Preconditions:
Maven
Java 1.8
Git Bash
1. Clone API to your PC from git repository;

2. Open `application.properties` file by `src\main\resources` path and change `spring.h2.console.enabled` property to `true`;
`spring.h2.console.enabled=true`

3. Load root folder, open console here and execute command:
`mvn spring-boot:run`
This will build and up API on Tomcat server on the localhost with 8080 port (by default, but you can change it in `application.properties` in `server.port` property)
Wait for something like this message in your console:
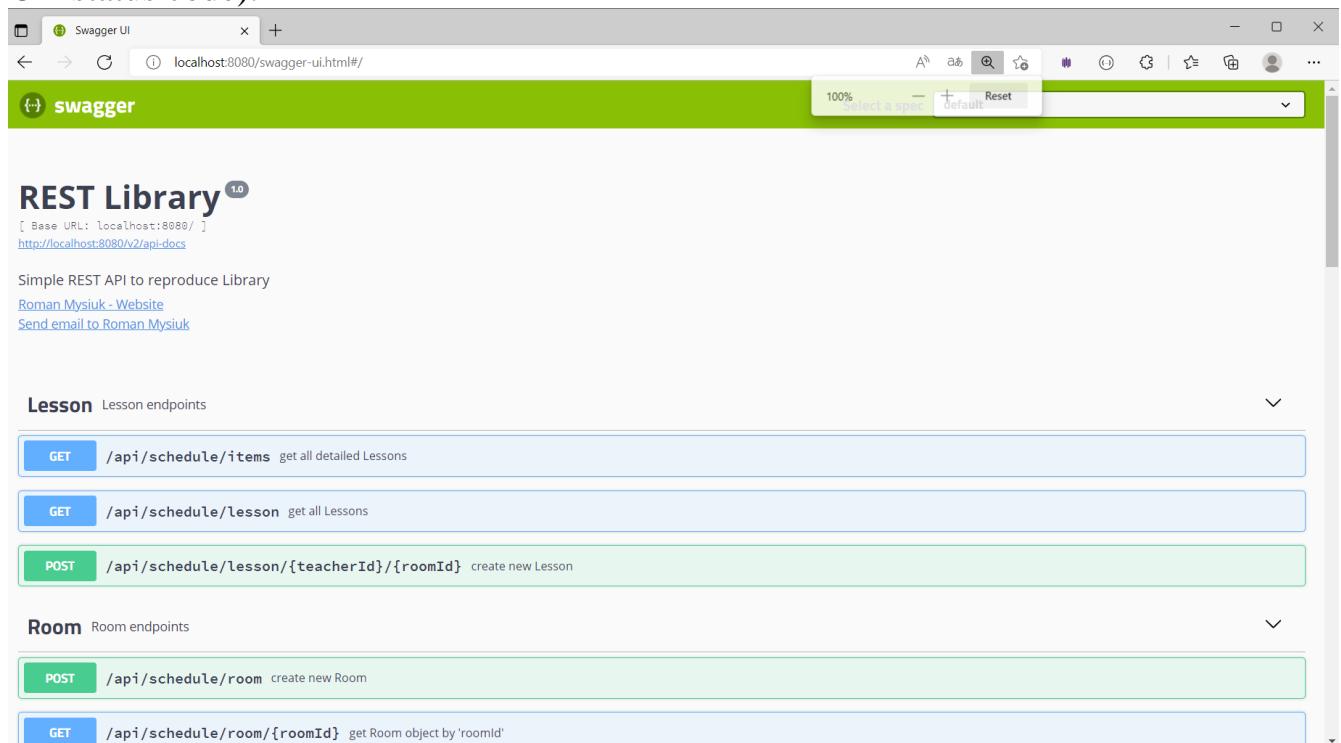`2019-08-19 13:11:42 INFO 59940 - [main] ua.com.epam.app.App: Started App in 6.829 s.`
That proof that API was started and ready for work;

