

## 14 结构化你的代码

### 14.1 使用默认包

如果一个类代码中不包含一个包声明，那么他将被考虑为在一个默认包中，这种情况不被推荐的，我们应该避免这种情况。如果你的Spring Boot 应用使用了@ComponentScan、@EntityScan或@SpringBootApplication注解，那么这种情况会造成很大问题，因为jar包中的每个类，都能被读取。

我们建议你遵循Java包命名规约，使用反向域名作为包名（例如，com.example.protect）。

### 14.2 main应用类放置

我们建议你就要把你的main应用类放置在一个根包里。@SpringBootApplication注解会被放在主类里，它隐式地为项目定义了一个基本的搜索包。例如，如果你正在写一个JPA应用，那么@SpringBootApplication注解将会被用来查找@Entity项。将main应用类放置在根包中也保证了注解能扫描到项目中所有的组件。

如果你不想使用@SpringBootApplication，你可以使用@EnableAutoConfiguration和@ComponentScan来代替。

这里有一个实例的模板：

```
com
+- example
    +- myproject
        +- Application.java
        |
        +- domain
        |   +- Customer.java
        |   +- CustomerRepository.java
        |
        +- service
        |   +-CustomerService.java
        |
        +- web
        |   +- CustomerController.java
```

这个Application.java 文件将会声明main方法，并被标注基础的@SpringBootApplication。

```
package com.example.myproject;

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

@SpringBootApplication
public class Application
{
    public static void main(String [] args)
    {
```

```
        SpringApplication.run(Application.class, args);  
    }  
}
```