Unable to submit Spring boot java application to Spark cluster

Asked 9 years, 3 months ago Modified 8 years, 1 month ago Viewed 6k times

8

I have developed a web application with Spring Boot that uses Apache Spark for querying data from different datasources (like Oracle). At the beginning, I had planned to run the application without submitting it using the spark-submit script, but it looks like I cannot connect to the Master cluster without submitting a jar. I have successfully generated an uber jar which includes all the dependencies and sub-projects that I am using, but it seems that Spark does not like Spring Boot applications. When I try to submit the app, spark shows the following error:

```
Exception in thread "main" java.lang.IllegalArgumentException: LoggerFactory is
not a Logback LoggerContext but Logback is on the classpath. Either remove
Logback or the competing implementation (class org.slf4j.impl.Log4jLoggerFactory
loaded from file:/home/rojasmi1/spark/spark-1.4.0/assembly/target/scala-
2.10/spark-assembly-1.4.0-hadoop2.2.0.jar). If you are using Weblogic you will
need to add 'org.slf4j' to prefer-application-packages in WEB-INF/weblogic.xml
Object of class [org.slf4j.impl.Log4jLoggerFactory] must be an instance of class
ch.qos.logback.classic.LoggerContext
at org.springframework.util.Assert.isInstanceOf(Assert.java:339)
at
org.springframework.boot.logging.logback.LogbackLoggingSystem.getLoggerContext(Logba
at
org.springframework.boot.logging.logback.LogbackLoggingSystem.getLogger(LogbackLoggi
at
org.springframework.boot.logging.logback.LogbackLoggingSystem.beforeInitialize(Logba
at
org.springframework.boot.logging.LoggingApplicationListener.onApplicationStartedEver
at
org.springframework.boot.logging.LoggingApplicationListener.onApplicationEvent(Loggi
at
org.springframework.context.event.SimpleApplicationEventMulticaster.invokeListener(S
at
org.springframework.context.event.SimpleApplicationEventMulticaster.multicastEvent(S
at
org.springframework.boot.context.event.EventPublishingRunListener.publishEvent(Event
at
org.springframework.boot.context.event.EventPublishingRunListener.started(EventPubli
at org.springframework.boot.SpringApplication.run(SpringApplication.java:277)
at org.springframework.boot.SpringApplication.run(SpringApplication.java:957)
at org.springframework.boot.SpringApplication.run(SpringApplication.java:946)
at
ch.dlx.QubidaOracleConnectorApplication.main(QubidaOracleConnectorApplication.java:1
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43
at java.lang.reflect.Method.invoke(Method.java:497)
at
org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMain(Sp
at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:169)
at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:192)
at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:111)
```

```
at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
```

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

I have tried to exclude the slf4j-log4j12 dependency in the pom file, but I am still getting the same error.

The pom file contains the following configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>ch.dlx</groupId>
<artifactId>qubida-oracle-connector</artifactId>
<version>0.0.1-SNAPSHOT</version>

<name>qubida-oracle-connector</name>
<description></description>

<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <java.version>1.8</java.version>
</properties>

<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-dependencies</artifactId>
            <version>1.2.5.RELEASE</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>

<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
        <exclusions>
            <exclusion>
                <groupId>org.slf4j</groupId>
                <artifactId>log4j-over-slf4j</artifactId>
            </exclusion>

        </exclusions>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-tomcat</artifactId>
    </dependency>
```

```
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>

    <!-- Spark -->

    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-core_2.11</artifactId>
        <version>1.4.0</version>
        <scope>provided</scope>
        <exclusions>
                    <exclusion>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    </exclusion>
        </exclusions>
    </dependency>

    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-sql_2.11</artifactId>
        <version>1.4.0</version>
        <scope>provided</scope>
    </dependency>

    <dependency>
        <groupId>org.mongodb</groupId>
        <artifactId>mongo-hadoop-core</artifactId>
        <version>1.3.0</version>
        <exclusions>
            <exclusion>
                <groupId>org.slf4j</groupId>
                <artifactId>log4j-over-slf4j</artifactId>
            </exclusion>
        </exclusions>
    </dependency>