3. Swagger will be available here:
`http://localhost:8080/swagger-ui.html`
It is possible that Swagger will not loading after first build. To fix this just open console and press any button on keyboard (e. g. `Enter`);
Try to make call on some endpoint to verify that everything works OK (just to get 200 OK status code).
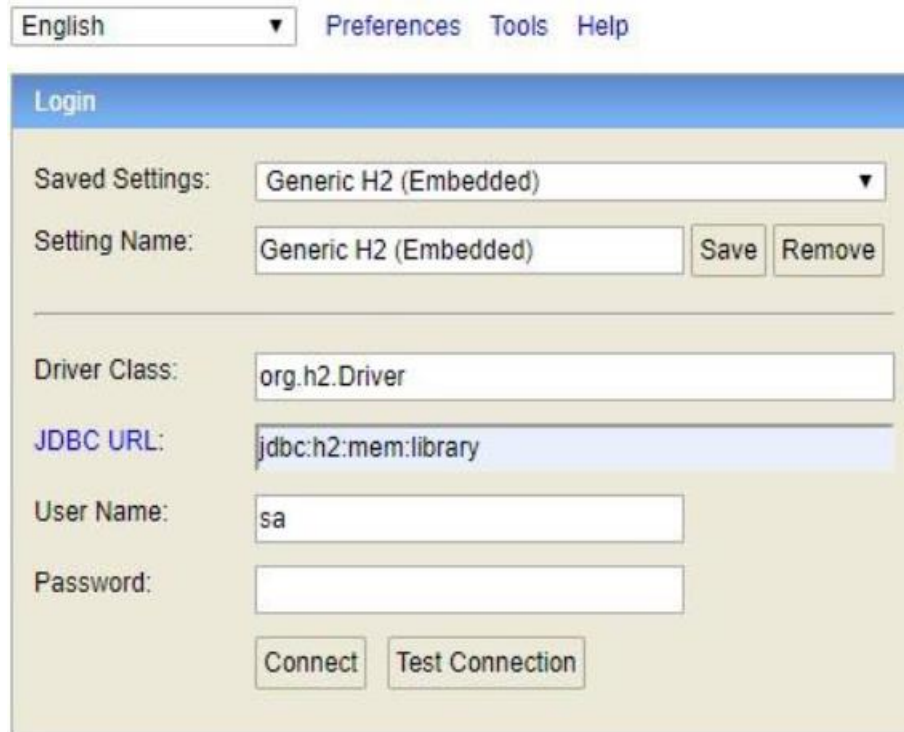
4. REST-library-API application use built in database H2.
You can access to H2 console (something like MySQL Workbench for MySQL database). Database file was created on application compiling stage and locates here: `src/main/resources/db` with name ended on `.mv.db`;
H2 console is available on:
http://localhost:8080/h2-workbench
It will load form like this:

In ``JDBC:URL`` input paste: **jdbc:h2:file:<absolute_path_to_.mv.db_file>**
(E. g.: `jdbc:h2:file:` C:\REST-library-API\src\main\resources\db\library)
In ``User Name`` input paste: **sa**
In ``Password`` input paste: **password**
And click ``Connect`` button. After this you will see something like this:

H2 Console

localhost:8080/h2-workbench/login.do?jsessionid=45c5e5bcaab7cd811ff7570def4868e9

Auto commit Max rows: 1000 Run Run Selected Auto complete Clear Auto complete Off Auto select On

jdbc:h2:D:/PhD/SWE/Бази даних | Run | Run Selected | Auto complete | Clear | SQL statement:

SELECT * FROM LESSON;

- DETAILED_LESSON
- ITEM
- LESSON
- ROOM
- SUBJECT
- TEACHER
- INFORMATION_SCHEMA
- Sequences
- Users
- H2 1.4.199 (2019-03-13)

SELECT * FROM LESSON;

ID	DAY_OF_WEEK	LESSON_NUMBER	ROOM_ID	SUBJECT_ID	TEACHER_ID	ROOM_NAME	SECOND_NAME	SUBJECT_NAME
1	1	1	1	2	1	null	null	null
2	1	2	2	1	2	null	null	null

(2 rows, 21 ms)

Edit

HTTP запити
GET, POST,
PUT, DELETE

Веб сторінка

Веб сервіс
(Переписує у
відповідний формат
JSON -> QUERY)

Запити в базу
даних
SELECT, INSERT,
UPDATE, DROP

База даних
MySQL

Що таке веб сервіс?

Це обгортка з HTTP запитами для запитів до бази даних, яка дозволяє безпечно працювати з даними

---

controller
- LessonController
- RoomController
- SubjectController
- TeacherController

HTTP запит

service
- mapper
- util
- LessonService
- RoomService
- SubjectService
- TeacherService

**Переведення
в моделі для
JSON**

entity
- dto
  - item
    - DetailedLessonDto
  - lesson
    - LessonDto
  - room
    - RoomDto
  - subject
    - SubjectDto
  - teacher
- DAY_OF_WEEK
- DetailedLesson
- Lesson
- Room
- Subject
- Teacher

Моделі з MySQL

Dto – моделі для JSON

repository
- ItemRepository
- JsonKeysConformity
- LessonRepository
- RoomRepository
- SearchFor
- SubjectRepository
- TeacherRepository

**Запити в базу
даних**