

# بسيط باسخدام Spring Boot تطبيق

باسخدام Spring Boot، يمكننا إنشاء تطبيق بسيط باستخدام Spring Boot. هذا المثال هذا يخلق لتوزيعه. Spring Boot يسهل على المبرمجين إنشاء تطبيقات Spring Boot. مع Spring Boot، يمكنك إنشاء تطبيق Spring Boot. Spring Boot يسهل على المبرمجين إنشاء تطبيقات Spring Boot. Spring Boot يسهل على المبرمجين إنشاء تطبيقات Spring Boot.

## خطوة خطوة دليل

### 1. إنشاء مشروع Spring Boot

لإنشاء هذا، سنستخدم Spring Boot. سنستخدم Spring Boot. سنستخدم Spring Boot.

#### المجلدات: بنية

```
SimpleServletApp/  
  pom.xml  
  src/  
    main/  
      java/  
        com/  
          example/  
            HelloServlet.java  
      webapp/  
        WEB-INF/  
          web.xml
```

الحاجة: حسب متطلباتها، لإنشاء الأمر هذا بتشغيل قم بإنشاء: Spring Boot مع إنشاء

```
mvn archetype:generate -DgroupId=com.example -DartifactId=simple-servlet-app -DarchetypeArtifactId=maven-archetype-webapp
```

الاقامة. الخطوات في ستعددها والتعليق، بنية يخلق هذا

### 2. كود كتابة

التالي: المحتوى مع src/main/java/com/example/ HelloServlet.java باسم ملف إنشاء

```
package com.example;
```

```
import javax.servlet.http.HttpServlet;
```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public class HelloServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {
        resp.setContentType("text/plain");
        resp.getWriter().write("Hello World!");
    }
}

```

الأسطر doGet طريقة يستخدم عادي. بنص! الأسطر بـ الأسطر لطلبات يستجيب الأسطر هذا شرح: الأسطر  
الأسطر web.xml تلوين مع لل توافق الـ تعليقات استخدام ويحب

### 3. التوزيع الوصف web.xml ملف إنشاء

الأسطر: الأسطر مع /WEB-INF/webapp/main/src في web.xml باسم ملف إنشاء

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0_
    version="4.0">
    <servlet>
        <servlet-name>HelloServlet</servlet-name>
        <servlet-class>com.example.HelloServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>
</web-app>

```

الأسطر نستخدم لا لأننا ضروري هذا /hello الأسطر نمط إلى ويخبرها HelloServlet في web.xml ملف يحدد شرح: الأسطر  
@WebServlet.

### 4. pom.xml ملف تلوين

الأسطر: الأسطر مع /SimpleServletApp في pom.xml تحديث أو إنشاء

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>
<artifactId>simple-servlet-app</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>war</packaging>

<properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
</properties>

<dependencies>
    <!-- Servlet API (provided by WLP) -->
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>javax.servlet-api</artifactId>
        <version>4.0.1</version>
        <scope>provided</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <!-- Maven WAR Plugin to build the WAR file -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-war-plugin</artifactId>
            <version>3.3.1</version>
            <configuration>
                <finalName>myapp</finalName>
            </configuration>
        </plugin>
        <!-- Liberty Maven Plugin for deployment -->
        <plugin>
            <groupId>io.openliberty.tools</groupId>
            <artifactId>liberty-maven-plugin</artifactId>

```

```

<version>3.3.4</version>
<configuration>
  <installDirectory>/opt/ibm/wlp</installDirectory>
  <serverName>myServer</serverName>
  <appsDirectory>dropins</appsDirectory>
  <looseApplication>false</looseApplication>
  <stripVersion>true</stripVersion>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

شرح:

- ويجب. لبرنامج war إلى packaging تعيّن يتم. و version, artifactId, groupId مع المشروع يحدد العمل ملامات:
- والموجهة. المصدّر لـ نسخة 8 يحدد الخصائص:
- التشغيل. وقت في يوفرها حيث، provided نطاق مع يضمن الاعتماديات:
- <finalName> باسم myapp.war إلى ملف اسم يحدد: myServer
- توزيعه و myServer الخادم اسم، /opt/ibm/wlp في يحدد: dropins مجلد إلى

