**Oracle Java 17 (JDK)**

* Install Latest LTS JDK 17 from <https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>
  + If you have Admin Rights, download the “Windows x64 MSI Installer” version and run the MSI Installer (Change Installation Directory to: C:\JDK directory)
  + If you do not have Admin Rights, download the “Windows x64 Compressed Archive” and unzip it in C:\JDK directory
* Add C:\JDK\bin to Windows PATH Environment Variable
* Add Windows Environment Variable: JAVA\_HOME = C:\JDK
  + Open Command Prompt and Run below DOS commands to ensure Version is showing 1.7.X: java -version

**Docker**

* Download & Install Docker Desktop (Windows AMD 64) from http://docker.com/

**Curl**

* Download Curl (**Binary ZIP**) for 64-bit Windows from <https://curl.se/windows/>
* Unzip in %USERPROFILE%\dev\curl folder
* Add %USERPROFILE%\dev\curl\bin folder to Windows PATH Environment Variable

**Node.JS LTS**

1. Download Node.JS (**Binary ZIP**) from <https://nodejs.org/en/download/prebuilt-binaries>
2. Unzip in %USERPROFILE%\dev\node folder
3. Add %USERPROFILE%\dev\node folder to Windows PATH Environment Variable
4. Take a DOS Prompt and Check Node installation: node -v
5. Open Command Prompt and Run below DOS commands [Preferably Admin Mode]:

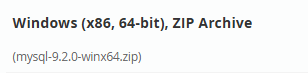
npm cache clean --force

npm install -g npm@latest

npm install -g yarn

npm install -g typescript

**Download MySQL (NO ZIP)**



1. Download MySQL (Windows X86, 64-Bit, NO-INSTALL **Binary ZIP Archive**) from <https://dev.mysql.com/downloads/mysql/>
2. Unzip in %USERPROFILE%\dev\mysql folder
3. Create a “%USERPROFILE%\dev\mysql\my.ini” text file with following contents: (Replace XYZ)

[mysqld]

# set basedir to your installation path

basedir="C:/Users/XYZ/dev/mysql/"

# set datadir to the location of your data directory

datadir="C:/Users/XYZ/dev/mysql/data/"

1. Add %USERPROFILE%\dev\mysql\bin folder to Windows PATH Environment Variable
2. Open a DOS Prompt to %USERPROFILE%\dev\mysql\bin
3. Run: mysqld.exe --initialize-insecure
4. Check database has been created in %USERPROFILE%\dev\mysql\data
5. Run MySQL Server: mysqld

Ref: <https://www.sqlshack.com/learn-mysql-install-mysql-server-8-0-19-using-a-noinstall-zip-archive/>

**Download Source Code**

**Git Credentials:**

* Repository URL: <https://github.com/connect-ocean/PoCApp>
* Username: connect-ocean
* Password: ConnectOcean@123
* Email: [eVLCon.CO@gmail.com](mailto:eVLCon.CO@gmail.com)

Copy the attached "id\_rsa\_connect\_ocean", "id\_rsa\_connect\_ocean\_pub", and "config" files in the Windows "%USERPROFILE%\.ssh" directory.

**Clone the repository:**

git clone [git@connect\_ocean:connect-ocean/PoCApp.git](mailto:git@connect_ocean:connect-ocean/PoCApp.git)

**Run Back-End Spring Boot Microservices Server:**

* Open a DOS Prompt.
* Go to folder: \back-end
* Run command: go (Then Choose Options)
* Create Database Initialization & Build
* Test in Browser: <http://localhost:8080/api/books>

**Run Front-End React.JS Microfrontend Server:**

* Open another DOS Prompt.
* Go to \front-end
* Run: npm run
* Test in Browser: <http://localhost:3000/books>

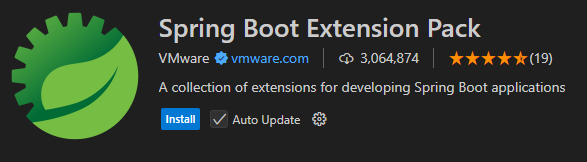
**Visual Studio Code**

1. Download & Install Latest VS Code Editor <https://code.visualstudio.com/download>
2. Install “Extension Pack for Java” and Point Correct Java Version

A screenshot of a computer

AI-generated content may be incorrect.

1. Install “Spring Boot Extension Pack”



**React JS [Not Needed when downloaded from Git]**

npm init react-app front-end

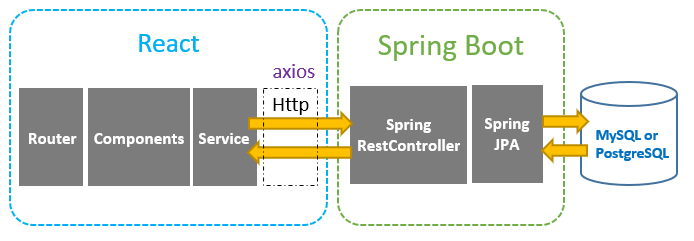
npm install -g bootstrap

npm install -g axios

npm install -g react-router-dom

npm link typescript

**Architecture:**



|  |  |
| --- | --- |
| Component | Role |
| React | HTML5 UI & Java Script Framework |
| Bootstrap |  |
| Axios | Routing Framework for Back-End Service Call  Request and Response Interception  Streamlined Error Handling  Protection against XSRF  Support for Upload progress  Response Timeout  The ability to Cancel requests  Support for Older browsers  Automatic JSON data transformation |
| React Router DOM | * Navigation and Routing: React Router provides a declarative way to navigate between different views or pages in a React application. It allows users to switch between views without refreshing the entire page. * Dynamic Routing: React Router supports dynamic routing, which means routes can change based on the application’s state or data, making it possible to handle complex navigation scenarios. * URL Management: React Router helps manage the URLs in your application, allowing for deep linking, bookmarkable URLs, and maintaining the browser’s history stack. * Component-Based Approach: Routing is handled through components, making it easy to compose routes and navigation in a modular and reusable way. |

**Spring Boot:**

**For any Jar Package Build:**