공하지 않습니다.

따라서 해당 프로젝트에 Logging plugin [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html#core_logging_part]이 설치되어 있는 경우에는 Log4j를, Logback plugin [<http://dev.anyframejava.org/docs/anyframe/plugin/optional/logback/1.0.3-SNAPSHOT/reference/htmlsingle/logback.html>]이 설치되어 있는 경우에는 Logback을 지원 하도록 동작합니다.

3.Resources

- 참고자료
 - Anyframe Log Manager [<http://dev.anyframejava.org/docs/logmanager/1.1.0/reference/htmlsingle/logmanager.html>]
 - MongoDB [<http://www.mongodb.org/>]