Guía Técnica - Bookstore

## Introducción

La aplicación Bookstore consiste en una simple interfaz REST para una compañía que desea publicar su librería en la web. Esta aplicación le permite a la empresa listar los libros que tiene en una base de datos y sus contenidos (paginas) en formato texto y HTML. Fue desarrollada en Java utilizando Spring boot, Maven, REST Web Services, SQL y Hibernate con el objetivo de mantener la sencillez y su fácil ejecución como un simple archivo JAR.

## Requisitos

Para ejecutar correctamente, se necesita tener instalado previamente:

* Java versión 1.8 en adelante
* MySQL 5+
* Apache Maven versión 3.6.0
* IntelliJ o Eclipse (opcional)

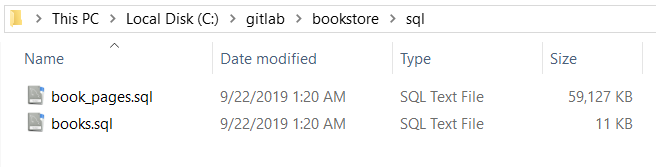
## Instalación y Ejecución

Para instalar la aplicación se tienen que realizar una serie de pasos ya que involucra varios software/sistemas juntos:

1. Ejecutar los scripts de la base de datos.
2. Actualizar los parámetros de conexión de la DB.
3. Construir el proyecto usando Maven.
4. Lanzar la aplicación.

### Paso 1 – Ejecutar los scripts de la base de datos

Los archivos de la base de datos están bajo la carpeta sql dentro de la carpeta raíz proyecto bookstore.

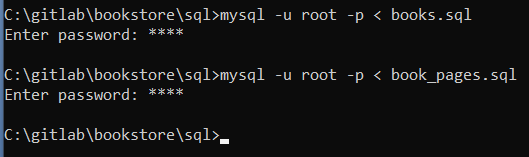


*Figura 1 – La carpeta* ***sql*** *esta justo en la raíz del proyecto. Esta contiene dos archivos de SQL (books.sql y book\_pages.sql).*

Para ejecutar los scripts, navegue hasta la carpeta donde se encuentran y ejecute los siguientes comandos:

* mysql -u <user> -p < books.sql
* mysql -u <user> -p < book\_pages.sql

Sustituya <user> por el usuario deseado con permiso de crear base de datos (schemas). Para ambos casos el programa le pedirá insertar la password para el usuario digitado. Ver ejemplo más abajo.



*Figura 2 – Ejecución de los scripts de base de datos. Tenga pendiente utilizar un usuario con permiso de crear base de datos, así también como tablas y otros DDL. En este caso se utilizo el súper usuario root.*

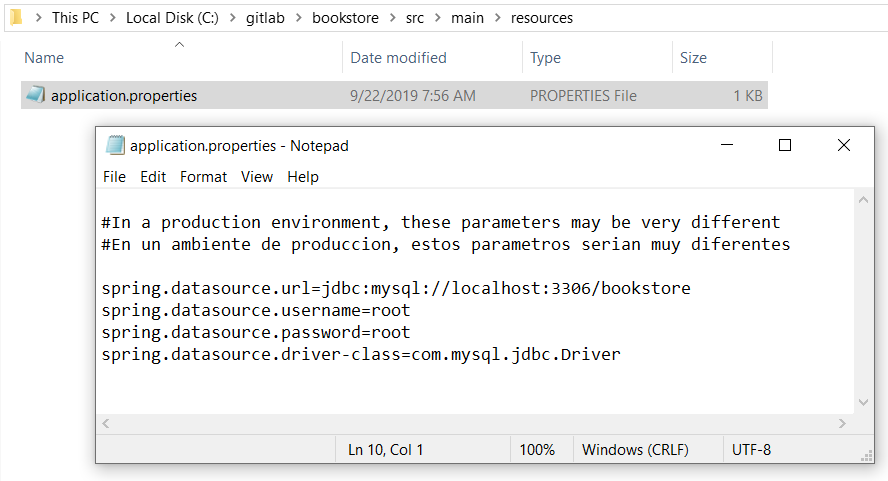
### Paso 2 – Actualizar los parámetros de la conexión de la DB

Una vez ejecutado correctamente los scripts de la base de datos, se tendrá que actualizar los parámetros de conexión para que la aplicación pueda acceder a esta. Se recomienda utilizar los mismos usuario/contraseña que se digitaron en el paso anterior.

