1. **springboot基本知识：整合springmvc**

如果你想要保持Spring Boot 的一些默认MVC特征，同时又想自定义一些MVC配置（包括：拦截器，格式化器, 视图控制器、消息转换器 等等），你应该让一个类实现WebMvcConfigurer，并且添加@Configuration`注解，但是**千万不要**加@EnableWebMvc注解。如果你想要自定义HandlerMapping、HandlerAdapter、ExceptionResolver等组件，你可以创建一个WebMvcRegistrationsAdapter实例 来提供以上组件。如果你想要完全自定义SpringMVC，不保留SpringBoot提供的一切特征，你可以自己定义类并且添加@Configuration注解和@EnableWebMvc注解。

1. **整合jdbc（连接池）和事务、mybatis（见项目user-service2）、通用mapper（见项目user-service）**。

**spring官方连接池依赖和数据库驱动**（这里写的mysql驱动）

<dependency>  
     <groupId>org.springframework.boot</groupId>  
     <artifactId>spring-boot-starter-jdbc</artifactId>  
 </dependency>

<dependency>  
     <groupId>mysql</groupId>  
     <artifactId>mysql-connector-java</artifactId>  
 </dependency>

Springboot整合的连接池里面的**HikariCP应该是目前速度最快的连接池**。

# 连接四大参数  
 spring.datasource.url=jdbc:mysql://localhost:3306/heima  
 spring.datasource.username=root  
 spring.datasource.password=123  
 # 可省略，SpringBoot自动推断  
 spring.datasource.driverClassName=com.mysql.jdbc.Driver

如果是com.mysql.cj.jdbc.Driver需要自己写，与mysql驱动有关

**Druid连接池**：

<dependency>  
      <groupId>com.alibaba</groupId>  
      <artifactId>druid-spring-boot-starter</artifactId>  
     <version>1.1.6</version>  
 </dependency>

连接信息的配置与上面是类似的，只不过在连接池特有属性上，方式略有不同

#初始化连接数  
spring.datasource.druid.initial-size=1  
#最小空闲连接  
spring.datasource.druid.min-idle=1  
#最大活动连接  
spring.datasource.druid.max-active=20  
#获取连接时测试是否可用  
spring.datasource.druid.test-on-borrow=true  
#监控页面启动  
spring.datasource.druid.stat-view-servlet.allow=true

**SpringBoot中通过@Transactional注解来控制事务**：

@Service  
public class UserService {  
    @Autowired  
    private UserMapper userMapper;  
  
    public User queryById(Long id){  
        return this.userMapper.selectByPrimaryKey(id);  
    }  
  
    @Transactional  
    public void deleteById(Long id){  
        this.userMapper.deleteByPrimaryKey(id);  
    }  
}