

Table of Contents

- [CRUD web application with responsive design](#)
 - [Frameworks and Tools used](#)
 - [Language](#)
 - [Inversion of Control / Dependency Injection / Middleware](#)
 - [Persistence](#)
 - [Presentation / User Interface](#)
 - [Application Server \(Servlet Container, actually\)](#)
 - [Dependency Management and Build Tool](#)
 - [IDE](#)
 - [Purpose](#)
 - [How to run this project](#)
 - [Embedded H2 database console](#)

CRUD web application with responsive design

This project is a simple CRUD web application with responsive design, using Java + Spring Framework + JPA + JSF and other technologies.

Frameworks and Tools used

Language

- Java 8 (JDK 1.8)

Inversion of Control / Dependency Injection / Middleware

- Spring Framework 4.3.x

Persistence

- JPA 2.1
- Hibernate 5.2.x
- H2 Database 1.4.x (it could be any relational database)

Presentation / User Interface

- JSF 2.3.x
- Primefaces 6.x
- OmniFaces 2.6.x

Application Server (Servlet Container, actually)

- Tomcat 8.5.x

Dependency Management and Build Tool

- Maven 3.0

IDE

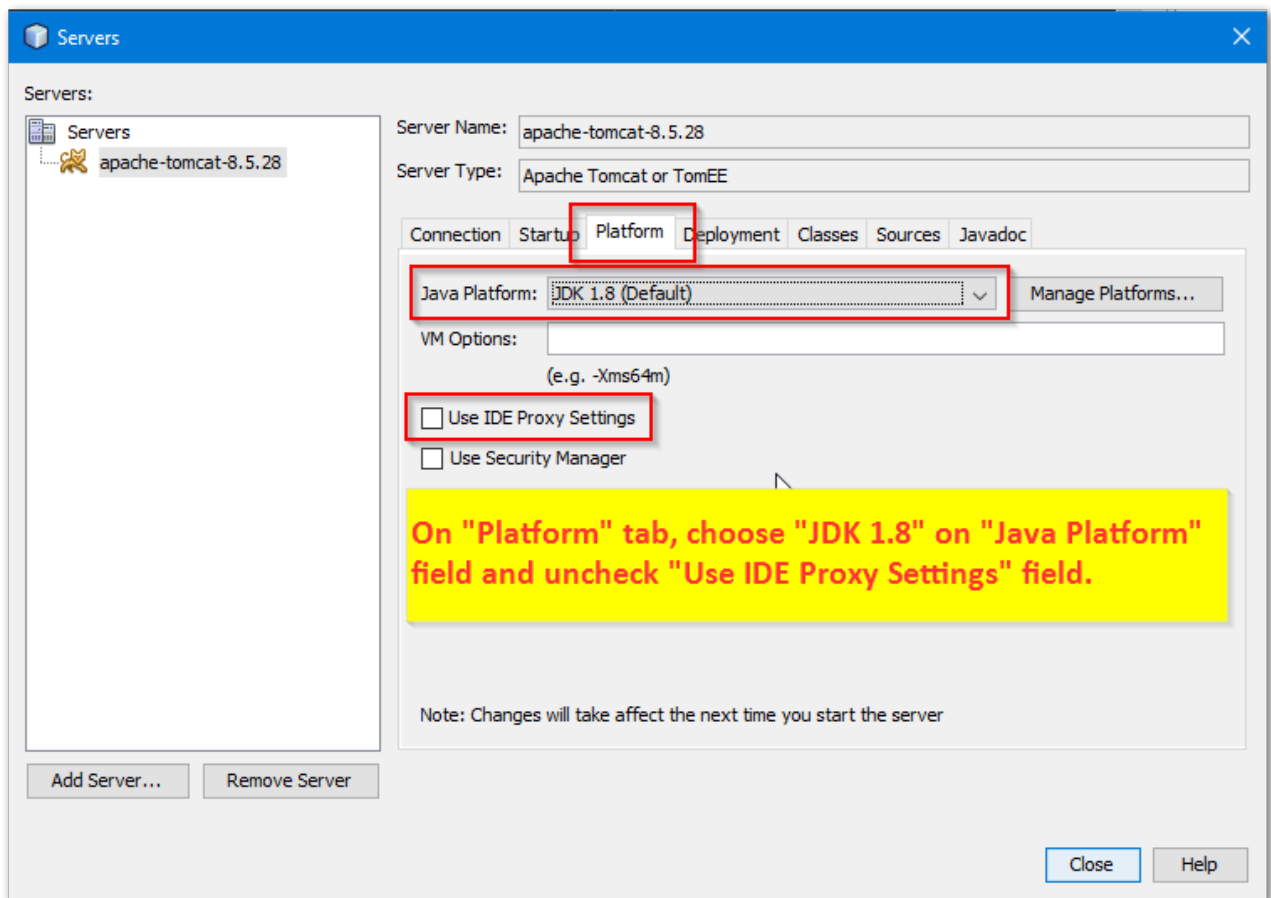
- Netbeans IDE 8.2 (it could be Eclipse IDE or any Maven compatible IDE, or just plain Maven via command line is fine)