Dentro de la carpeta raíz del proyecto bookstore, navege hasta ubicar el arhivo: <carpeta\_raiz>/src/main/resources/application.properties y utilize su editor de texto favorito para realizar la modificación.

* spring.datasource.url=jdbc:mysql://<host>:<port>/bookstore
* spring.datasource.username=<user>
* spring.datasource.password=<pass>

Sustituya <host>, <port>, <user> y <pass> por el servidor, puerto, usuario y contraseña que desea respectivamente. Ver ejemplo justo debajo.



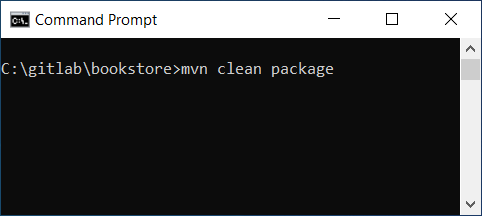
*Figura 3 – Ejemplo de los parámetros de la base de datos. Es posible que estos valores sean muy diferentes en otro ambiente.*

### Paso 3 – Compilar y construir el proyecto usando Maven

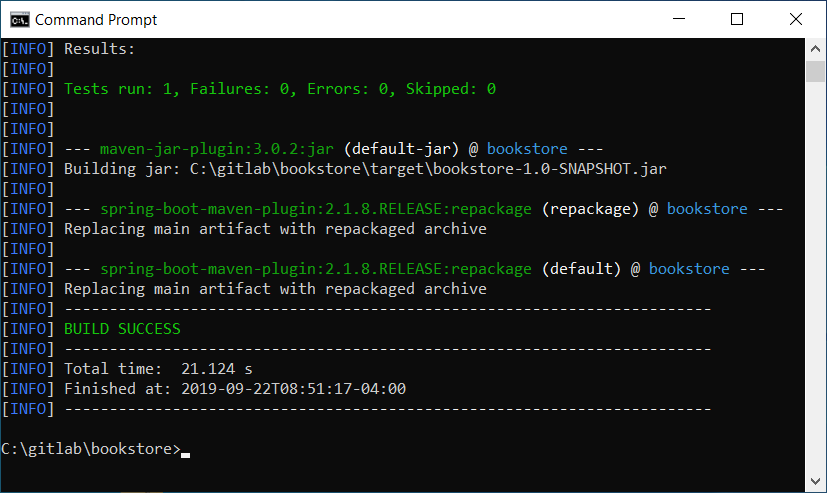
Para compilar el proyecto es necesario que navegue justo dentro de la carpeta raíz del proyecto en la línea de comandos y ejecute:

mvn clean package

Maven tiene que estar propiamente instalado en el sistema y **agregado en el path**. Maven se encargará de descargar todas las dependencias necesarias para ejecutar la aplicación. También generara un JAR ejecutable con todas las dependencias dentro del mismo archivo JAR, haciendo que sea más fácil su posterior ejecución.



*Figura 4 – Ejemplo del comando para compilar y construir el proyecto. Nota: el comando* ***mvn*** *tiene que estar previamente en el path del sistema.*



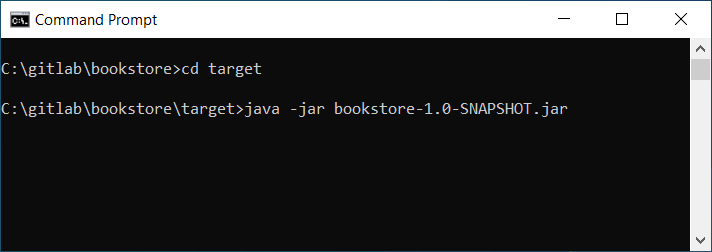
*Figura 5 – Resultado de la compilación y construcción del proyecto.*

Una vez que Maven termine de construir el proyecto asegúrese de que el archivo bookstore-1.0-SNAPSHOT.jar ha sido creado dentro de la carpeta target.

