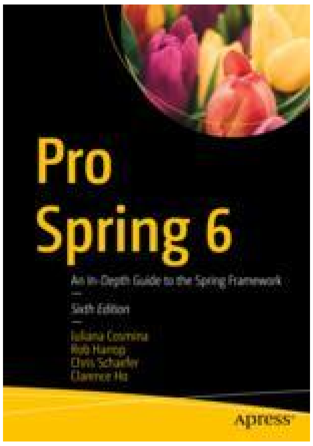
Apress Source Code

This repository accompanies **Pro Spring 6** by Iuliana Cosmina, Rob Harrop, Chris Schaefer, Clarence Ho (Apress, 2023).



Download the files as a zip using the green button, or clone the repository to your machine using Git.

NOTE Please don't skip reading this document before you get to work and the AsciiDoc files specific to each project. This project contains comments and references explaining implementation decisions that could make another book in itself. So, enjoy!

Releases

Release v6.0 corresponds to the code in the second edition of the published book, without corrections or updates. There have been small changes in the project configuration files to provide a more stable and a more up-to-date build.

This project was built successfully with **JDK 19**, Gradle **8.4** / Maven **3.9.5**. The syntax is specific to Java versions up to and including 17.

NOTE If you want to build it with **JDK 17**, modify the Maven/Gradle configuration files and edit all Java versions references.

Corrections

For corrections to the content in the published book, see the file Errata.adoc.

Contributions

See the file Contributing.adoc for more information on how you can contribute to this repository.

Building and Deploying

NOTE This project requires **Docker** installed and running.

NOTE Projects `chapter07-jooq` and `chapter07-jooq-boot` were updated to allow for a full build, which required some changes in the configuration of the Maven/Gradle files. However, they are still unpredictable, and sometimes they might fail because the container is not ready in time. I expect this to happen on slower computers. If this is an issue for you, just remove these projects from the general build, or just don't run the full build at all.

NOTE For easier Java versions management, if you are using a Linux/macOS I also recommend installing SDKMAN. And running the following command, to use the same JDK I used when writing this project:

```
> sdk install java 19.0.1-amzn
```

Using Gradle

The project is configured with the following default Gradle tasks:

```
defaultTasks 'clean', 'build'
```

This means that you do not have to specify those tasks when building the project from the terminal. Build it from scratch using the Gradle wrapper:

```
> ./gradlew
```

Or if you have Gradle installed locally, open a terminal and just run:

```
> gradle
```

If you want to skip the tests (build time will be shorter) run the wrapper with the following arguments:

```
> ./gradlew -x test
```

Or run Gradle with the following arguments

```
> gradle -x test
```

IMPORTANT Since I did not add the Gradle wrapper jar to the repository, download it by running `gradle wrapper --gradle-version 8.4`.

Using Maven

The project is configured with the following default Maven goals:

```
<defaultGoal>clean install</defaultGoal>
```

This means that you do not have to specify those goals when building the project from the terminal. Build it from scratch using the Maven wrapper:

```
> ./mvnw
```

Or if you have Maven installed locally, open a terminal and just run:

```
> mvn
```

If you want to skip the tests run the wrapper with the following arguments:

```
> ./mvnw -DskipTests
```

Or run Maven with the following arguments

```
> mvn -DskipTests
```

IMPORTANT Since I did not add the Maven wrapper jar to the repository, download it by running `mvn -N wrapper:wrapper -Dmaven=3.9.5`.