

- 1、**maven**本地仓库必要的、**mysql**必要的、**maven**项目必要，最终springboot打包为jar包
- 2、创建springboot项目有两种方式：
 - 1、直接创建一个maven项目的父结构
 - 2、用工具打包一个springboot的maven项目 <https://start.spring.io/>
- 3、在pom.xml里将springboot的父组件及其它两个引入

```
<parent>

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

    <artifactId>spring-boot-starter-parent</artifactId>

    <version>2.1.2.RELEASE</version>

</parent>

<dependencies>    ---->依赖关系

    <dependency>        ---->依赖

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

        <artifactId>spring-boot-starter-web    </artifactId>

    </dependency>

</dependencies>
```

4、引导类和启动springboot

可以启动springboot类上有 @SpringBootApplication 这个注解

能启动springboot的类有main方法并且main方法里

参数一：申明有注解的启动类是谁

参数二：是main方法的参数

```
SpringApplication.run(App.class, args);
```

```

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

/**
 * @SpringBootApplication 通过注解申明 该类为springBoot的引导类
 * 入口启动类
 */

@SpringBootApplication
public class App{

    public static void main(String[] args) {
        System.out.println("启动了");
        SpringApplication.run(App.class, args);
    }

}

```

5、启动模式

```

public static void main(String[] args) {
    /**
     * 隐藏banner启动方式
     */
    SpringApplication springApplication = new SpringApplication(Chapter33Application.class);
    //设置banner的模式为隐藏
    springApplication.setBannerMode(Banner.Mode.OFF);
    //启动springboot应用程序
    springApplication.run(args);

    //原启动方式
    /*SpringApplication.run(Chapter33Application.class, args);*/
}

```

做为了解没太大用

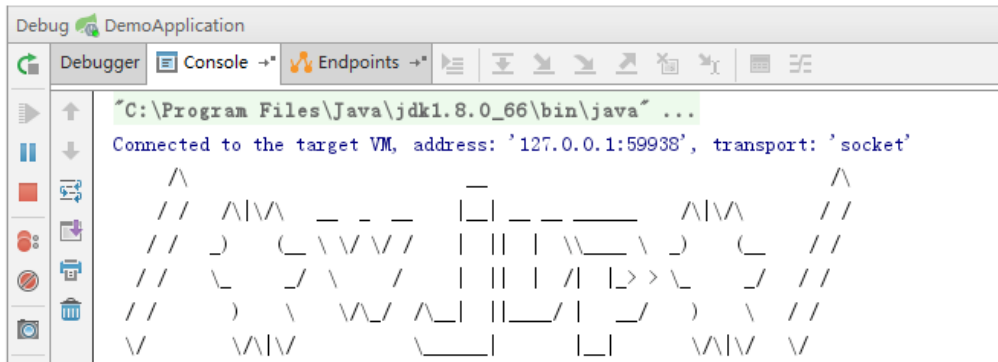
最后给个番外篇如何更改启动时显示的字符拼成的字母，就是更改下图标红框的地方

只需要在resources下新建一个txt文件就可以，命名为banner.txt，那这种字符该怎么拼出来呢，推荐一个网址，有这种工具，

链接传送

门：<http://patorjk.com/software/taag/#p=display&f=Graffiti&t=Type%20Something%20>

直接输入要生成的字母，系统会自动转换，然后复制下面转换好的字符到新建的banner.txt文件中，重新启动项目就可以了。



6、SpringBoot整合mybatis

第一步、在pom.xml里加入

```
<dependency>

    <groupId>org.mybatis.spring.boot</groupId>

    <artifactId>mybatis-spring-boot-starter</artifactId>

    <version>2.0.0</version>

</dependency>
```

第二大步骤：2、3、4（这三步可以通过插件完成

MyBatisCodeHelperPro）

2、编写javabean

3、写mapper接口

4、写mapper配置映射文件

5、配置数据源及端口、连接mysql

在main\resources下的 application.properties 文件配置

#配置访问端口

server.port=8081

#配置数据源

#连接最新版mysql的写法

```
spring.datasource.url = jdbc:mysql://127.0.0.1:3306/goodshome?
useUnicode=true&characterEncoding=utf-
8&serverTimezone=UTC&useSSL=false

spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver

spring.datasource.username=root

spring.datasource.password=123456
```

低版本配置方法

```
# spring.datasource.url=jdbc:mysql://127.0.0.1:3306/goodshome?
useSSL=true&useUnicode=true&characterEncoding=utf-8

# spring.datasource.driverClassName=com.mysql.jdbc.Driver

# spring.datasource.username=root

# spring.datasource.password=123456
```

#配置接口映射文件及javabean的所在目录

```
mybatis.mapper-locations=classpath:/mapper/*.xml

mybatis.type-aliases-package=com.xahhvk. *. domain

mybatis.configuration.call-setters-on-nulls=true
```

```
#配置访问端口
server.port=8081
```

```
#配置数据源
#连接最新版mysql的写法
spring.datasource.url = jdbc:mysql://127.0.0.1:3306/goodshome?useUnicode=true
&characterEncoding=utf-8&serverTimezone=UTC&useSSL=false
spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver
```

```
# 低版本配置方法
# spring.datasource.url=jdbc:mysql://192.168.10.33:3306/goodshome?useSSL=true&useUnicode=true&characterEncoding=utf-8
# spring.datasource.driverClassName=com.mysql.jdbc.Driver
spring.datasource.username=root
spring.datasource.password=123456
```

```
#配置接口映射文件及javabean的所在目录
mybatis.mapper-locations=classpath:/mapper/*.xml
mybatis.type-aliases-package=com.xahhvk. *. domain
mybatis.configuration.call-setters-on-nulls=true
```