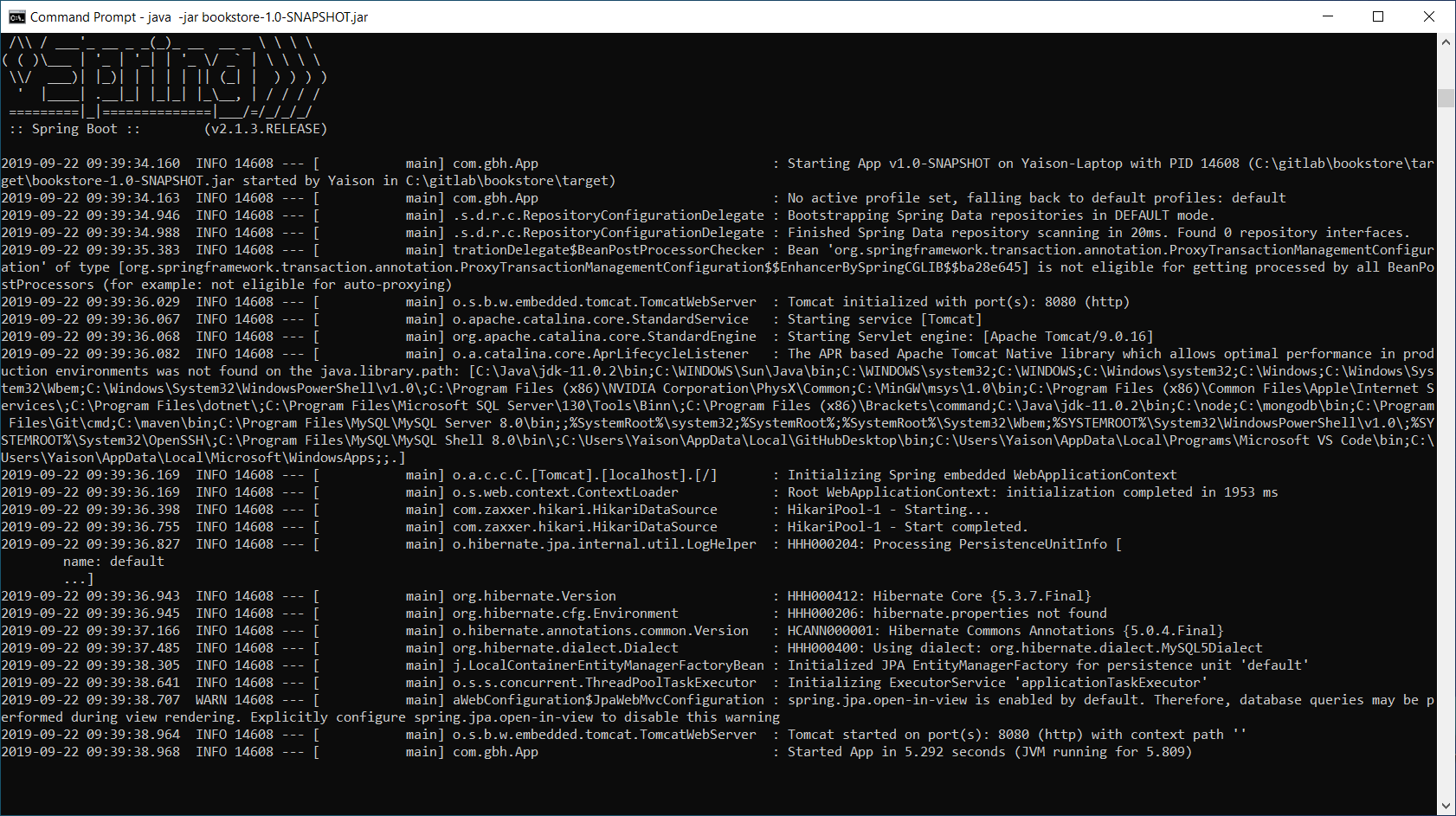
### Paso 4 – Lanzar la aplicación

Con el paso anterior el archivo que contiene todo lo necesario para ejecutar la aplicación estaría ubicado en la dirección: <carpeta\_raiz>/target/bookstore-1.0-SNAPSHOT.jar. En la línea de comando navegue hasta la carpeta target que está dentro de la carpeta raíz del proyecto. Asegúrese de que el puerto 8080 no se esté usando y ejecute el siguiente comando:

java –jar bookstore-1.0-SNAPSHOT.jar



*Figura 6 – Ejemplo de cómo lanzar la aplicación. Nota Java debe estar debidamente instalado y configurado en el* ***path*** *del sistema.*

