

19.运行程序

打包你的应用成为jar并且使用嵌入式HTTP 服务器的一个最大的特点是你可以在任何你想的地方运行你的程序。Debug Spring Boot应用也变得简单了，这意味着你不需要任何特殊的IDE插件或扩展。

本节仅介绍基于jar的打包。如果您选择将应用程序打包为war文件，则应参考您的服务器和IDE文档。

19.1 从IDE中运行

你可以将一个Spring Boot应用作为一个简单的Java应用从你的IDE里运行它。然而，首先你需要去导入你的项目。导入的步骤都会根据你的IDE不同而不同。大多数的IDE都能直接导入Maven项目，例如Eclipse用户能选择 **Import...->Existing Maven Projects** 从你的**File**菜单里。

如果你不能直接导入你的项目到IDE中，你可能需要去安装一个IDE插件。例如Eclipse和IDEA的Maven插件或Gradle插件。

如果你不小心运行一个应用程序运行了两次，那么你会看到一个**Port already in use**的错误。STS提供了一个**Relaunch**按钮，而不是**Run**按钮确保已经存在的实例被关上了。

19.2 作为一个打包的程序运行

如果你选择使用 Spring Boot Maven或者Gradle插件将应用程序打包成一个jar文件，那么你可以使用下面的指令去运行这个程序：

```
$ java -jar target/myproject-0.0.1-SNAPSHOT.jar
```

也可以在启用远程调试支持的情况下运行打包的应用程序。这使您可以将调试器附加到打包的应用程序中：

```
$ java -Xdebug -Xrunjdwp:server=y,transport=dt_socket,address=8000,suspend=n \  
-jar target/myproject-0.0.1-SNAPSHOT.jar
```

19.3 使用Maven插件

这个Spring Boot Maven插件包括一个**run**命令，这个命令可以用来编译和启动你的应用程序。

```
$ mvn spring-boot:run
```

你可能也想使用一些操作系统环境变量： `$ export MAVEN_OPTS=-Xmx1024m -XX:MaxPermSize=128M`

19.4 使用Gradle插件

Spring Boot Gradle插件也包含了一个**bootRun**指令，它可以被用来去使你的程序运行，这个**bootRun**指令可以在你引入**spring-boot-gradle-plugin**包后添加。

```
$ gradle bootRun
```

如果你想使用指定的操作系统参数：

```
$ export JAVA_OPTS=-Xmx1024m -XX:MaxPermSize=128M
```

19.5 热拔插

由于Spring Boot应用程序只是普通的Java应用程序，所以JVM热插拔应该可以开箱即用。JVM热交换在某种程度上受限于它可以替换的字节码，因此可以使用更完整的解决方案JRebel或Spring Loaded项目。spring-boot-devtools模块还包含对快速重新启动应用程序的支持。