# Java Springboot TEST

## TEST

* Create a Springboot project (no matter version or project name) from scratch that uses Java 8 or higher.
* Add to the project a database (in memory, it doesn't matter which one you choose), which contains a table (the name of the database and the schema doesn't matter)
* The table should be called: Users. It must contain 5 columns. User, Fullname, Phone, Address and Password
* The Users table must be initialized with the following data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User** | **Fullname** | **Phone** | **Address** | **Password** |
| User1 | Thomson, Elias | 555-8596 | Diamond St. 4G3 NY | 123 |
| User2 | Simond, Ella | 415-9687 | Fleet st. 45 B, 56 BR-NY | 456 |
| User3 | Clifford, Thomas | 416-69883 | Meet town, 45 - FL | 789 |

- Create a free access end-point /api/login POST, with user and password in the body, which must be crossed against the database table. In case of a match, a user identification token and a 200 HTTP code must be returned. The format of the token is up to the candidate's choice and for simplicity it is not necessary for it to be signed. If user and pass do not work, a 401 HTTP code must be returned. For simplicity, passwords are not hashed

- The rest of the endpoints in this document must be secured in such a way that it is verified that the request receives a valid token generated by the login endpoint, from which it must be possible to extract who the user is making the call. In case a call arrives without a token or with an invalid token, a 401 HTTP code must be returned

- The project must contain an end-point (in a controller) “/api/users”, GET, without input parameters, which returns a file containing the data from the Users table except the password column.

- The format of the file does not matter (CSV, TXT, etc.), the important thing is that it contains all the data from the Users table (the format of the content also does not matter).