传统ssm存在的问题

1. 依赖太多，存在版本问题
2. 配置太多（数据源，事务，扫描等待）
3. 部署太麻烦，需要tomcat

Springboot是什么

Springboot为快速启动且最小化配置的spring应用而设计的，使用特定的方式进行配置，从而使开发人员不在需要定义样板化的配置

简化spring应用的初始化搭建以及开发流程

Springboot的特性

1. 创建独立的spring应用
2. 直接嵌入tomcat，jetty等文本容器，不用不是war包
3. 提供固化的starter依赖，简化构件配置
4. 当满足条件时自动装配spring或第三方类库
5. 提供运维特性，比如指标信息，健康检查以及外部化配置
6. 不需要xml配置

创建独立运行的spring应用

Springboot使用嵌入式容器启动方式，嵌入式容器是应用的一部分，从本质上来说他是spring应用上下文的组件beans，这些组件和其他组件均由自动装配特性组装成springbean定义，随着spring应用上下文启动而注册初始化，启动spring应用上下文的核心组件就是springboot的SpringApplication，所以是独立的spring应用

创建springboot应用可执行的jar

添加spring-boot-maven-plugin到pom.xml文件中即可

<plugin>

<groupId>org.springframework.boot</groupId>

< artifactId >spring-boot-maven-plugin</artifactId>

</plugin>

在MANIFEST.MF文件中指定Main-Class，start-Class等信息

Main-Class: org.springframework.boot.loader.JarLauncher

Start-Class: com.tom.ControllerApp

Springboot应用的可执行jar文件被java –jar命令执行，java –jar命令引导的具体启动类必须配置在MANIFEST.MF的Main-Class中，项目的引导类定义在Start-Class属性中

可执行jar文件启动器-JarLauncher，执行jar的启动器，项目的引导莱尔被JarLauncher状态并执行

可执行war文件启动器-WarLauncher，执行war的启动器

创建springboot工程

打包在pom.xml文件目录下进入命令窗口，使用mvn clean package命令来打包