Purpose

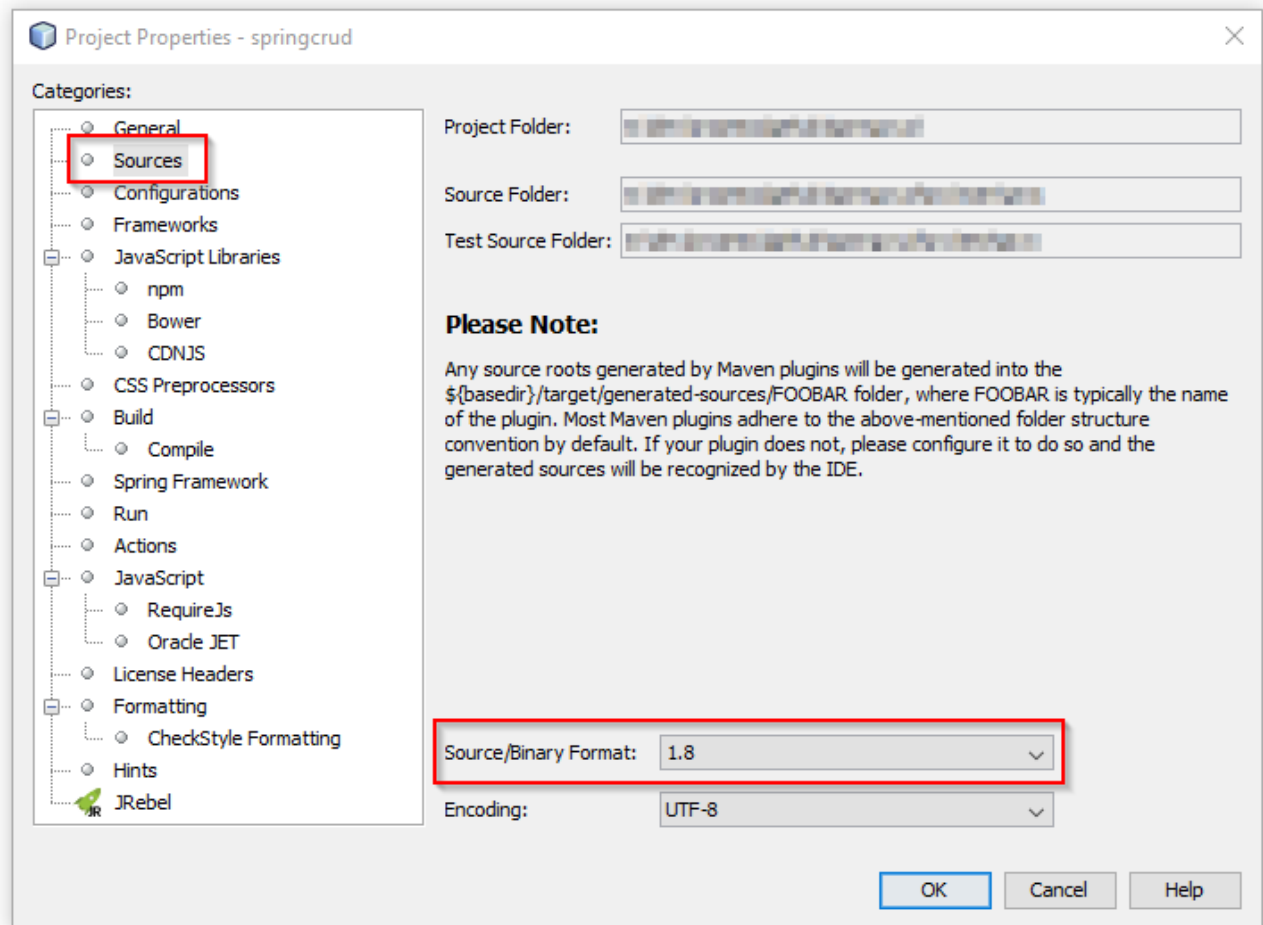
The main purpose of this project is to serve as a starting point for those who are struggling to put these above mentioned technologies to work together.

