

Title: Docker Multimodule Maven Build - Copy All POM Files to Appropriate Directories

Post Body:

I am trying to Dockerize the maven build for my multi-module project. I have a nested file structure in which all modules for my project are contained in a `modules/` directory. As part of the build, I would like to cache the dependencies so the container does not need to pull the dependencies unless the POM files are updated (i.e. Java changes should not necessitate pulling the dependencies every time). I have this working currently by copying each POM into the appropriate directory in the build container using individual Docker commands for each file; however, I have roughly 10 modules, currently, and may add more in the future. I would like to avoid having to individually COPY each POM as modules are added. I see that COPY supports the wildcard syntax for the source file parameter, but simply cannot get this working. The command I currently have is:

```
COPY modules/*/pom.xml ./
```

but this does not seem to properly copy. I'm sure there is a way to COPY these files without having to specify a command for each one. What is the proper command for doing this? Thank you very much in advance for any help!

Accepted Answer: None

Highest Rated Answer:

Do you need to build all the sibling modules when building a specific one? If not, my solution was to copy only the required POMs, then alter the parent `pom.xml` after copying it in the container:

```
RUN mv pom.xml pom.xml.org RUN cat pom.xml.org | grep -vP '<module>(?!my-commons|my-module)' >> pom.xml RUN mvn -pl my-commons
```

By removing the other module references from the parent you can build your project as if it would have only the desired modules.