**6-2｜SpringBoot应用的Docker容器化部署**

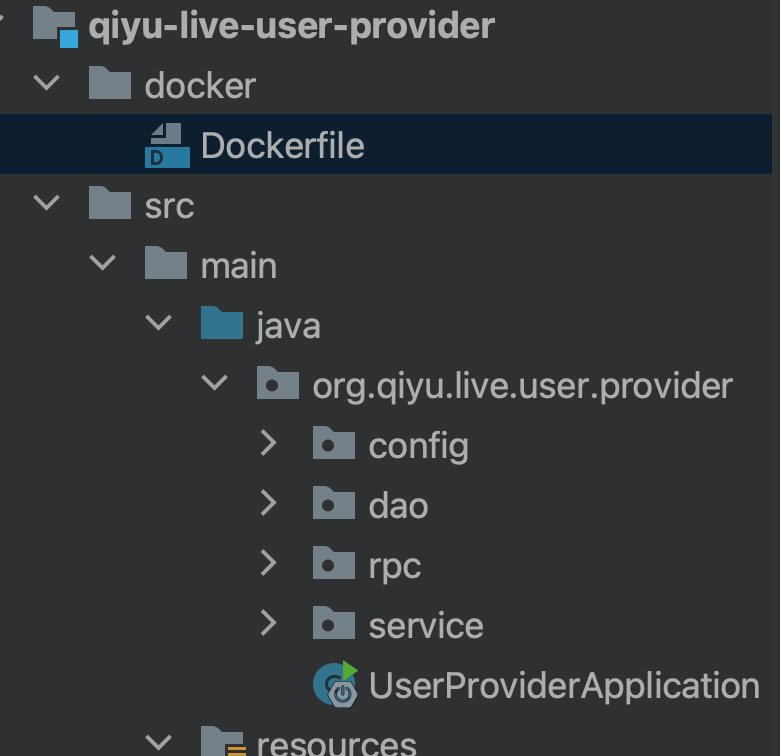
**maven配置**

|  |
| --- |
| Java <build>  <finalName>${artifactId}-docker</finalName>  <plugins>  <plugin>  <groupId>com.spotify</groupId>  <artifactId>docker-maven-plugin</artifactId>  <version>1.2.0</version>  <executions>  <!-- 当mvn执行install操作的时候，执行docker的build -->  <execution>  <id>build</id>  <phase>install</phase>  <goals>  <goal>build</goal>  </goals>  </execution>  </executions>  <configuration>  <imageTags>  <imageTag>${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 Docokerfile 中 -->  <resources>  <resource>  <targetPath>/</targetPath>  <!-- 将下边目录的内容，拷贝到docker镜像中 -->  <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 'hahhome:47.120.12.106' --add-host 'nacos.server:192.168.111.1' qiyu-live-user-provider-docker:1.0.2 |