How to run this project

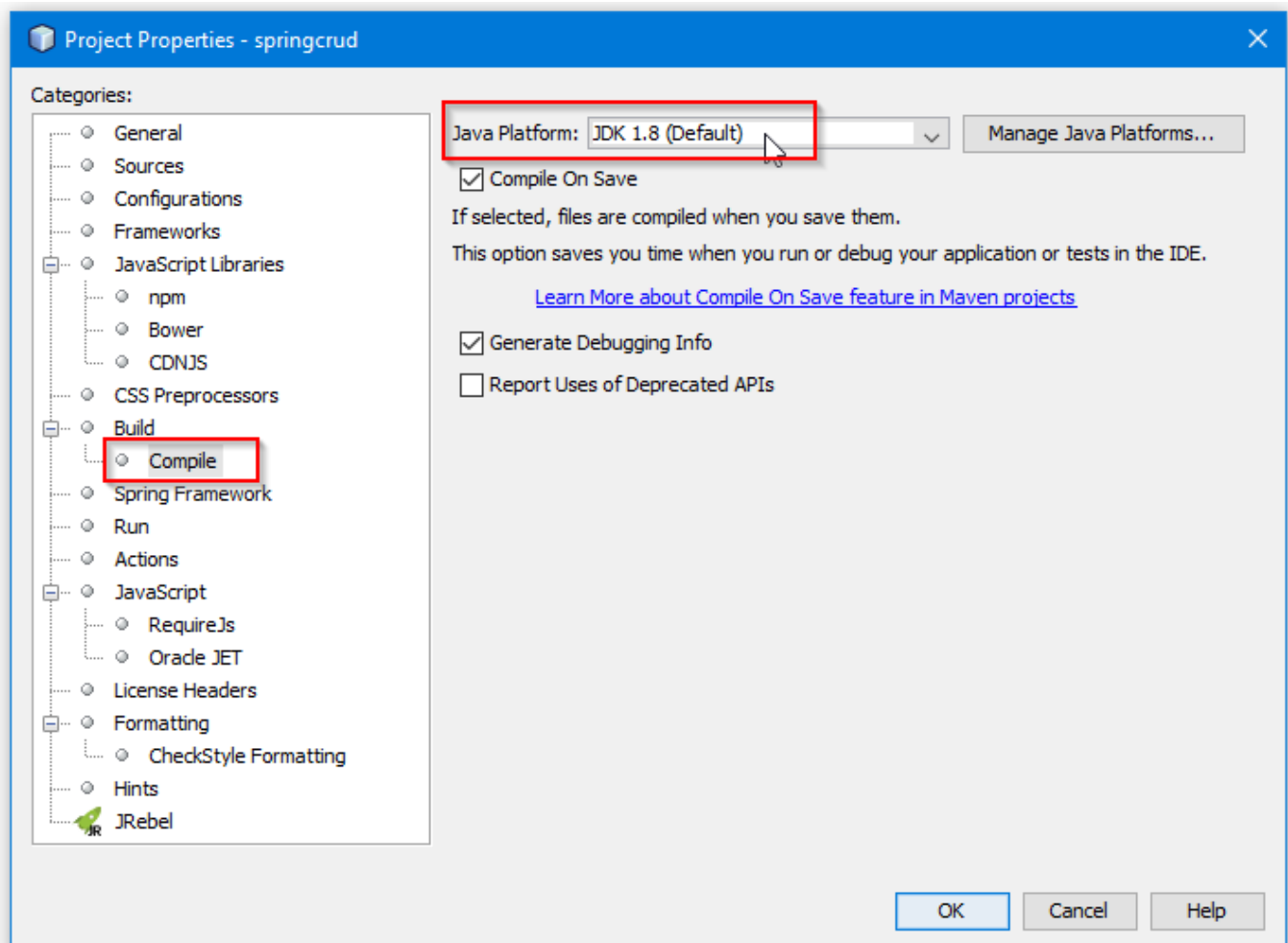
1. Download **Apache Tomcat 8.5.x** from <https://tomcat.apache.org/download-80.cgi> ([Windows](#) or [Linux](#)) and extract it to any folder on your computer
2. Open **Netbeans IDE**, go to "**Services**" window (Ctrl+5), right click on "**Servers**" node and select "**Add Server...**". Select "**Apache Tomcat or TomEE**" on "**Server**" field, change server name if you will and click "**Next**" button. On "**Server Location**", choose the folder where you extracted Tomcat on step 1 above.
3. Still on "**Services**" window, right-click on server you just created under "**Servers**" node and select "**Properties**". Under "**Platform**" tab, choose **JDK 1.8** in "**Java Platform**" field and **unchecked** "**Use IDE Proxy Settings**" field, as shown below