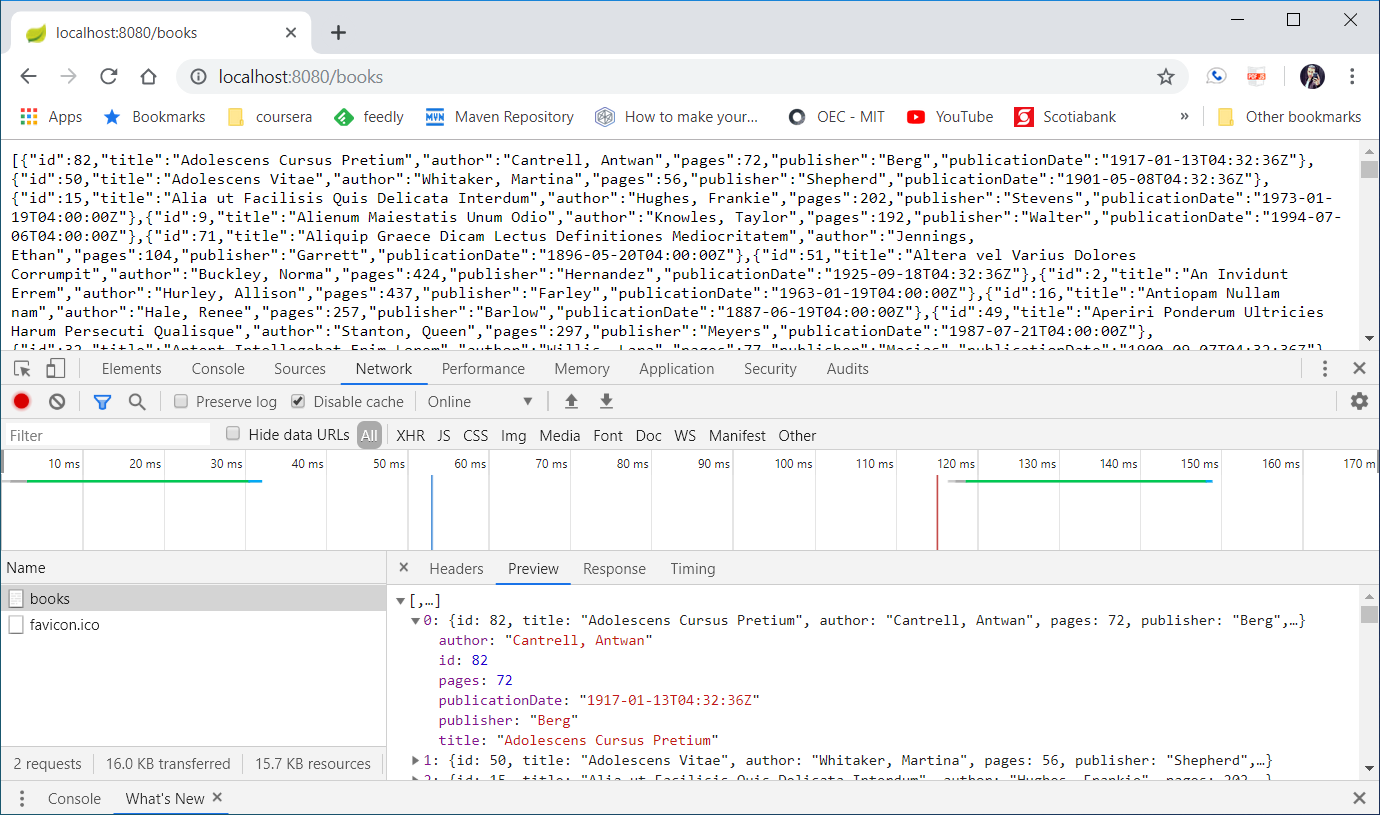


*Figura 7 – Resultado de los logs cuando la aplicación está iniciando.*

Para confirmar que la aplicación está corriendo correctamente, habrá su navegador preferido y coloque en la barra de dirección:

<http://localhost:8080/books>

Le debe mostrar una lista de libros en formato JSON.



*Figura 8 – Resultado de la petición a /books en donde la app lista los libros que están en la base de datos.*

## Uso de la Aplicación

Esta aplicación es un servicio web tipo REST en donde permite que otras aplicaciones y software se conecten a ella vía el protocolo HTTP para extraer información de los libros. El protocolo HTTP es un estándar muy reconocido y utilizado mundialmente. La gran mayoría de los lenguajes y frameworks ofrecen soporte para esta tecnología tanto de forma cliente como servidor. A continuación se detallan algunos routing como ejemplo de uso.

|  |  |  |
| --- | --- | --- |
| Method | Route | Descripción |
| GET | [/books](http://localhost:8080/books) | Lista los primero libros que están en el sistema ordenados alfanuméricamente por el título. Este path está limitado a retornar no más de 1000 libros. |
| GET | [/books/1](http://localhost:8080/books/1) /books/20 | Muestra un libro en especifico por su ID. Los ID de los libros son enteros positivos. |
