Spring Boot不需要任何特定的代码布局即可工作。但是，有一些最佳做法会有所帮助。

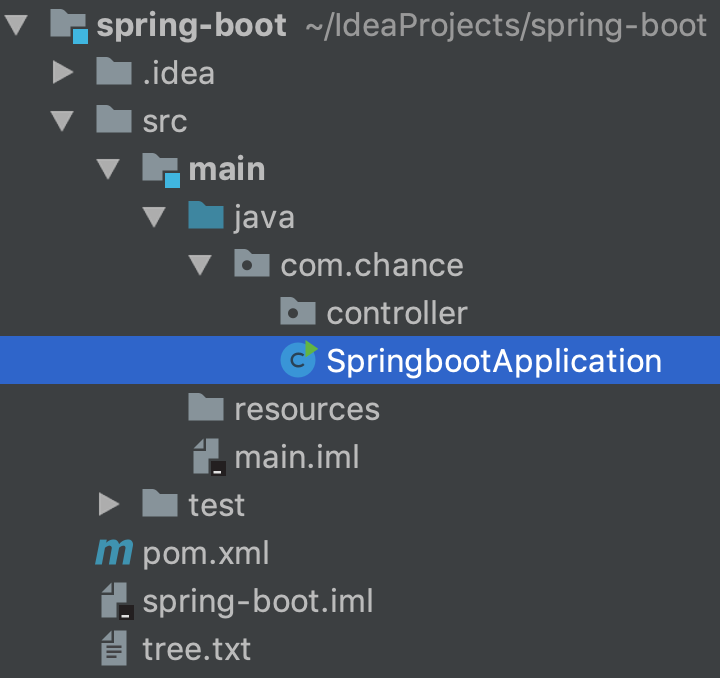
### 1、使用“default”包

当类不包含程序需包声明时，将其视为在“default package”中。通常不建议使用“default package”，应避免使用。对于使用@ComponentScan，@EntityScan或@SpringBootApplication注解的Spring Boot应用程序，这可能会导致特定的问题，因为每个jar中的每个类都会被读取。

建议：遵循Java建议的程序包命名约定，并使用反向域名（如com.chance.project）

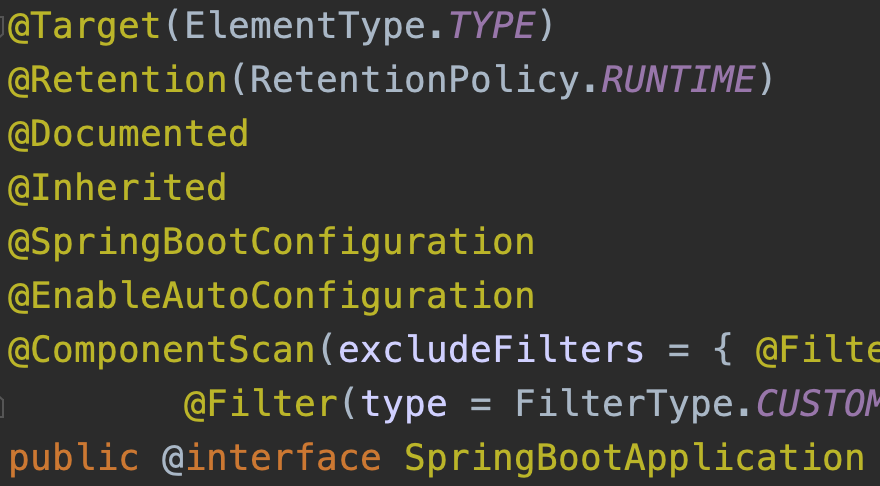
### 2、定位主应用程序类

建议将主应用程序类放在其他类之上的根包。



@SpringBootApplication注解通常放在主类上，为某些项目隐式定义了基本的“search package”。例如，你正在编写JPA应用程序，则使用@SpringBootApplication注解来搜索带@Entity注解的类。使用root package还允许在你的项目范围内进行组件扫描。

贴士：如果不想使用@SpringBootApplication，则它导入的@EnableAutoConfiguration和@ComponentScan注解定义了该注解，也可以使用它们替代。



以下显示了典型的布局：

com

+- example

+- myapplication

+- Application.java 主应用程序类

|

+- customer

| +- Customer.java

| +- CustomerController.java

| +- CustomerService.java

| +- CustomerRepository.java

|

+- order

+- Order.java

+- OrderController.java

+- OrderService.java

+- OrderRepository.java

Application.java文件声明main方法以及基本的@SpringBootApplication注解：

*@SpringBootApplication*

**public** **class** Application {

**public** **static** **void** main(String[] args) {

SpringApplication.run(Application.**class**, args);

}

}