Title: Illegal character causes spring boot microservice build to fail during creation of docker image

Post Body:

I am establishing inter communication with Micro services in a Spring boot application. The response is generated in JSON format and logged in log file using logback.xml. This is further leveraged on ELK side with docker-maven plugins. While performing mvn clean install, it throws exception during creation of docker image through pom.xml

Error stack :

> Caused by: com.spotify.docker.client.exceptions.DockerException: com.spotify.docker.client.shaded.com.fasterxml.jackson.core.JsonParseException: Illegal character ((CTRL-CHAR, code 0)): only regular white space (\r, \n, \t) is allowed between tokens at [Source: (File); line: 1, column: 2] at com.spotify.docker.client.auth.ConfigFileRegistryAuthSupplier.authForBuild (ConfigFileRegistryAuthSupplier.java:108) at com.spotify.docker.client.auth.MultiRegistryAuthSupplier.authForBuild (MultiRegistryAuthSupplier.java:77) at com.spotify.docker.client.DefaultDockerClient.build (DefaultDockerClient.java:1483) at com.spotify.docker.client.DefaultDockerClient.build (DefaultDockerClient.java:1460) at com.spotify.plugin.dockerfile.BuildMojo.buildImage (BuildMojo.java:240) at com.spotify.plugin.dockerfile.BuildMojo.execute (BuildMojo.java:135) at com.spotify.plugin.dockerfile.AbstractDockerMojo.tryExecute (AbstractDockerMojo.java:265) at com.spotify.plugin.dockerfile.AbstractDockerMojo.execute (AbstractDockerMojo.java:254) at org.apache.maven.plugin.DefaultBuildPluginManager.executeMojo (DefaultBuildPluginManager.java:137) at org.apache.maven.lifecycle.internal.MojoExecutor.execute (MojoExecutor.java:210)

pom.xml:

```
<!-- Dockerfile from Spotify -->            <plugin>                <groupId>com.spotify</groupId>                <artifact
```

I debugged code and found the cause of error. This is caused in below method at line highlighted:

```
private List<Comment> findCommentsForFeed(Feeds feed) {         log.info("Finding comments of feed with id {}", feed.getId());
```

My controller looks like :

```
@RestController @RequiredArgsConstructor @RequestMapping(value = "/feeds", produces = MediaType.APPLICATION_JSON_VALUE) public
```

My logback.xml file for json output appears like:

```
<springProfile name="docker">      <appender name="jsonConsoleAppender"         class="ch.qos.logback.core.ConsoleAppender">
```

Accepted Answer: None

Highest Rated Answer:

I did some analysis and found the issue is because of Spring boot version. There has been changes in directory structure of Spring boot after the introduction of Springboot 2.3.0. I am using Spring boot version 2.3.3. I did some changes in dockerfile and it worked. Maven docker plugin successfully created the image.
Dockerfile :

```
FROM adoptopenjdk/openjdk14 as builder  WORKDIR application ARG JAR_FILE=target/*.jar COPY ${JAR_FILE} application.jar RUN jav
```