

6-2 | SpringBoot 应用的 Docker 容器化部署

maven 配置

```
Java
<build>
  <finalName>${artifactId}-docker</finalName>
  <plugins>
    <plugin>
      <groupId>com.spotify</groupId>
      <artifactId>docker-maven-plugin</artifactId>
      <version>1.2.0</version>
      <executions>
        <!-- 当 mvn 执行 install 操作的时候，执行 docker 的 build -->
        <execution>
          <id>build</id>
          <phase>install</phase>
          <goals>
            <goal>build</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
  <configuration>
    <imageTags>
      <imageTag>${project.version}</imageTag>
    </imageTags>
    <imageName>${project.build.finalName}</imageName>
    <!-- 指定 Dockerfile 文件的位置 -->
    <dockerDirectory>${project.basedir}/docker</dockerDirectory>
    <!-- 指定 jar 包路径，这里对应 Dockerfile 中复制 jar 包到 docker 容器指定目录配置，也可以写到 Dockerfile 中 -->
    <resources>
      <resource>
        <targetPath></targetPath>
        <!-- 将下边目录的内容，拷贝到 docker 镜像中 -->
      </resource>
    </resources>
  </configuration>
</build>
```

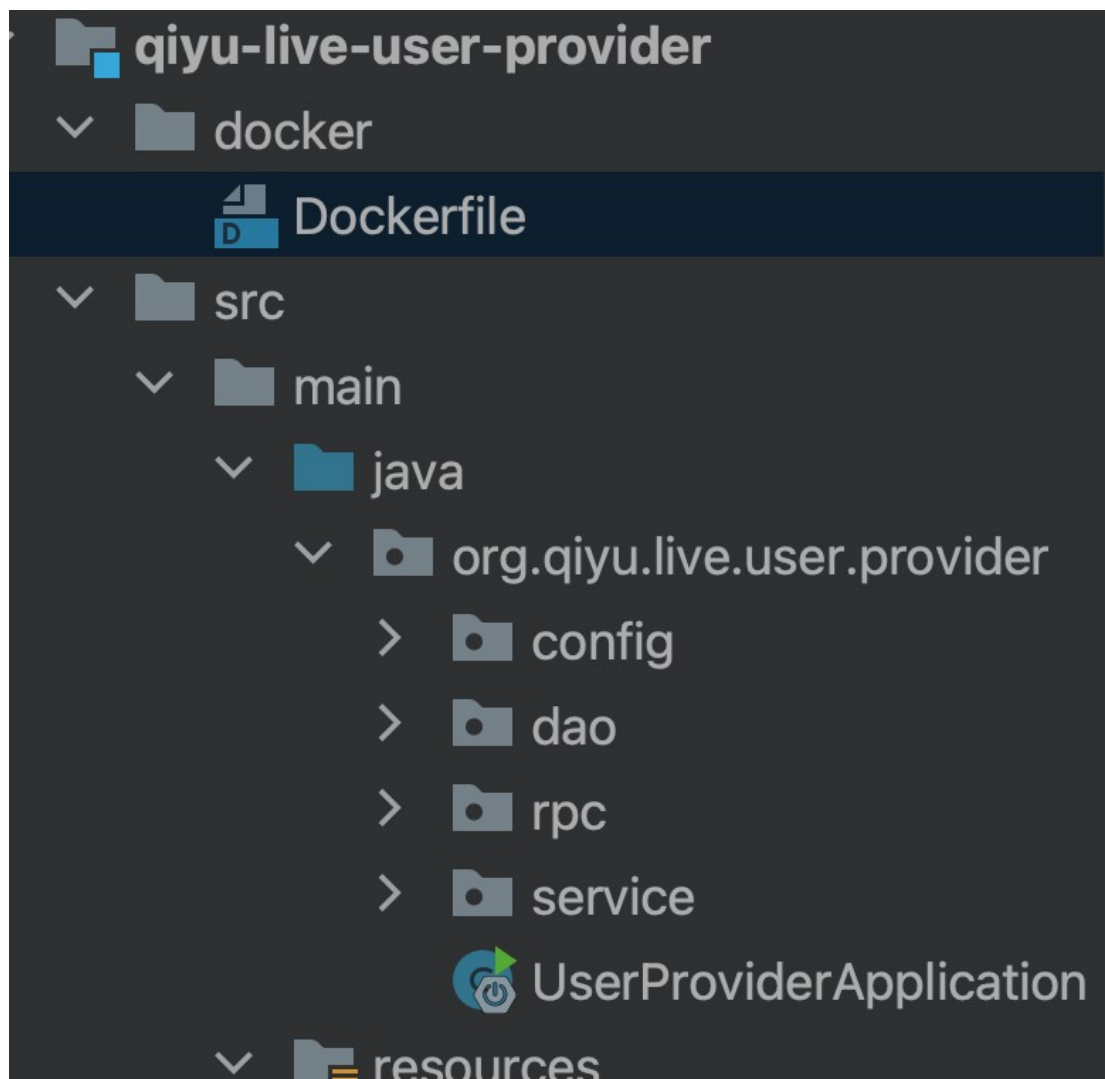
```
<directory>${project.build.directory}</directory>

<include>${project.build.finalName}.jar</include>
    </resource>
</resources>
</configuration>
</plugin>
<plugin>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
```

dockerfile 文件：

```
XML
FROM openjdk:17-jdk-alpine
VOLUME /tmp
ADD qiyu-live-user-provider-docker.jar app.jar
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]
```

具体位置如下：



命令执行部分

mvn 打包 Docker 镜像的命令：

```
Java  
mvn install
```

此时会把 docker 镜像安装在本地上。

在本地环境进行容器的启动操作：

```
XML  
docker run -p 9090:9090 --name qiyu-live-user-provider-01  
--add-host 'qiyu.nacos.com:相关 ip'  
--add-host 'cloud.db:相关 ip'  
--add-host 'qiyu.rmqs.com:相关 ip'  
registry.baidubce.com/qiyu-live-test/qiyu-live-user-provider-
```

docker:1.0.1