开始

Servlet 3.0之后可以使用注解定义Servlet和过滤器，就无需在web部署描述符(web.xml)中建立Servlet/过滤器配置了，tomcat7.0以上版本支持Servlet 3.0。

注意: 打包的jdk版本要和tomcat容器运行的版本一至,否则不但无法自动注册,而且不报任何错误!!!

~~有时候1.8打包后没法运行,可尝试先用1.7进行打包部署~~

pom.xml

```
    <!-- ... -->
    <packaging>war</packaging>
    <!-- ... -->
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <!-- 这里指定打包的时候不再需要tomcat相关的包 -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-tomcat</artifactId>
            <scope>provided</scope>
        </dependency>
        <!-- ... -->
    </dependencies>
    <build>
        <plugins>
            <!-- maven打包的时候告诉maven不需要web.xml,否则有可能会报找不到web.xml错误 -->
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-war-plugin</artifactId>
                <version>2.4</version>
```

```
                <configuration>
                    <failOnMissingWebXml>false</failOnMissingWebXml>
                </configuration>
            </plugin>
        </plugins>
    </build>
```

Boot启动类

```
/**
 * Hello world!
 *
 */
@SpringBootApplication
public class App extends SpringBootServletInitializer
{
    public static void main( String[] args )
    {
        // 直接运行和Jar的启动方式(必须有)
        SpringApplication.run(App.class, args);
    }

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder
builder) {
        // war包启动时注入启动类，使用main方法进行启动
        return builder.sources(App.class);
    }
}
```

SpringBoot提供了一种以编码的方式初始化Web配置。通过继承SpringBootServletInitializer类 Spring Boot应用能够使用嵌入的Spring上下文来注册配置，这个Spring上下文是在容器初始化的时候创建的。