    <!-- DB Drivers -->

    <dependency>
        <groupId>com.oracle</groupId>
        <artifactId>ojdbc14</artifactId>
        <version>10.2.0.4.0</version>
    </dependency>


</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-shade-plugin</artifactId>
            <configuration>
                <createDependencyReducedPom>false</createDependencyReducedPom>

<keepDependenciesWithProvidedScope>true</keepDependenciesWithProvidedScope>
```

```
                <artifactSet>
                    <excludes>
                        <exclude>org.slf4j</exclude>
                    </excludes>
                </artifactSet>
            </configuration>
            <executions>
                <execution>
                    <phase>package</phase>
                    <goals>
                        <goal>shade</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
```

Is there a way to submit a Spring Boot application to the cluster? Should I use another type of project taking into account that I need to expose a RESTful API? Is there a way for connecting to the spark cluster without submitting the .jar?

Thanks in advance for your help.

java jar apache-spark spring-boot

Share Edit Follow

asked Jul 16, 2015 at 22:26

Stefan S

267 ● 2 ● 6 ● 13

3 Answers

Sorted by: Highest score (default)

4

I had a similar issue, for solving it try removing Spring Boot logging with the following exclusion:

```
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
        <exclusions>
            <exclusion>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-logging</artifactId>
            </exclusion>
        </exclusions>
    </dependency>
```

If you still get an error while initializing the servlet

> java.lang.NoSuchMethodError:
> javax.servlet.ServletContext.getVirtualServerName()Ljava/lang/String;

Then try using the 1.2.1.RELEASE version of the starter parent, since that is caused because of the servlet-api version used by the Spark Cluster.

Share Edit Follow

answered Jul 20, 2015 at 16:18

rojasmi
94 ● 4

This is exactly the solution for my problem. Hi 5! – Stefan S Jul 20, 2015 at 20:12

1

At build-time Spring Boot looks to see if you've included a particular logging implementation in your build and, if you haven't, uses Logback by default. Apparently Spark is adding Log4J to the classpath when running your application, which in turn causes a run-time error because Spring Boot now finds two logger implementions on the classpath: the one it included at build-time (Logback) and the one Spark is adding at run-time (Log4J).

If Spark provides a way to suppress the inclusion of Log4J at run-time, you could do that and just let Spring Boot wire in Logback by default.

If Spark is forcing Log4J on you, then the solution would be to explicitly include Log4J (not Logback) in your build so that Spring Boot will "see" it a build-time, and thus not include Logback.

EDIT: I should have checked my assumption by looking at the Spring Boot docs. You also have to explicitly exclude Log4J. See Spring Boot's Logging Docs.

Share Edit Follow

edited Jul 17, 2015 at 21:36

answered Jul 16, 2015 at 22:55

RichW
2,024 ● 2 ● 15 ● 25

Dear RichW. I followed the second approach you suggested (If Spark is forcing Log4J on you, then the solution would be to explicitly include Log4J (not Logback) in your build so that Spring Boot will "see" it a build-time, and thus not include Logback), but the problem unfortunately remains. This is how I explicitly added the Log4j depedency: `<dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> </dependency>` Do you know what else I could try? – Stefan S Jul 17, 2015 at 18:47

Sorry about that, Stefan, I goofed. I edited my answer. – RichW Jul 17, 2015 at 21:36

Spark supports log4j only. In order to force spring-boot to use log4j instead of logback by default, apply this procedure from spring-boot reference documentation but make sure to

0

change log4j2 to log4j and give it a version, e.g. 1.2.17. You will also need to put a log4j.properties file in `src/main/resources`. You can copy log4j.properties.template from Spark's /conf directory and rename it to log4j.properties.

Share Edit Follow

edited Sep 27, 2016 at 14:31

answered Sep 27, 2016 at 14:15

Daniel

1,656 ● 3 ● 21 ● 37