

# java在线运行工具（教你创建一个普通的Java项目）

作者：有故事的人 来源：范文网 www.wtabcd.cn/fanwen/

本文原地址：<https://www.wtabcd.cn/fanwen/zuowen/d15945725a51f8242e6abeadfc36a1c3.html>

## 范文网，为你加油喝彩！

今天我们要介绍的是spring-native，它可以让你的spring boot程序，体验graalvm编译器的特性，把你的应用直接编译成native的！

不再需要安装jre，你的应用程序将和exe一样，直接在目标机器上运行！而且启动时间不到1秒钟。

要体验这个功能，我们从spring boot拿一个demo。

<https://start.spring.io/>

在这里选择这个实验性的功能springnative。下载下来之后，就可以使用m对老师的感谢语aven进行打包测试。

mvn spring-boot:build-imagegradle bootbuildimage

看一下这无敌的启动速度.... 0.038秒... 几乎是瞬时的！

神奇！

这一切，都得益于graalvm编译器。不过，你至少要把jdk升级到11才能用，也算是堆java8用户的一种别样的驱动吧。

当然，只有在2.4.5以后的springboot版本中，才支持spring native。

## graalvm是什么？

graalvm也是oracle的项目，它的代码地址是残酷月光 林宥嘉：

mvn spring-boot:build-imagegradle bootbuildimage

项目地址是：

[www.graalvm.org/docs/](http://www.graalvm.org/docs/)

graalvm是一个想要统一天下的虚拟机。因为它相比较与hotspotvm，还能够运行其他语言比如ruby，python，php等。

它是一个新的jvm，不同的是由于做了适配，它能够让不同的语言跑在同一个vm下面。

看看下面这张图，就知道graalvm的野心有多大。

这还没完，它最吸引人的地方就在于，它能够将应用代码，直接打包成native的二进制可执行代码，运行时连jvm都不需要了！

大家都知道，native和跑在vm里完全是两个档次，否则也不会有jit这么牛x的技术存在了。连android和ios都知道，native的应用流畅性比跑在monotouch上或者hybrid上高很多很多，对于追求性能的企业级应用来说，这个功能就更加实用一些。

让人惊讶的是，它为各个语言实现了一个可以沟通的桥梁。比如我看好的js中的某个库，不需要重新开发一个了，在java中直接就可以用。这是因为，graalvm开发了跨语言互操作协议，能保证跨语言的互操作性。中国人民大学校长

现在这个功能，大多数平台已经支持了。

```
public class example {    public static void main(string[] args) {        string str = "native image is awesome";        string reversed = reversestring(str);        system.out.println("t英语提分he reversed string is: " + reversed);    }    public static string reversestring(string str) {        if (str.isEmpty())            return str;        return reversestring(str.substring(1)) + str.charAt(0);    }}
```

通常情况下，我们直接这样运行，或者打包成jar包。

```
javac example.java
```

但我们还可以多一步，就是把class文件native化。

```
native-image example
```

执行的时候，只需要输入 ./example 就可以了。

## 有什么好处？

使用native编译的应用，可以实现秒级别的启动，运行更快，占用内存更小。它与主流的部署方式如微服务、京剧荀慧生k8s等，更加的切合。

但它与传统的jvm也有很多不同，主要体现在：

系统的性能分析会在编译阶段就给出没用的部分和代码将不会编译，直接会被移除，这得益于前

些java

版本的模块化需要提前对反射、资源和动态代理进行转换，没有类加载的延迟classpath在编译阶段固定class将不会被懒加载，回在启动的时候一股脑放到内存

虽然native有很多好处，但它的编译时间却很长，因为要做大量的代码静态分析，这也是所有native程序的通病吧。

end

这种thin jar的思路，是不是感觉java的发展越来越像golang了呢？docker镜像也会因为这种改变便得更小更纯粹，而脱离jre的java应用也越来越像一个真正的程序了。

但可惜的是，这种编译成native的思路虽然好，现阶段还是无法和golang相抗衡，主要还是在于编译器的差异上。

但愿graalvm能够继续发力，带java继续飞上几十年，养我三代子孙！

更多 作文 请访问 [https://www.wtabcd.cn/fanwen/list/92\\_0.html](https://www.wtabcd.cn/fanwen/list/92_0.html)

文章生成doc功能，由[范文网](#)开发