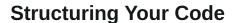
Structuring Your Code

Edit this Page

GitHub Project

Stack Overflow





Spring Boot does not require any specific code layout to work. However, there are some best practices that help.

**Ω** | TIP

If you wish to enforce a structure based on domains, take a look at Spring Modulith.

## Using the "default" Package

When a class does not include a package declaration, it is considered to be in the "default package". The use of the "default package" is generally discouraged and should be avoided. It can cause particular problems for Spring Boot applica-

Spring Boot / Reference / Developing with Spring Boot / Structuring Your Code

Q | TIP

We recommend that you follow Java's recommended package naming conventions and use a reversed domain name (for example, com.example.project).

## **Locating the Main Application Class**

We generally recommend that you locate your main application class in a root package above other classes. The @SpringBootApplication annotation is often placed on your main class, and it implicitly defines a base "search package" for certain items. For example, if you are writing a JPA application, the package of the @SpringBootApplication annotated class is used to search for <code>@Entity</code> items. Using a root package also allows component scan to apply only on your project.

Q | TIP

If you do not want to use @SpringBootApplication, the @EnableAutoConfiguration and @ComponentScan annotations that it imports defines that behavior so you can also use those instead.

The following listing shows a typical layout:

```
com
+- example
    +- myapplication
         -- MyApplication.java
        +- customer
           +- Customer.java
            +- CustomerController.java
            +- CustomerService.iava
            +- CustomerRepository.java
        +- order
            +- Order.java
            +- OrderController.java
            +- OrderService.java
            +- OrderRepository.java
```

The MyApplication.java file would declare the main method, along with the basic @SpringBootApplication, as

```
Java Kotlin
@SpringBootApplication
public class MyApplication {
    public static void main(String[] args) {
        SpringApplication.run(MyApplication.class, args);
```

< Build Systems Configuration Classes >

Spring Boot 3.5.0 Q Search CTRL + k

spring

Documentation

Community

System Requirements

Installing Spring Boot

Upgrading Spring Boot

- > Tutorials
- ➤ Developing with Spring Boot

**Build Systems** 

## Structuring Your Code

Configuration Classes

Auto-configuration

Spring Beans and Dependency

Using the @SpringBootApplication

Running Your Application

Developer Tools

Packaging Your Application for

- > Core Features
- > Web
- > Messaging
- > Testing > Packaging Spring Boot Applications
- > Production-ready Features
- > How-to Guides
- > Build Tool Plugins
- > Spring Boot CLI > Rest APIs
- > Java APIs
- > Specifications
- > Appendix