# Java作业

zhanghui 2019.2.11

# 1. MySQL数据库

a.创建数据库 materialdb：

CREATE DATABASE materialdb;

b.创建表 material\_requirement\_plan：

DROP TABLE IF EXISTS `material\_requirement\_plan`;

CREATE TABLE `material\_requirement\_plan` (

`id` int(10) unsigned NOT NULL AUTO\_INCREMENT COMMENT '计划编号',

`material\_id` int(10) unsigned NOT NULL COMMENT '材料编号',

`material\_name` varchar(25) DEFAULT NULL COMMENT '材料名称',

`remark` varchar(255) DEFAULT NULL COMMENT '备注',

PRIMARY KEY (`id`)

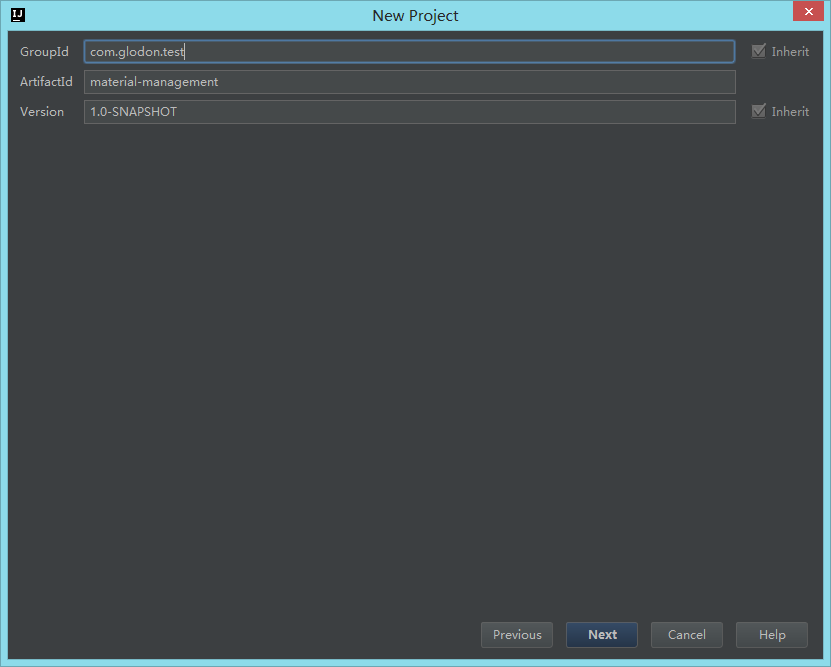
) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8;

