1. Pom基础配置

*<!--起步依赖-->*<**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-parent</**artifactId**>  
 <**version**>1.5.2.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->*</**parent**>

<**dependencies**>  
 *<!--开发web项目相关依赖-->* <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-web</**artifactId**>  
 </**dependency**>  
 *<!--springboot单元测试-->* <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 *<!--configuration相关依赖-->* <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-configuration-processor</**artifactId**>  
 <**optional**>true</**optional**>  
 </**dependency**>  
  
</**dependencies**>

*<!--maven构建-->*<**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
</**build**>

1. 基础配置文件

Springboot默认读取resources下的application.properties application.yml

可以自定义配置文件一般放在resources/config下

自定义配置文件需要加入以下注解

@Component  
@PropertySource(**"classpath:config/my-web.properties"**)//配置文件资源路径  
@ConfigurationProperties(prefix = **"web"**)//前缀

配置文件类提示classpath 问题 可以加入*<!--configuration相关依赖-->*

1. 配置扫描包路径

Springboot默认扫描App.java同级及子包下的类如果App.java放在其他包中，需要扫描则可以用注解实现配置。

第一种：App类放在父包下面，所有有注解的类放在同一个下面或者其子包下面

第二种：指定要进行扫描注解的包URL，上面的问题的解决方案可以在App类上面加注解扫描mapper接口包下面的类。既在APP类中加入以下注解：

@ComponentScan(basePackages={**"com.config"**,**"com.test"**})

1. yml配置文件样例：

**server:  
 port:** 8080  
  
**spring:  
 datasource:  
 name:** test  
 **url:** jdbc:mysql://127.0.0.1:3306/talent  
 **username:** root  
 **password:** root  
 *# 使用druid数据源* **type:** com.alibaba.druid.pool.DruidDataSource  
 **driver-class-name:** com.mysql.jdbc.Driver  
 **filters:** stat  
 **maxActive:** 20  
 **initialSize:** 1  
 **maxWait:** 60000  
 **minIdle:** 1  
 **timeBetweenEvictionRunsMillis:** 60000  
 **minEvictableIdleTimeMillis:** 300000  
 **validationQuery:** select 'x'  
 **testWhileIdle:** true  
 **testOnBorrow:** false  
 **testOnReturn:** false  
 **poolPreparedStatements:** true  
 **maxOpenPreparedStatements:** 20  
**mybatis:  
 mapper-locations:** classpath:mapping/\*.xml  
 *#config-location: classpath:mybatis-config.xml* **type-aliases-package:** com.model  
  
*#pagehelper分页插件***pagehelper:  
 helperDialect:** mysql  
 **reasonable:** true  
 **supportMethodsArguments:** true  
 **params:** count=countSql

1. 整合mybatis

配置pom依赖关系

*<!-- mysql start-->*<**dependency**>  
 <**groupId**>mysql</**groupId**>  
 <**artifactId**>mysql-connector-java</**artifactId**>  
 <**version**>5.1.35</**version**>  
</**dependency**>  
*<!-- mysql end -->  
  
<!-- mybatis start -->*<**dependency**>  
 <**groupId**>org.mybatis.spring.boot</**groupId**>  
 <**artifactId**>mybatis-spring-boot-starter</**artifactId**>  
 <**version**>1.3.0</**version**>  
</**dependency**>  
*<!-- mybatis end -->*

*<!-- alibaba的druid数据库连接池 -->*<**dependency**>  
 <**groupId**>com.alibaba</**groupId**>  
 <**artifactId**>druid</**artifactId**>  
 <**version**>1.0.11</**version**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>com.fasterxml.jackson.core</**groupId**>  
 <**artifactId**>jackson-core</**artifactId**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>com.fasterxml.jackson.core</**groupId**>  
 <**artifactId**>jackson-databind</**artifactId**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>com.fasterxml.jackson.datatype</**groupId**>  
 <**artifactId**>jackson-datatype-joda</**artifactId**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>com.fasterxml.jackson.module</**groupId**>  
 <**artifactId**>jackson-module-parameter-names</**artifactId**>  
</**dependency**>  
*<!-- 分页插件 -->*<**dependency**>  
 <**groupId**>com.github.pagehelper</**groupId**>  
 <**artifactId**>pagehelper-spring-boot-starter</**artifactId**>  
 <**version**>1.1.2</**version**>  
</**dependency**>  
*<!-- alibaba的druid数据库连接池 -->*<**dependency**>  
 <**groupId**>com.alibaba</**groupId**>  
 <**artifactId**>druid-spring-boot-starter</**artifactId**>  
 <**version**>1.1.0</**version**>  
</**dependency**>

配置yml

配置数据源、mybatis的mapper配置文件路径和mybatis-config路径

**spring:  
 datasource:  
 name:** test  
 **url:** jdbc:mysql://127.0.0.1:3306/talent  
 **username:** root  
 **password:** root  
 *# 使用druid数据源* **type:** com.alibaba.druid.pool.DruidDataSource  
 **driver-class-name:** com.mysql.jdbc.Driver  
 **filters:** stat  
 **maxActive:** 20  
 **initialSize:** 1  
 **maxWait:** 60000  
 **minIdle:** 1  
 **timeBetweenEvictionRunsMillis:** 60000  
 **minEvictableIdleTimeMillis:** 300000  
 **validationQuery:** select 'x'  
 **testWhileIdle:** true  
 **testOnBorrow:** false  
 **testOnReturn:** false  
 **poolPreparedStatements:** true  
 **maxOpenPreparedStatements:** 20  
**mybatis:  
 mapper-locations:** classpath:mapping/\*.xml  
 *#config-location: classpath:mybatis-config.xml* **type-aliases-package:** com.model  
  
*#pagehelper分页插件***pagehelper:  
 helperDialect:** mysql  
 **reasonable:** true  
 **supportMethodsArguments:** true  
 **params:** count=countSql