# 关于日志：

日志：能够描述系统运行状态的所有时间都可以算作日志。

日志框架：一套能够实现日志输出的工具包。

日志框架的能力：定制输出目标、定制输出格式、日志携带的上下文信息、运行时的选择输出灵活的配置、优异的性能。



LogBack配置：

区分info和error。每天产生一个日志文件。

# Git拉取慕课网上源码：

git clone https://git.imooc.com/coding-117/coding-117.git

切换分支(首次)：git checkout –b 4-3 4-3

切换分支(非首次)：git checkout 4-3

切回master分支：git checkout master

# SpringBoot单元测试注解

SpringBoot提供了@SpringBootTest注解，当你需要SpringBoot特性时，它可以代替标准spring-test的@ContextConfiguration注解，这个注解通过测试类中创建ApplicationContext生效。除了@SpringBootTest注解，还有很多可以用来测试更具体slices的注解。

如果你在使用Junit4，别忘了添加@RunWith(SpringRunner.class)注解，否则注解会被忽略。同@RunWith(SpringJUnit4ClassRunner.class)

@RunWith是JUnit的一个注解, 用来告诉JUnit不要使用内置的方式进行单元测试, 而应该使用指定的类做单元测试 对于Spring单元测试总是要使用 SpringRunner.class。

@SpringBootTest 用来指定SpringBoot应用程序的入口类, 该注解默认会根据包名逐级往上找, 一直找到一个SpringBoot主程序class为止, 然后启动该类为单元测试准备Spring上下文环境。

# 技巧

1、在pojo中使用**@DynamicUpdate**注解，使得自动更新字段生效。

*2、<!--自动添加getter、settery以及toString的工具-->*

并在pojo类上添加@Data注解

<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-data-jpa</**artifactId**>  
</**dependency**>

3、单元测试在方法上添加注解**@Transactional**，自动回滚，保证了测试数据不会留在数据库。