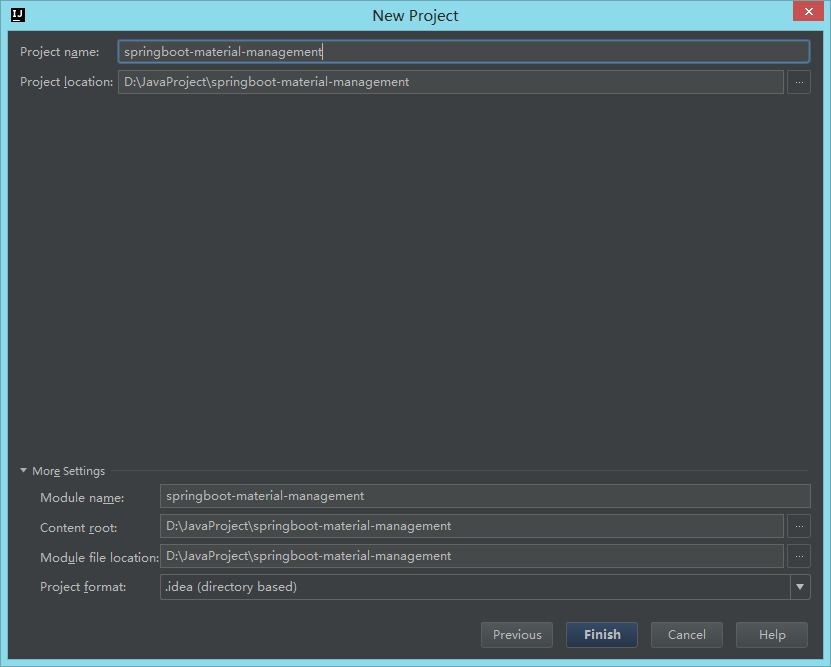
c.插入数据

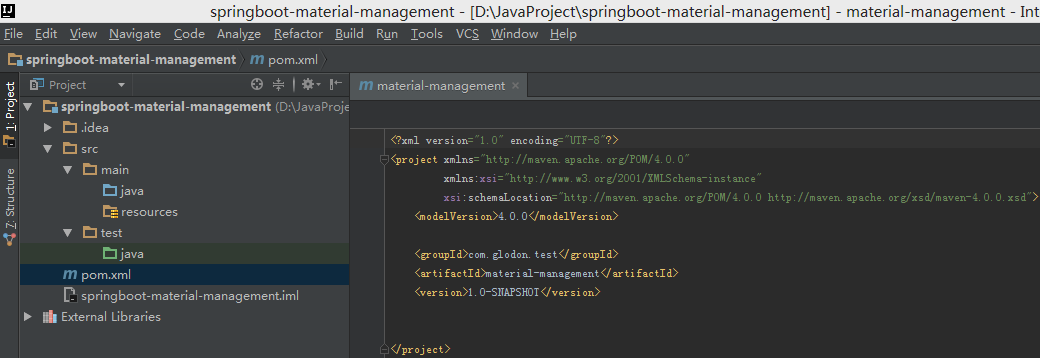
INSERT material\_requirement\_plan VALUES (1 ,1,'钢材','消耗材料的需要计划一');

# 2. 创建Maven项目

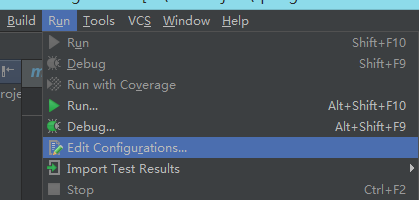
填写maven的坐标，“groupId”，“artifactId”，以及“version”，其中groupId一般采用公司域名的反写，而artifactId是项目名或模块名，而version就是该项目或模块所对应的版本号：

