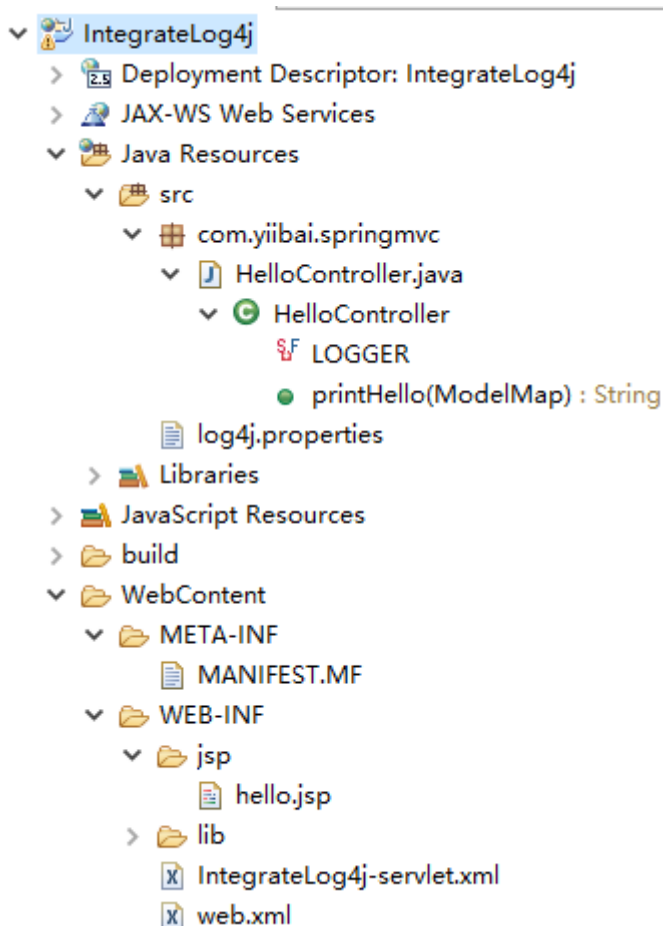


Spring MVC 集成 Log4j

以下示例显示如何使用 Spring Web MVC 框架集成 [LOG4J](#)。首先使用 Eclipse IDE，并按照以下步骤使用 Spring Web Framework 开发基于动态表单的 Web 应用程序：

1. 创建一个名称为 `IntegrateLog4j` 的动态 WEB 项目。
2. 在 `com.yiibai.springmvc` 包下创建一个 Java 类：`HelloController`。
3. 从 [maven](#) 存储库页面下载 Log4 库：[log4j](#)。把它放在 CLASSPATH 中。
4. 在 `src` 文件夹下创建一个 `log4j.properties` 文件。
5. 最后一步是创建所有源和配置文件的内容并运行应用程序，详细如下所述。

完整的项目文件目录结构如下所示 -



`HelloController.java` 的代码如下所示 -

```
package com.yiibai.springmvc;
import org.apache.log4j.Logger;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.ui.ModelMap;

@Controller
@RequestMapping("/hello")
public class HelloController{
    private static final Logger LOGGER = Logger.getLogger(HelloController.class);
    @RequestMapping(method = RequestMethod.GET)
    public String printHello(ModelMap model) {
        LOGGER.info("printHello started.");

        //logs debug message
        if(LOGGER.isDebugEnabled()) {
            LOGGER.debug("Inside:  printHello");
        }
        //logs exception
        LOGGER.error("Logging a sample exception", new Exception("Testing"));

        model.addAttribute("message", "Hello Spring MVC Framework!");
        LOGGER.info("printHello ended.");
        return "hello";
    }
}
Java
```

log4j.properties 的代码如下所示 -

```
# Root logger option
log4j.rootLogger=DEBUG, stdout, file

# 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
log4j.appender.file=org.apache.log4j.RollingFileAppender
#outputs to Tomcat home
log4j.appender.file.File=${catalina.home}/logs/myapp.log
log4j.appender.file.MaxFileSize=5MB
log4j.appender.file.MaxBackupIndex=10
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
```

Shell

IntegrateLog4j-servlet.xml 配置如下所示 -

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.0.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd">
  <context:component-scan base-package="com.yiibai.springmvc" />

  <bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/jsp/" />
    <property name="suffix" value=".jsp" />
  </bean>
</beans>
```

XML

hello.jsp 文件中的配置如下所示 -

