为什么要把SpringBoot打成war包

正常情况下SpringBoot项目是以jar包的形式，通过命令行：

java -jar demo.jar

来运行的，并且SpringBoot是内嵌Tomcat服务器，所以每次重新启动都是用的新的Tomcat服务器。正因如此，也出现了一些问题：

上传到项目的文件，如果是保存在项目中的，那么重启过后文件就会丢失。比如我们上传了一个头像，重启项目后，这个头像就没了。而如果将文件保存在本地磁盘中的话，html中标签没办法获取。打包内嵌了Tomcat服务器，打出来的包也比较大，往服务器上更新也比较慢。因此，我们才需要将SpringBoot项目打成war包，放到Tomcat中去运行。

**修改方法**

**1、在pom.xml文件中添加下面的依赖：**

<!--因配置外部TOMCAT 而配置-->

        <dependency>

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

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

            <scope>provided</scope>

        </dependency>

**2、将pom.xml文件首部的jar改成war**

    <groupId>com.star</groupId>

    <artifactId>yiyong</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <packaging>war</packaging>

    <!--<packaging>jar</packaging>-->

**3、最后，将启动类从**

@SpringBootApplication

public class YiyongApplication {

    public static void main(String[] args) {

        SpringApplication.run(YiyongApplication.class, args);

    }

}

修改为

@SpringBootApplication

public class YiyongApplication extends SpringBootServletInitializer{

    @Override

    protected SpringApplicationBuilder configure(SpringApplicationBuilder builder) {

        return builder.sources(YiyongApplication.class);

    }

    public static void main(String[] args) {

        SpringApplication.run(YiyongApplication.class, args);

    }

}

注意：该类继承了SpringBootServletInitializer并且重写了configure方法，这是关键所在。

**打包部署**

**1、IDEA**

在IDEA右侧Maven栏双击package等待Build Success即可。

**2、Eclipse**

在项目名称上单击鼠标右键，执行 Run As 》Maven Install

**部署**

然后把target目录下的war包放到tomcat的webapps目录下，启动tomcat，即可自动解压部署。

最后在浏览器中输入

http://localhost:[端口号]/[打包项目名]/

发布成功