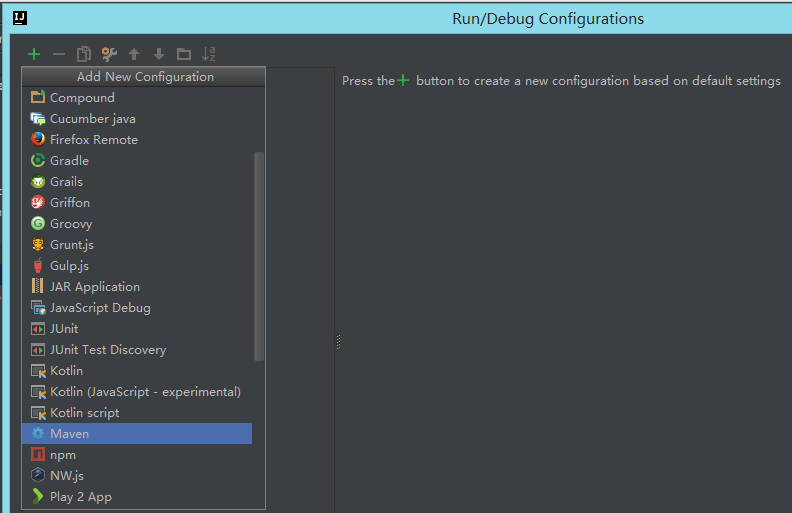


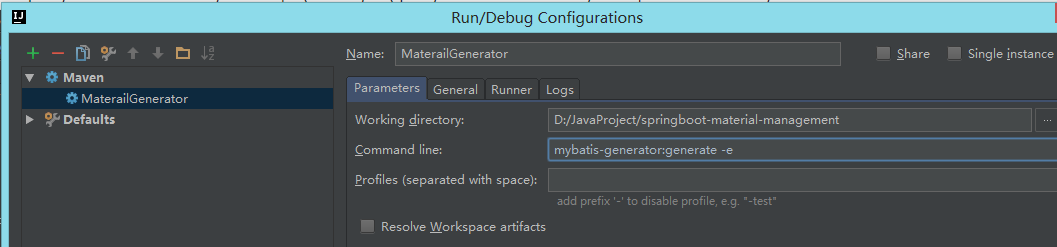




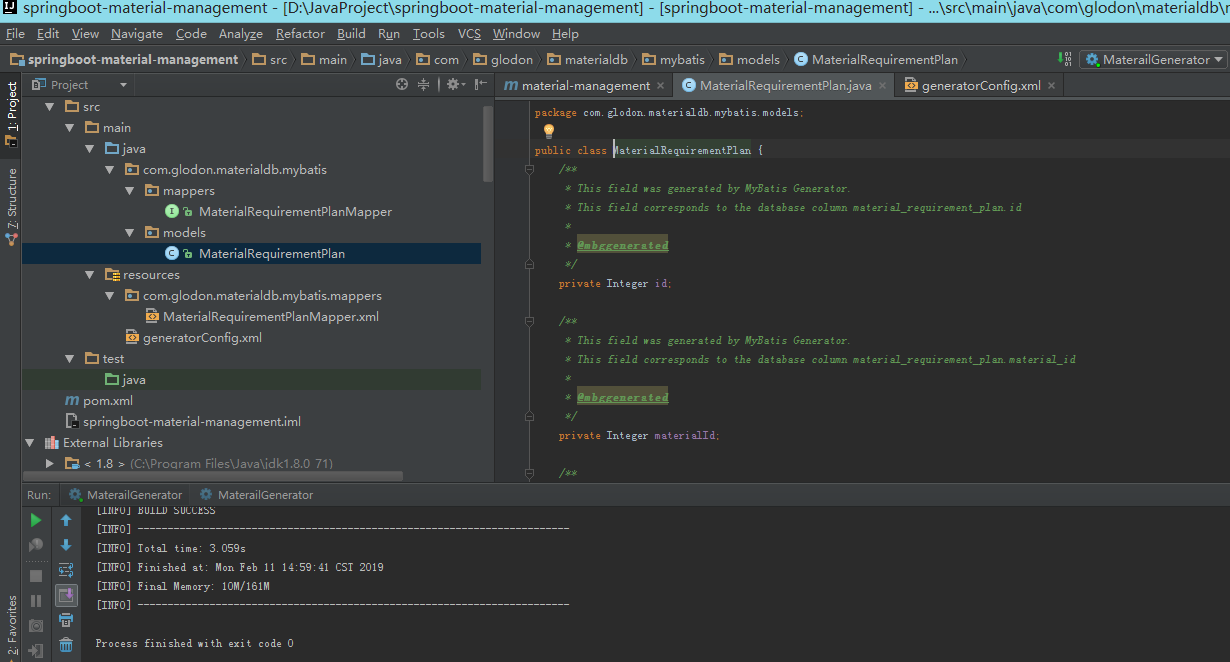
# 3. mybatis访问数据库







成功生成文件：



# 4. Spring.boot框架

springboot-material-management工程项目结构如下图所示：

com.glodon.materialdb.mybatis.controller - Controller 层

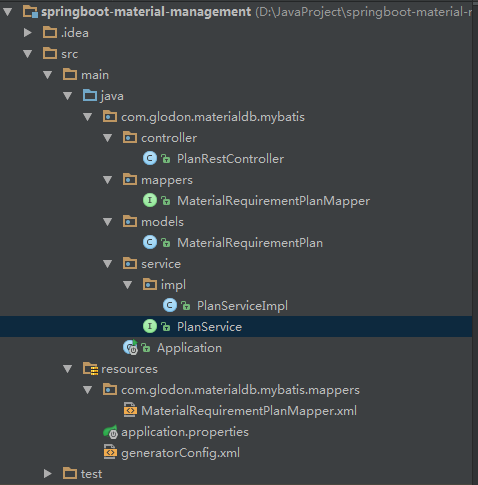
com.glodon.materialdb.mybatis.mappers - 数据操作层

com.glodon.materialdb.mybatis.models - 实体类

com.glodon.materialdb.mybatis.service - 业务逻辑层

Application - 应用启动类

application.properties - 应用配置文件，应用启动会自动读取配置



# 5. 使用redis缓存数据



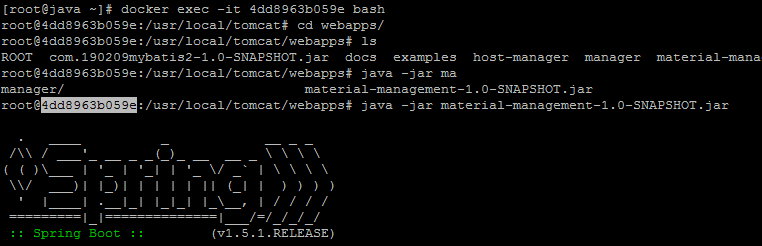
在浏览器中第一次输入<http://127.0.0.1:8080/api/plan/1>，服务端 Console 输出的日志：



后续再在浏览器中输入<http://127.0.0.1:8080/api/plan/1>，服务端 Console 输出的日志：



# 6. linux+docker+tomcat部署



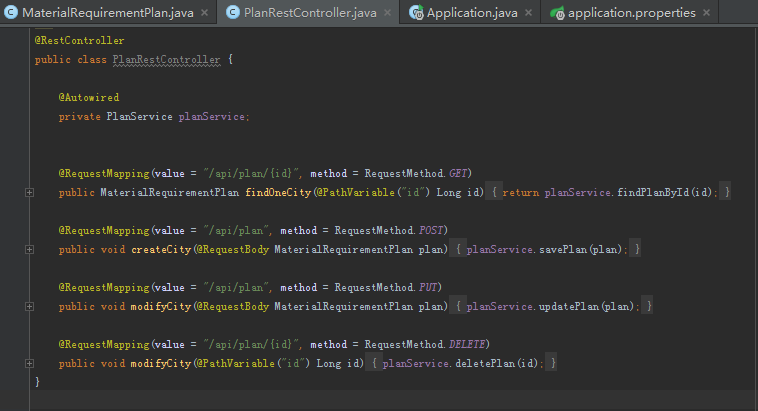
启动后在浏览器中访问



服务端 Console 输出的日志：



# 7. Springboot 实现 Restful 服务，基于 HTTP / JSON 传输



运行结果：