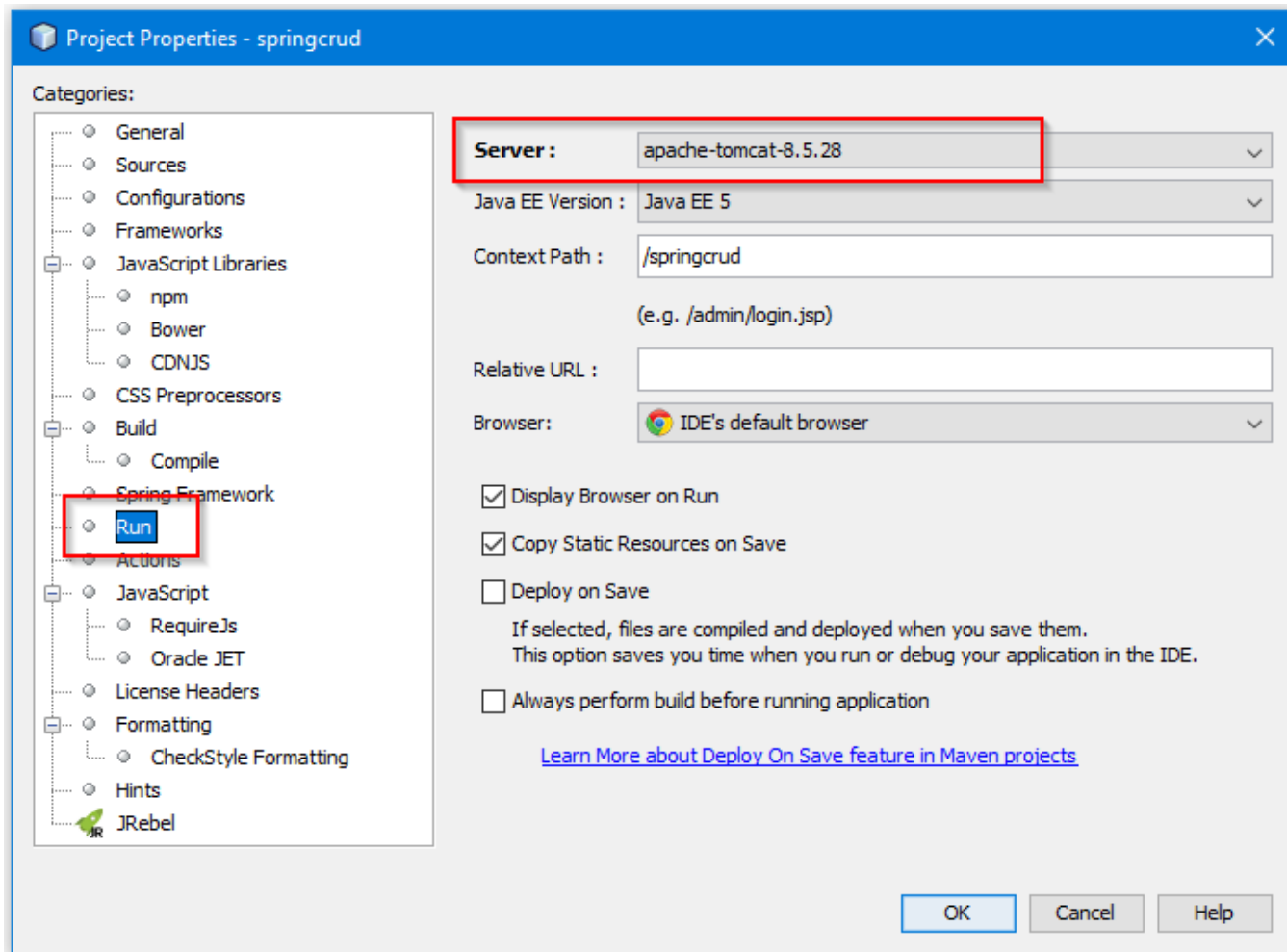
4. Clone/Fork this project to your machine and open it in Netbeans ("**File**" / "**Open Project...**" menu)
5. In Netbeans, right-click on project and select "**Properties**". In "**Sources**" node, select "**1.8**" on "**Sources/Binary Format**" field, as shown below