| GET | [/books/13/page/5](http://localhost:8080/books/13/page/5)  /books/4/page/9?format=text  [/books/7/page/3?format=html](http://localhost:8080/books/10/page/2?format=html) | Muestra la página de un libro. Por defecto el formato es texto plano. A través de los query params se puede especificar el formato deseado. En cuanto a la decisión de porque se hizo así, por favor ver la sección **“Notas finales”**. |

## Notas finales

Esta aplicación ha sido creada con el fin de ser evaluado como posible candidato a Ing. de Software Web Senior.

Se trato de mantener la aplicación lo más sencilla posible por razones de restricciones de tiempo y su fin en sentido general, sin perder los elementos esenciales de las buenas prácticas de desarrollo de software.

**Importante:** en las instrucciones en GitLab se recomendó utilizar la URL y su path para especificar el formato del contenido de una página de un libro. Las mejores prácticas de implementación de RESTFUL Web Services van en contra de lo sugerido y en vez se utilizo un mecanismo diferente: query params.  
  
Ya que se recomienda que los path de las URL no se modifiquen una vez creado y su diseño jerárquico desde el origen, hacen que la especificación de formatos ahí no sea la más adecuada. RFC 2616 recomienda Content Negotiation, pero requiere enviar headers. En cambio los query params son más flexibles, no estorban los path de las URL y no requieren headers. Por favor consultar las siguientes fuentes:

**Hypertext Transfer Protocol -- HTTP/1.1**

Section: 12 Content Negotiation  
<https://tools.ietf.org/html/rfc2616>

**How RESTful Is Your REST?**  
Video at 27:33

https://www.youtube.com/watch?v=GQMfR0\_t4H8&t=2361s