```
<%@ page contentType="text/html; charset=UTF-8" %>
<html>
<head>
<title>Hello World</title>
</head>
<body>
  <h2>${message}</h2>
</body>
</html>
```

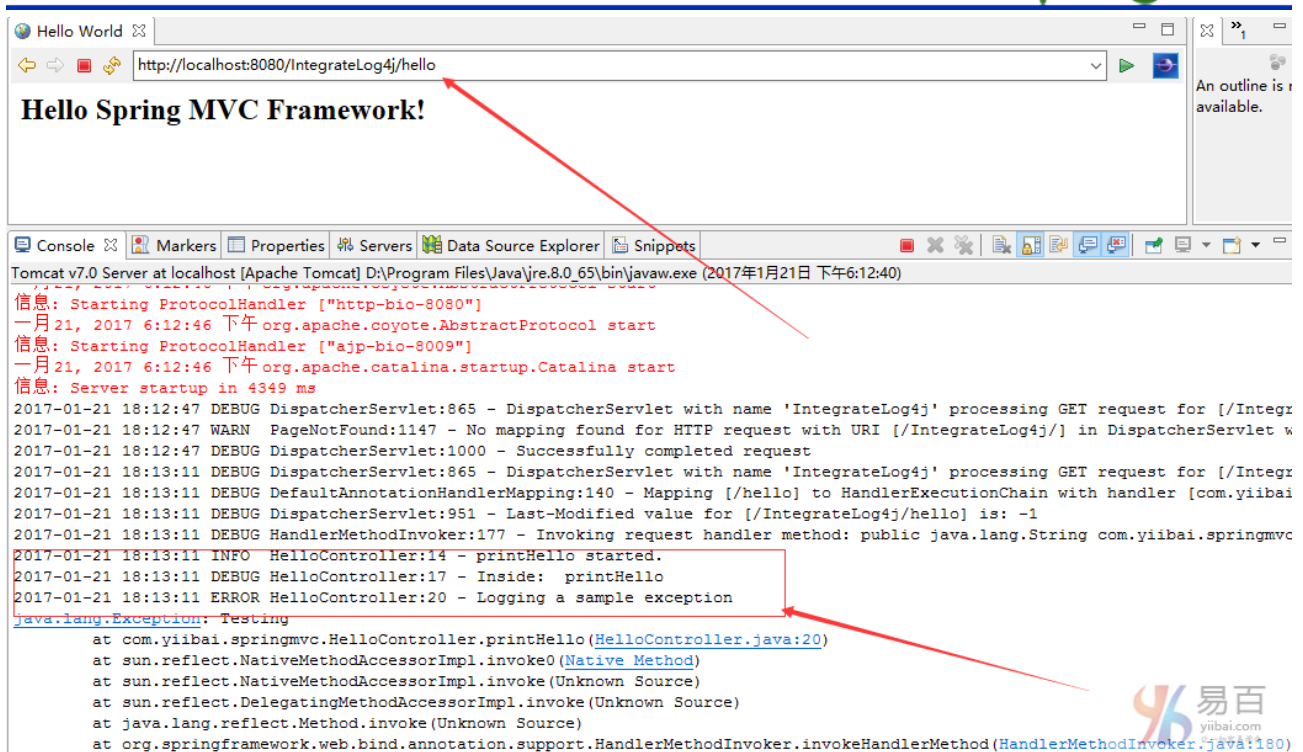
Jsp

在上面的代码中，已经在 tomcat 控制台中配置了 log4j，用它来记录日志详细信息，并且在 tomcat 目录下将日志文件保存为：myapp.log。

完成创建源和配置文件后，发布应用程序到 Tomcat 服务器。

现在启动 Tomcat 服务器，当访问 URL =>

<http://localhost:8080/IntegrateLog4j/hello> ， 如果 Spring Web 应用程序没有问题，应该看到以下结果：



Hello World  <http://localhost:8080/IntegrateLog4j/hello>

Hello Spring MVC Framework!

Tomcat v7.0 Server at localhost [Apache Tomcat] D:\Program Files\Java\jre.8.0_65\bin\javaw.exe (2017年1月21日 下午6:12:40)

信息: Starting ProtocolHandler ["http-bio-8080"]
一月 21, 2017 6:12:46 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["ajp-bio-8009"]
一月 21, 2017 6:12:46 下午 org.apache.catalina.startup.Catalina start
信息: Server startup in 4349 ms

2017-01-21 18:12:47 DEBUG DispatcherServlet:865 - DispatcherServlet with name 'IntegrateLog4j' processing GET request for [/Integr
2017-01-21 18:12:47 WARN PageNotFound:1147 - No mapping found for HTTP request with URI [/IntegrateLog4j/] in DispatcherServlet w
2017-01-21 18:12:47 DEBUG DispatcherServlet:1000 - Successfully completed request
2017-01-21 18:13:11 DEBUG DispatcherServlet:865 - DispatcherServlet with name 'IntegrateLog4j' processing GET request for [/Integr
2017-01-21 18:13:11 DEBUG DefaultAnnotationHandlerMapping:140 - Mapping [/hello] to HandlerExecutionChain with handler [com.yiibai
2017-01-21 18:13:11 DEBUG DispatcherServlet:951 - Last-Modified value for [/IntegrateLog4j/hello] is: -1
2017-01-21 18:13:11 DEBUG HandlerMethodInvoker:177 - Invoking request handler method: public java.lang.String com.yiibai.springmvc
2017-01-21 18:13:11 INFO HelloController:14 - printHello started.
2017-01-21 18:13:11 DEBUG HelloController:17 - Inside: printHello
2017-01-21 18:13:11 ERROR HelloController:20 - Logging a sample exception
java.lang.Exception: Testing
at com.yiibai.springmvc.HelloController.printHello([HelloController.java:20](#))
at sun.reflect.NativeMethodAccessorImpl.invoke0([Native Method](#))
at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
at java.lang.reflect.Method.invoke(Unknown Source)
at org.springframework.web.bind.annotation.support.HandlerMethodInvoker.invokeHandlerMethod([HandlerMethodInvoker.java:180](#))

 易百
yiibai.com