## 5. المشروع بناء

البناء: باسم myServer ملف ببناء قم، SimpleServletApp/ من

```
mvn clean package
```

للتوزيع. ويعدّه، target/myapp.war إلى web.xml مع ويغلفه، يجمع هذا النتيجة:

## 6. تشغيله وتوزيعه

server.xml من تحقق مفعلة. servlet-4.0 الميزة مع تهئته تم قد myServer خادم أن من تأكد

```

<featureManager>
  <feature>servlet-4.0</feature>
</featureManager>

```

البناء: باسم myServer ملف ببناء قم، SimpleServletApp/ من

```
mvn liberty:run
```

## ❑ يحدث: ما

- ❑ بالفعّل. يعمل لكن لم إذا المقدمة في الخادم يبدأ
- ❑ تلقائيًا. dropins مجلد إلى myapp.war يوزع
- ❑ إيقافه. يتم حتى يعمل الخادم يظل

## ❑ مثل: سجل رسالة عن ابحت التوزيع: من تحقق

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

/opt/ibm/wlp/usr/servers/myServer/logs/console.log. في عادة سجلات

## 7. التطبيق إلى الوصول

إلى: واذهب ويب متصفح افتح

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

## ❑ الم توقع: الخروج

Hello World!

## ❑ التفاصيل:

- ❑ 9080 - ل. الافتراضي الميناء: 9080
- ❑ /myapp: ملف اسم من السيق الجذر myapp.war.
- ❑ /hello: من web.xml من مط

## 8. الخادم إيقاف

الواجهة. في Ctrl+C على بالضغط بإيقافه قم المقدمة، في الخادم يعمل mvn liberty:run لأن

## ملاحظات

### ❑ الأساسية: المتطلبات

- ❑ نظامك. على ومكونًا مثبتًا يكون أن يجب
- ❑ بتعديل قم myServer الخادم مثال هناك يكون أن يجب و /opt/ibm/wlp في مثبتًا يكون أن يجب installDirectory و pom.xml في serverName إذا web.xml إعدادك كان
- ❑ defaultServer أو /usr/local/wlp مثل مختلّفًا إعدادك كان إذا pom.xml في serverName و installDirectory
- ❑ server.xml في مفعلة 4.0-servlet ال ميزة تكون أن يجب

### ❑ بديل: توزيع

من فصل: بشركل وتوزيع لبناء

```
mvn clean package
```

```
mvn liberty:deploy
```

الأمر: لزم إذا ي دويّا الخادم بتشغيل قم

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

قم و `<httpEndpoint>` لـ `server.xml` من تحقيق مخططاً، `مينة` يستخدم `خادم` كان إذا `الميناء`: `تكوين`  
لذلك. وفقاً `بتعديل`

إلى الـ سيق الجذر لتغيّر `<finalName>app</finalName>` مثل `pom.xml` في `<finalName>` بتغيّر رقم الـ سيق: جذر  
/app.

□ الأخطاء: إصلا ح

❑ pom.xml في serverName و installDirectory من تحقق التوزيع، فشل إذا

الأخطاء. عن `opt/ibm/wlp/usr/servers/myServer/logs/messages.log` في السجلات من تحقق

maven.compiler. إعدادات مع متوافقًا أحدث، أو 8 استخداً من تأكيد

## ملخص

تكوين 2. web.xml و `HelloServlet.java` بسيط  `pom.xml`  مع  `pom.xml`  مشروع تكوين 1. كيفية: الدليل هذا يوضح  `myapp.war`  إلى التطبيق بناء 3.  `mvn clean package`  باستخدام  `mvn liberty:run`  باستخدام  `http://localhost:9080/myapp/hello`  إلى الوصول 5.  `mvn clean package`  وتشرغله وتوزيعه 4.  `http://localhost:9080/myapp/hello` !

تطبیق و توزیع لتطویر با استخدا م بسطانه ج ا هذای وفر