Under "**Build**" / "**Compile**" node, choose "**JDK 1.8**" in "**Java Platform**" field, as shown below



Under "**Run**" node, "**Server**" field, choose the app server you created in step 2 above and leave other fields as shown below



Close this Project Properties window.

6. Right-click on project and select "**Run**", wait for Tomcat start up and, if your browser won't open automatically, go to `http://localhost:8080/springcrud` . You'll see something like this:

Desktop mode:

Person CRUD

New Person

Person Form

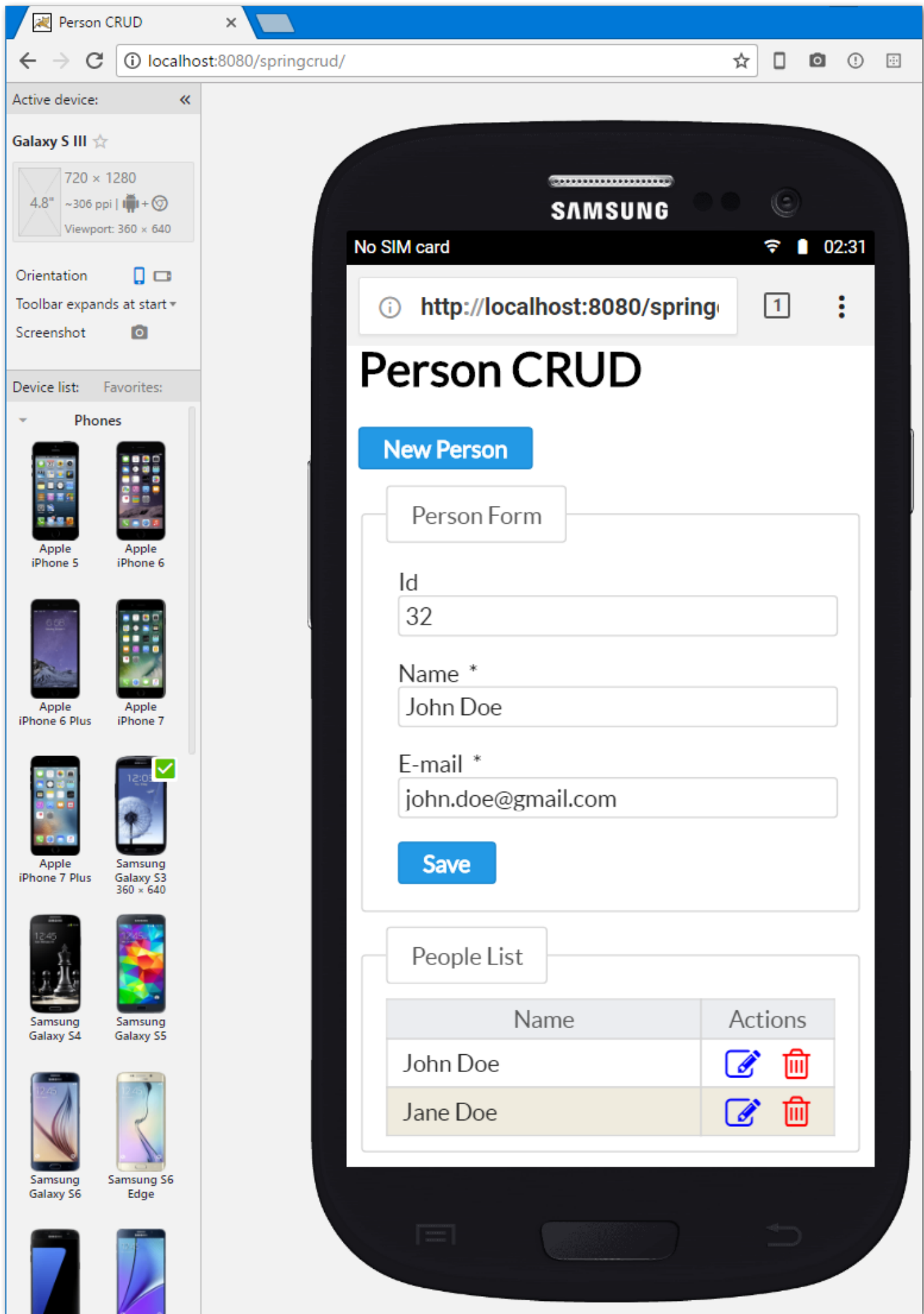
Id: 33, Name *: Jane Doe, E-mail *: jane.doe@gmail.com

Save

People List

Id	Name	E-mail	Actions
32	John Doe	john.doe@gmail.com	
33	Jane Doe	jane.doe@gmail.com	

