

الخطوة الأولى: إعداد بيئة التطوير

الهدف من هذه الخطوة هو إعداد بيئة التطوير الخاصة بك. سنحتاج إلى تثبيت Java 8.0 أو أحدث، وتثبيت Maven، وتثبيت Spring Boot. سنستخدم IntelliJ IDEA كمتكامل للكتابة.

خطوة 1: تثبيت Java 8.0 أو أحدث

1. طريقة أي أو تثبيت Java 8.0 أو أحدث. يمكنك تنزيله من الموقع الرسمي أو باستخدام مدير الحزم. سنستخدم مدير الحزم لتثبيت Java 8.0 أو أحدث.

تثبيت Java 8.0 أو أحدث

1. اذهب إلى الموقع الرسمي لـ Java 8.0 أو أحدث.

2. التالى: تثبيت Java 8.0 أو أحدث.

3. تثبيت Java 8.0 أو أحدث.

4. تثبيت Java 8.0 أو أحدث.

5. تثبيت Java 8.0 أو أحدث.

6. تثبيت Java 8.0 أو أحدث.

7. تثبيت Java 8.0 أو أحدث.

8. تثبيت Java 8.0 أو أحدث.

9. يمكنك التحقق من تثبيت Java 8.0 أو أحدث عن طريق تشغيل الأمر التالي:

10. يمكنك التحقق من تثبيت Java 8.0 أو أحدث عن طريق تشغيل الأمر التالي:

```
package com.example.demo;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
```

```
public class HelloController {
```

```
    @GetMapping("/")
```

```
    public String hello() {
```

```
        return "Hello World!";
```

```
    }
```

```
}
```

عادي. كنص! يمكنك التحقق من تثبيت Java 8.0 أو أحدث عن طريق تشغيل الأمر التالي:

البرامج التي تعمل على تشغيل التطبيقات التي تم إنشاؤها باستخدام Spring Boot، مثل خادم مع `src/main/java/com/example/demo/DemoApplication.java` لتطبيق كود، `src/main/resources` لملفات التطبيق، `src/main/webapp` لملفات الويب، `src/main/webapp/WEB-INF` لملفات الويب، `src/main/webapp/WEB-INF/classes` لملفات الكلاس، `src/main/webapp/WEB-INF/lib` لملفات المكتبات، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة.

البرامج التي تعمل على تشغيل التطبيقات التي تم إنشاؤها باستخدام Spring Boot، مثل خادم مع `src/main/java/com/example/demo/DemoApplication.java` لتطبيق كود، `src/main/resources` لملفات التطبيق، `src/main/webapp` لملفات الويب، `src/main/webapp/WEB-INF` لملفات الويب، `src/main/webapp/WEB-INF/classes` لملفات الكلاس، `src/main/webapp/WEB-INF/lib` لملفات المكتبات، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة.

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

@SpringBootApplication
public class DemoApplication extends SpringBootServletInitializer {

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(DemoApplication.class);
    }

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

البرامج التي تعمل على تشغيل التطبيقات التي تم إنشاؤها باستخدام Spring Boot، مثل خادم مع `src/main/java/com/example/demo/DemoApplication.java` لتطبيق كود، `src/main/resources` لملفات التطبيق، `src/main/webapp` لملفات الويب، `src/main/webapp/WEB-INF` لملفات الويب، `src/main/webapp/WEB-INF/classes` لملفات الكلاس، `src/main/webapp/WEB-INF/lib` لملفات المكتبات، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة.

البرامج التي تعمل على تشغيل التطبيقات التي تم إنشاؤها باستخدام Spring Boot، مثل خادم مع `src/main/java/com/example/demo/DemoApplication.java` لتطبيق كود، `src/main/resources` لملفات التطبيق، `src/main/webapp` لملفات الويب، `src/main/webapp/WEB-INF` لملفات الويب، `src/main/webapp/WEB-INF/classes` لملفات الكلاس، `src/main/webapp/WEB-INF/lib` لملفات المكتبات، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة.

```
<packaging>war</packaging>
```

البرامج التي تعمل على تشغيل التطبيقات التي تم إنشاؤها باستخدام Spring Boot، مثل خادم مع `src/main/java/com/example/demo/DemoApplication.java` لتطبيق كود، `src/main/resources` لملفات التطبيق، `src/main/webapp` لملفات الويب، `src/main/webapp/WEB-INF` لملفات الويب، `src/main/webapp/WEB-INF/classes` لملفات الكلاس، `src/main/webapp/WEB-INF/lib` لملفات المكتبات، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة، `src/main/webapp/WEB-INF/web.xml` لملف التهيئة.

```
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-web</artifactId>
```

```
</dependency>
```

تحت هذا:

```
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-tomcat</artifactId>
```

```

    <scope>provided</scope>
</dependency>

هذا: مثل الآن pom.xml في الاعتماديات قسم تبدو أن يجب
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-tomcat</artifactId>
        <scope>provided</scope>
    </dependency>
    <!-- spring-boot-starter-test -->
</dependencies>

```

3. `mvn clean package` باستخدام `mvn` ملف إلى التطبيق وتغليق بتجميع ق م `mvn` ملف بناء.

