

# 介绍

---

这篇文章主要简单介绍如何使用mysql的docker镜像在本地运行，并作为springboot应用的数据库。以及如何在springboot应用中连接该数据库。

## 1. 获取mysql的docker镜像

```
docker pull mysql
```

## 2. 运行mysql镜像

```
docker run --name mysql1 -e MYSQL_ROOT_PASSWORD=123456 -d mysql
```

## 3. 检查一下docker运行结果

```
docker ps
```

CONTAINER ID	IMAGE	COMMAND	C
1eb1b4e8496a	mysql	"docker-entrypoint.sh"	8

## 4. 进入mysql的容器

```
docker exec -i -t 1eb1b4e8496a /bin/bash  
root@1eb1b4e8496a:/#
```

1eb1b4e8496a 是上条中出现的 CONTAINER ID

## 5. 使用mysql命令创建数据库

```
mysql> mysql -uroot -p123456  
mysql> create database testdb;  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema|
| mysql             |
| performance_schema|
| sys               |
| testdb            |
+-----+
5 rows in set (0.00 sec)

mysql>
```

## 6. 打包自己程序的镜像

**gradle** buildDocker

检查下自己的Dockerfile之类的准备工作是否做好

## 7. 查看自己打包出来的镜像

```
docker images
REPOSITORY          TAG          IMAGE ID
umasuo/springboot-docker-mysql  latest      cb8010087d4f
```

## 8. 运行我们自己的容器

```
docker run -p 8080:8080 --name springboot-docker-mysql --link mysql
```

**注意** 再这里，我是用的 application.properties 是：

```
spring.datasource.url=jdbc:mysql://mysqlhost:3306/testdb
spring.datasource.username=root
spring.datasource.password=123456
```

```
spring.datasource.driverClassName=com.mysql.jdbc.Driver
spring.jpa.database=MYSQL
spring.jpa.hibernate.ddl-auto=update
spring.profiles.active=prod
```

其中可以看

到 `spring.datasource.url=jdbc:mysql://mysqlhost:3306/testdb` , `mysqlhost` 即是在运行我们自己的容器时置顶的mysql容器的名称。这里的其它配置都直接通过 `application.properties` build进了jar中, 不用再单独配置。

备注: 如果你有环境变量需要设置的, 请用-e添加。

## 9. 测试我们的程序

手动往数据库里面插入一条数据:

```
mysql> insert into test_table value(1,"kaka89","bruce Liu");
```

数据如下:

```
mysql> select * from test_table;
+----+-----+-----+
| id | name  | value  |
+----+-----+-----+
|  1 | kaka89 | bruce Liu |
+----+-----+-----+
1 row in set (0.00 sec)
```

测试

```
curl http://localhost:8080?id=1
```

10. 至此则检验完了.

平时可以使用 `docker --help` 等来查看相关命令的使用。

交流请看[博客主页](#)