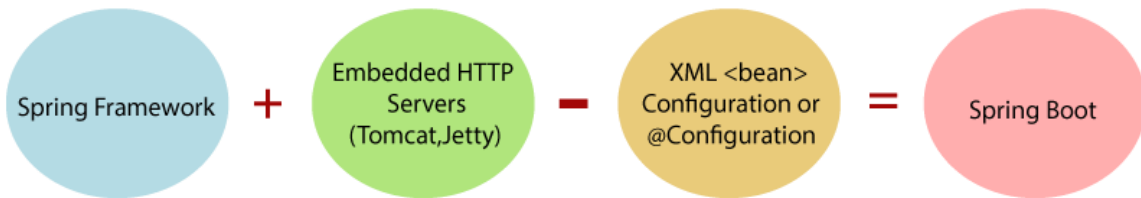


1. مقدمة في Spring Boot

سنتعرف في هذه المقدمة على مفاهيم أساسية وهي :

- منظم مكتبات Package Manager ووظيفته تنظيم و تثبيت المكتبات الخارجية.
- مصطلح RestApi وهو أسلوب أساسي لنقل البيانات على الشبكة يمكن التعرف عليه أكثر من هنا.
- ملف Json Object وهو طريقة لعرض البيانات وإرسال البيانات
- ملف XML وهو طريقة لترتيب وعرض البيانات .
- الفرق بين Spring Framework و Spring Boot .
- متطلبات التشغيل.
- تثبيت المتطلبات.
- بناء مشروع (Hello World) في إطار العمل Spring .

يساعدك Spring Boot على إنشاء تطبيقات قائمة بذاتها جاهزة للتشغيل وبشكل سريع، بحيث يمكنك البدء بأقل قدر من التعقيد، وتحتاج معظم تطبيقات Spring Boot إلى القليل جدًا من التخصيص لتكون جاهزة للعمل.



1.1 مميزات Spring Boot الأساسية هي:

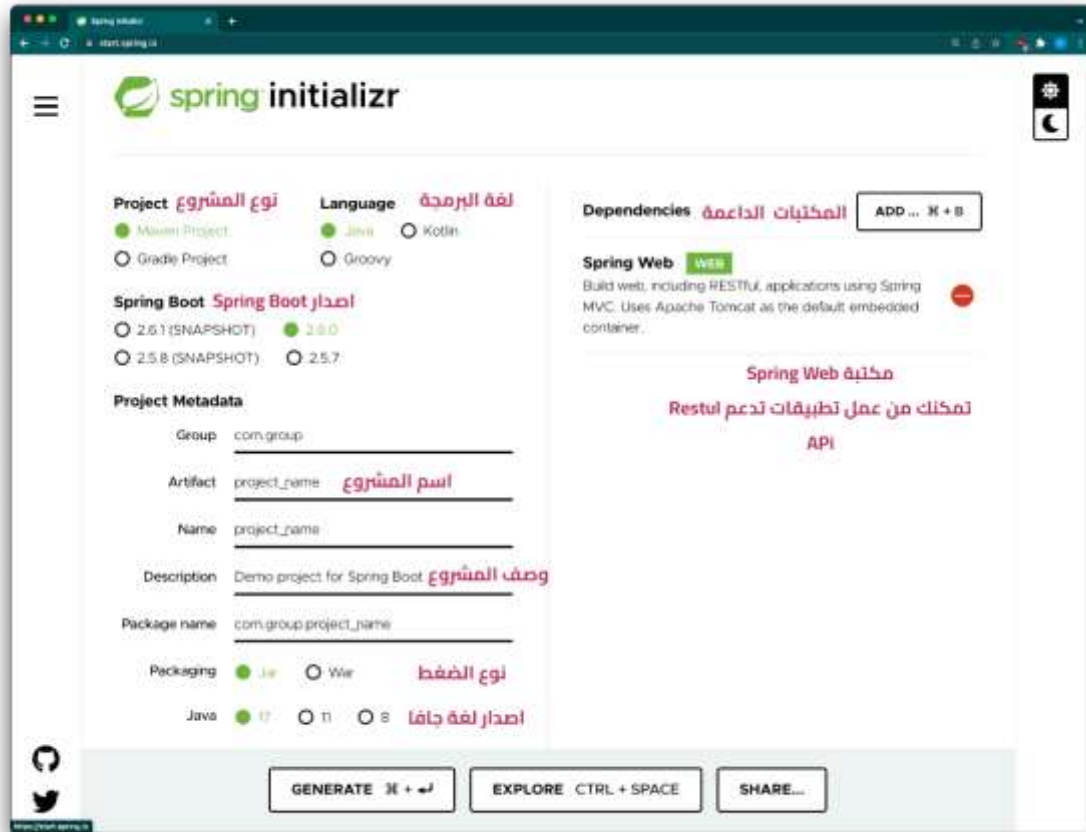
- توفير تجربة بدء أسرع بشكل جذري عن طريق استخدام Spring.io.
- يدعم أكثر من لغة مثل Kotlin و Java.
- توفير تجربة تطوير أسرع عن طريق الاستغناء عن بعض المتطلبات مثل استخدام ملفات XML. متطلبات التشغيل
- يتطلب استخدام Spring Boot تثبيت Java ويدعم حاليًا إصدارات Java 8 و Java 11 و Java 17.
- استخدام محرر نصوص، مثل VScode ويمكنك استخدام محررات أخرى مثل IntelliJ IDEA .
- تثبيت منظم مكتبات (Package Manager) للغة Java مثل Maven أو Gradle .