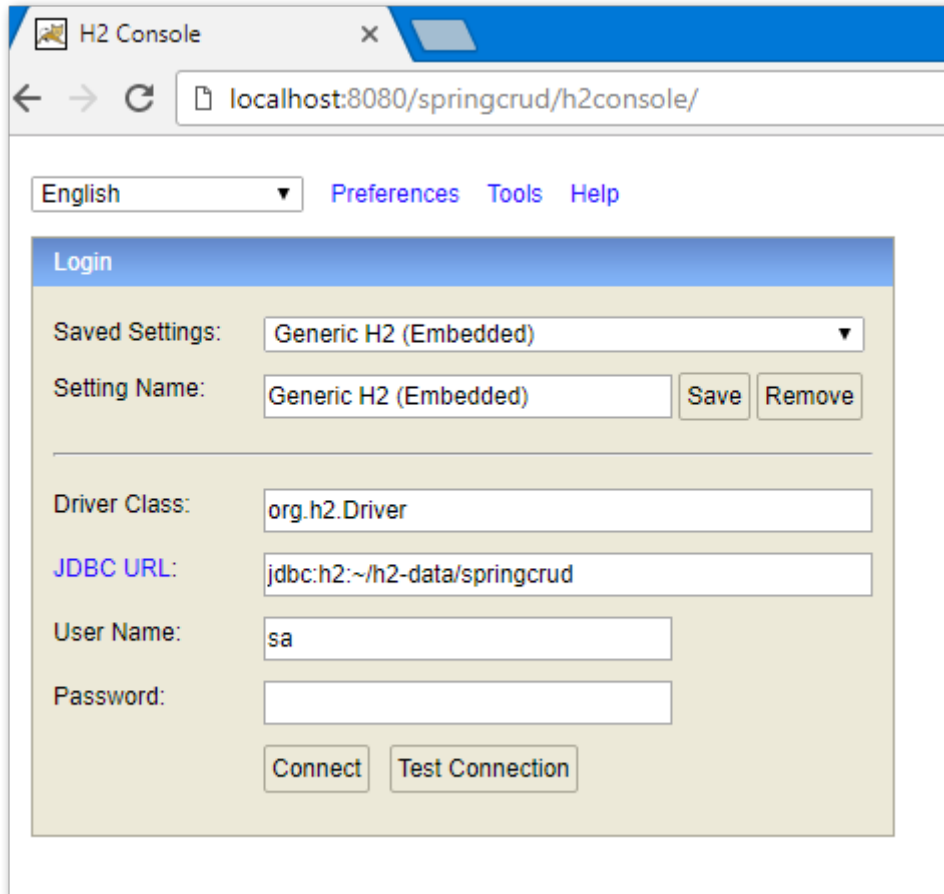
Responsive/Mobile mode:



Embedded H2 database console

This project comes with a built-in SQL client console, allowing you to access H2 Database that serves as the backend. Just follow the steps below to access it:

1. Run the project (step 6 above) and open the following URL in your browser:
`http://localhost:8080/springcrud/h2console`
2. Enter "**jdbc:h2:~/h2-data/springcrud**" in the "**JDBC URL**" field and click "**Connect**" button, as shown