ب تنفذ: ق م `mvn` `target/demo-0.0.1-SNAPSHOT.war`، حيث `target` مجلد من التجميع: الأمر تنفذ

```
mvn clean package
```

هذا `target/demo-0.0.1-SNAPSHOT.war`، `target` مجلد في `target` ملف يخلق هذا

`mv` `target/demo-0.0.1-SNAPSHOT.war` `target/myapp.war` إلى `myapp.war` ملف تسمية بإعادة ق م نظافة، أكثر `mv` اختياري: `mv` ملف تسمية إعادة

```
mv target/demo-0.0.1-SNAPSHOT.war target/myapp.war
```

`demo-0.0.1-SNAPSHOT` من بدل `myapp` إلى السيق الجذري بسيط هذا

4. تلاقئنا. نشرًا يتيح م، `dropins` مجلد باستخدام `mv` على `dropins` ملف بنشر ق م `mv` على `dropins` ملف نشر.

`myServer` هو خدمتك واسم `/opt/ibm/wlp` في مثبتًا `dropins` لخدمتك `dropins` مجلد ايجد: `dropins` مجلد تحديدي
هو: الممسار إن

```
/opt/ibm/wlp/usr/servers/myServer/dropins
```

`dropins` مجلد إلى `dropins` ملف بنقل ق م: `dropins` ملف نسخ

```
cp target/myapp.war /opt/ibm/wlp/usr/servers/myServer/dropins/
```

ب إطلاقه: ق م يعمل، `dropins` ي كن لم إذا يعمل: `dropins` إذا الخادم إطلاق

```
/opt/ibm/wlp/bin/server start myServer
```

تلقائيًا. ملف ونشر سيكتشف فإنه بالفعل، يعمل كأن إذا

مثال: لم `audit` الأمامية الواجهة أو الخادم سجلات في بالبحث قم النشر: من التحقق

```
[AUDIT ] CWWKT0016I: Web application available (default_host): http://localhost:9080/myapp/
```

الواجهة في عرضها أو الخل في `/opt/ibm/wlp/usr/servers/myServer/logs/console.log` الوضع في سجلات `./server run myServer`. مع الأمامي `@GetMapping("/")` الوضع الأمامية

المتصفح. في المنشور `myapp` تطبيق اختبار التطبيق إلى الوصول 5.

إلى: اذهب متصفحك: افتح

`http://localhost:9080/myapp/`

`myapp` ل- `myapp` ل- الافتراضي الميناء هو 9080.
ملف اسم من السيق الجذر هو `/myapp`.
`@GetMapping("/")` في `myapp` مع يتطابق /

تري: أن يجب المتوقعة: النتيجة

Hello World!

عادي. لنص

ملاحظات

الحاجة. حسب `myapp` ملف تسمية بإعادة بتعديله قم. ملف اسم من مستمد `/myapp` السيق الجذر السيق: الجذر
لذلك. وفقًا `myapp` بتحديث قم مختلًا، ميناء يستخدم خادمك كأن إذا `myapp` ل- الافتراضي 9080 `myapp` يستخدم الميناء: رقم
`myapp` ل- 11 أو 8 `myapp` مثل متوافقًا إصدار تستخدم بك الخاصة التجديع وبئية `myapp` أن من تأكد: إصدار `myapp` 2.7.0.

`myapp` 2.7.0 `myapp` متطلبات مع تتوافق والتي، 4.0 `myapp` دعم تتضمن `javaee-8.0` الميزة: ميزات `myapp` 3.1.0.

الأخطاء: إصلاح

عن `/opt/ibm/wlp/usr/servers/myServer/logs/messages.log` في بالبحث قم الصفحة، تحميل يتم لم إذا
الأخطاء.

تعمل. وخدمتك `dropins` في `myapp` ملف أن من تأكد

ملخص

توسيع طريق عن `dropins`.
الوصول إلى `SpringServletInitializer` وتحتوي على `provided`.
ملف ونشر بناء.
الوصول إلى `http://localhost:9080/myapp/`.

على وظيفيًّا ولكن بسيطًا ويب تطبيقي يوفر مما، مميزات المرونة والتلقائي نشر قدرات من الانهـ هذا يستفيد
المستخدمين المستخدمين.