تثبيت متطلبات التشغيل

- نقوم بتثبيت Java عن طريق تثبيت Java Development Kit
- نقوم بتثبيت IntelliJ IDEA

1.2 بناء مشروع Spring Boot عن طريق Spring Boot Initializr

من أجل القيام بأول مشروع Spring Boot سوف نتجه إلى موقع Spring Boot Initializr والذي سيمكننا من إنشاء مشروع Spring Boot بشكل بسيط وسريع.

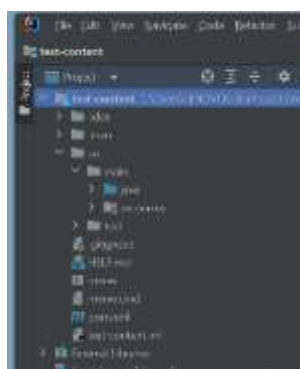


البيانات المهمة:

1. نوع المشروع
 - i. مشروع Maven: سوف ينشئ مشروع يحتوي على Maven كمنظم لملفات المشروع (اختيارنا).
 - ii. مشروع Gradle: سوف ينشئ مشروع يحتوي على Gradle كمنظم لملفات المشروع.
2. إصدار Spring Boot من الأفضل اختيار آخر نسخة مستقرة.
3. نوع الضغط
 - i. نوع الضغط Jar.
 - ii. نوع الضغط War.
4. إصدار لغة Java.
5. المكتبات الداعمة
 - i. قمنا بإضافة مكتبة Spring Web والتي من خلالها يمكننا بناء تطبيقات Web.

ملفات المشروع

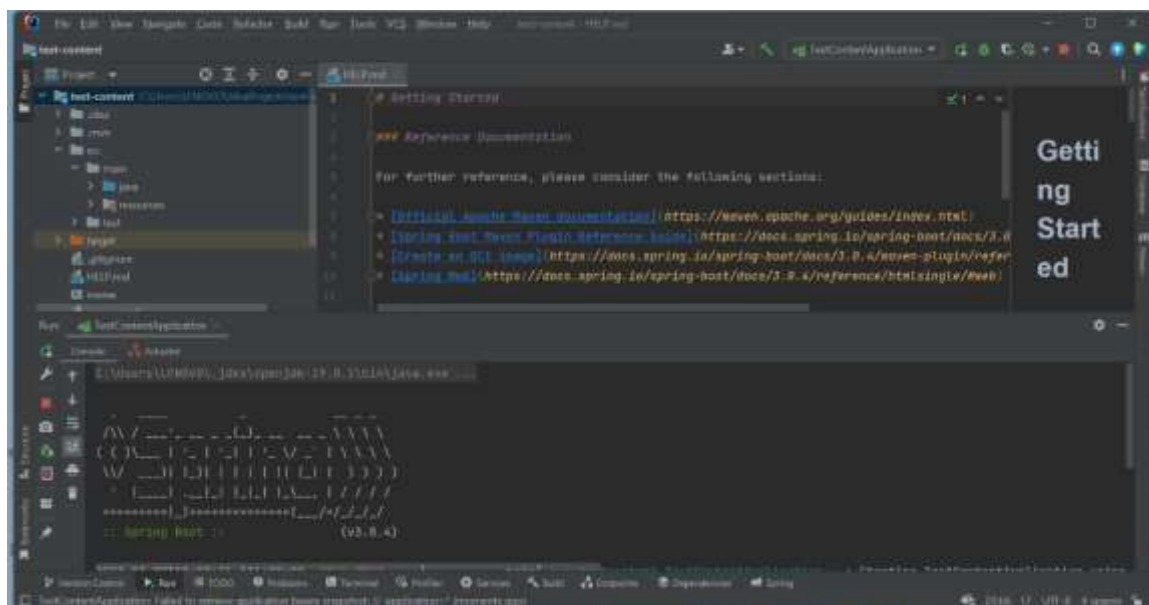
بعد تحميل الملف من موقع Spring Boot Initializr سوف نحصل على مشروع يحتوي على هذا الترتيب:



- ملف pom.xml هو الملف الخاص بإضافة المكتبات المساندة ويمكننا من خلاله إضافة مكتبات جديدة إلى مشروعنا.
- مجلد main هو المجلد المسؤول عن إضافة ملفات البرمجة الخاصة بالمشروع.

تشغيل المشروع

لتشغيل المشروع علينا الضغط على زر Run ، بعدها سيقوم maven بتشغيل المشروع



1.3 إضافة Hello World Api

في البداية سوف نتجه إلى ملف java داخل ال main ونقوم بإضافة التعديلات التالية :

```
package com.example.testcontent;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@RestController
public class TestContentApplication {
```

```

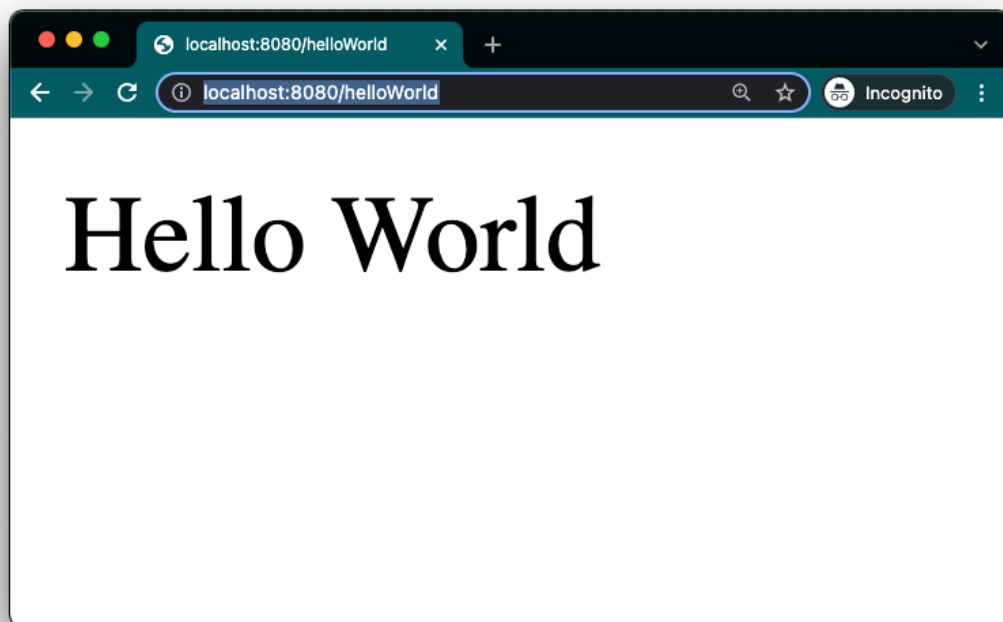
public static void main(String[] args) {
    SpringApplication.run(TestContentApplication.class, args);
}

@GetMapping("/hello-world")
public String helloWorld() {
    return "Hello world!!";
}
}

```

- قمنا بإضافة Annotaion باسم Rest Controller وهو المسؤول عن تحديد ما إذا كان هذا Class لديه خواص Controller مثل استقبال الطلبات Request، وسنتعرف عليه بشكل أكبر لاحقًا.
- قمنا بإضافة Annotaion باسم GetMapping ثم قمنا بإضافة method اسمها helloWorld وظيفته GetMapping أنها تبين أن method عبارة عن getRequest ويمكننا الوصول إليها عن طريق هذا المسار "helloWorld/"

والآن سنقوم بتشغيل المشروع مرة أخرى ونتجه إلى الرابط التالي في المتصفح، وهو رابط localhost؛ وذلك لأن المشروع يعمل على مستضيف من نفس الجهاز : <http://localhost:8080/helloWorld>



1.4 إيجابيات Spring boot مقارنة ب Spring

- يقوم بإنشاء تطبيقات Spring قائمة بذاتها يمكن البدء باستخدامها Java-.Jar
- لا توجد متطلبات لتكوين XML
- يوفر أداة CLI لتطوير واختبار تطبيق Spring Boot
- يزيد من الإنتاجية ويقلل من وقت التطوير.