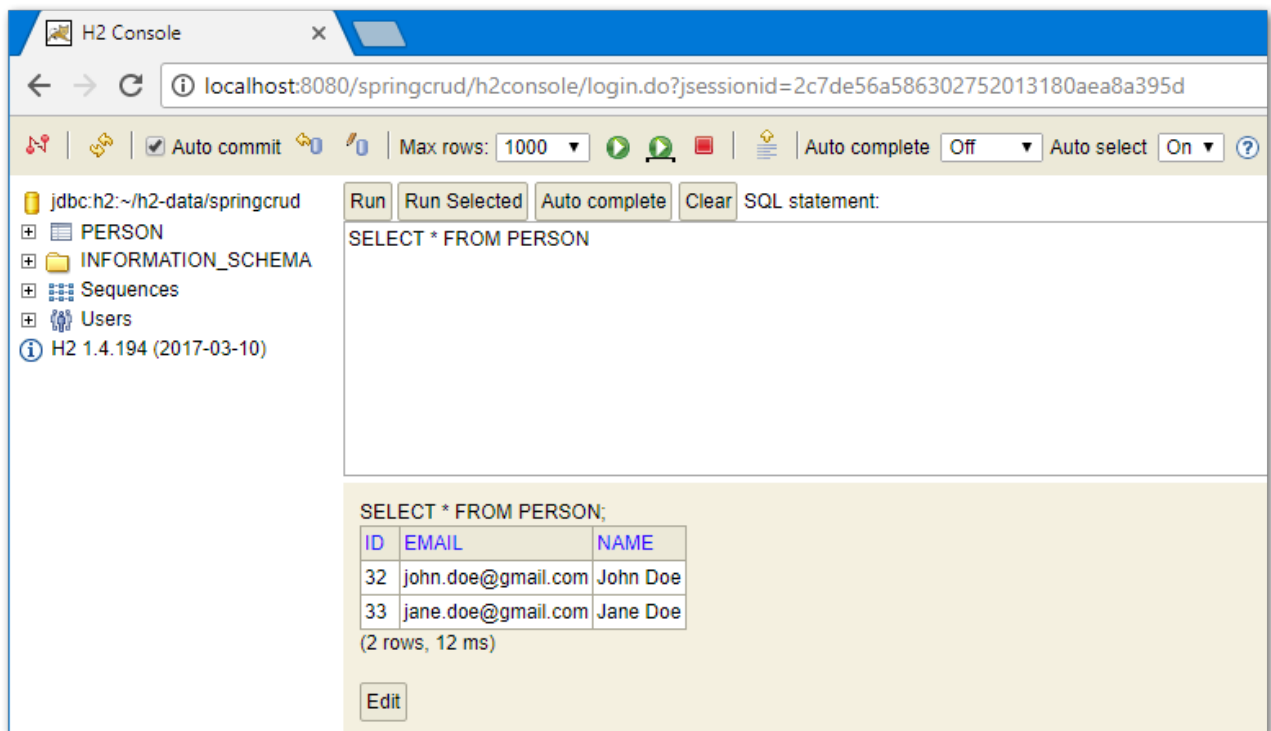


The screenshot shows the H2 Console web interface in a browser window. The address bar displays `localhost:8080/springcrud/h2console/`. The interface includes a language dropdown set to "English" and navigation links for "Preferences", "Tools", and "Help". A "Login" section contains the following fields and controls:

- Saved Settings:** A dropdown menu showing "Generic H2 (Embedded)".
- Setting Name:** A text input field containing "Generic H2 (Embedded)", with "Save" and "Remove" buttons to its right.
- Driver Class:** A text input field containing "org.h2.Driver".
- JDBC URL:** A text input field containing "jdbc:h2:~/h2-data/springcrud".
- User Name:** A text input field containing "sa".
- Password:** An empty text input field.
- Buttons:** "Connect" and "Test Connection" buttons at the bottom of the form.

below

3. You can execute any SQL command to show table data, as shown below



The screenshot shows the H2 Console web interface in a browser. The address bar displays the URL: `localhost:8080/springcrud/h2console/login.do?jsessionid=2c7de56a586302752013180aea8a395d`. The interface includes a toolbar with icons for connection, execution, and settings. The left sidebar shows the database structure: `jdbc:h2:~/h2-data/springcrud`, `PERSON`, `INFORMATION_SCHEMA`, `Sequences`, `Users`, and `H2 1.4.194 (2017-03-10)`. The main area contains a text input for the SQL statement: `SELECT * FROM PERSON`. Below the input, the results of the query are displayed as a table with columns `ID`, `EMAIL`, and `NAME`. The results show two rows: `32 john.doe@gmail.com John Doe` and `33 jane.doe@gmail.com Jane Doe`. The status bar indicates `(2 rows, 12 ms)`. An `Edit` button is located at the bottom left of the results area.

SQL statement:

```
SELECT * FROM PERSON
```

ID	EMAIL	NAME
32	john.doe@gmail.com	John Doe
33	jane.doe@gmail.com	Jane Doe

(2 rows, 12 ms)

Edit