LKSN 2020 – Web Documentation

ESEMKA School is about to develop e-learning online system. Their main business model are focusing on managing student data, teacher data, subject data, and scoring.

To help you in understanding the project, we have provided you with several documentations such as:

1. API Documentation which you can used with Swagger or Postman. This API Documentation will describe the endpoints that needs to be build.
2. ERD Diagram which you can used to describe the database structure which you may need to help you to understand the structure.
3. Database script which you can used to export the database scheme.

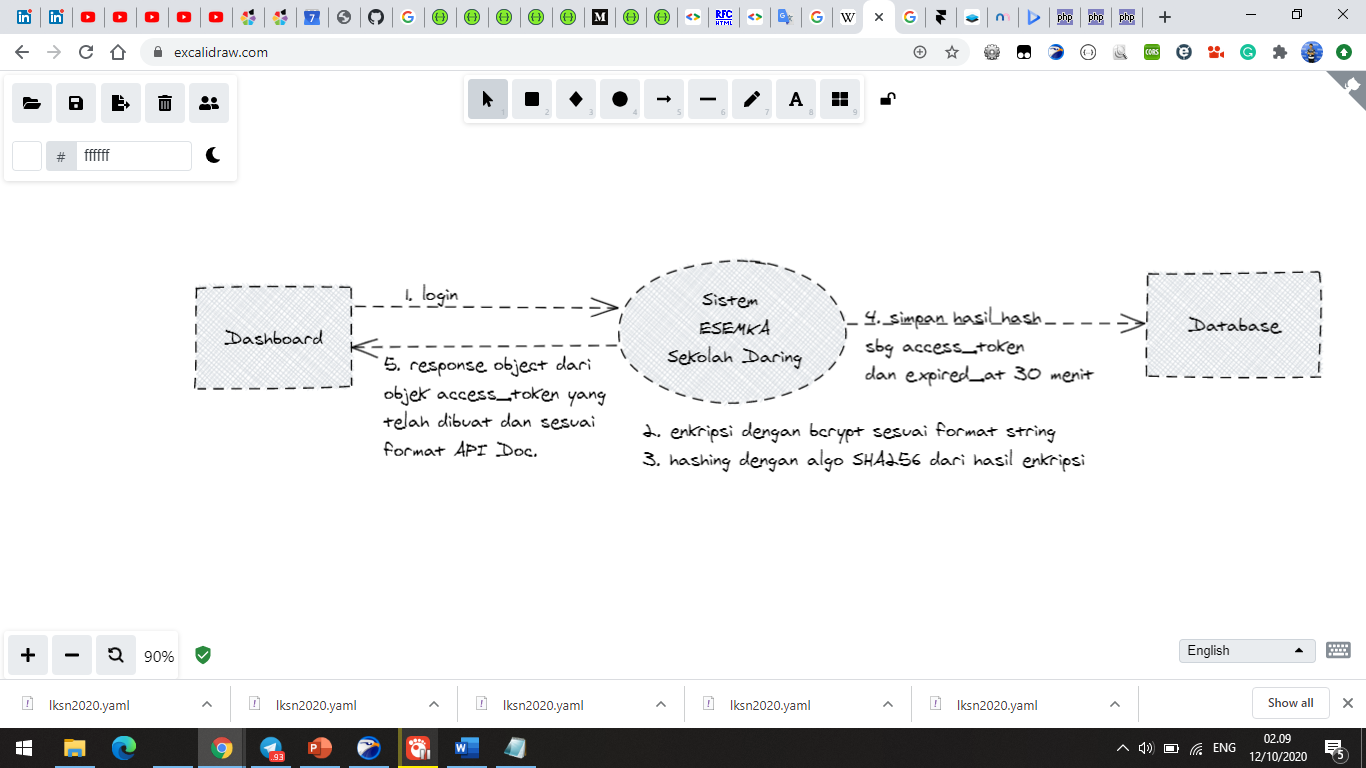
Below we are the description of some modules which may need special attention to develop.

1. **Authentication**

On authentication process, in order to retrieve access token, you need to login first and if your credential match the data in database then the system will encrypt the combination of user id (whether teacher id or student id) with user type and the timestamp (e.g. “1-student-2020-10-01 10:10:00.000”). Detail explanation about the encryption process with bcrypt can be seen in <https://www.php.net/manual/en/function.password-hash.php>.

The encrypted result then will be hashed using SHA256 algorithm. The detail explanation about hashing using SHA256 algorithm can be seen in <https://www.php.net/manual/en/function.hash.php>.

The result of hashed value later will be stored in access\_token table in database with expire time about 30 minutes. The process to retrieve access token will be describe in diagram down below.

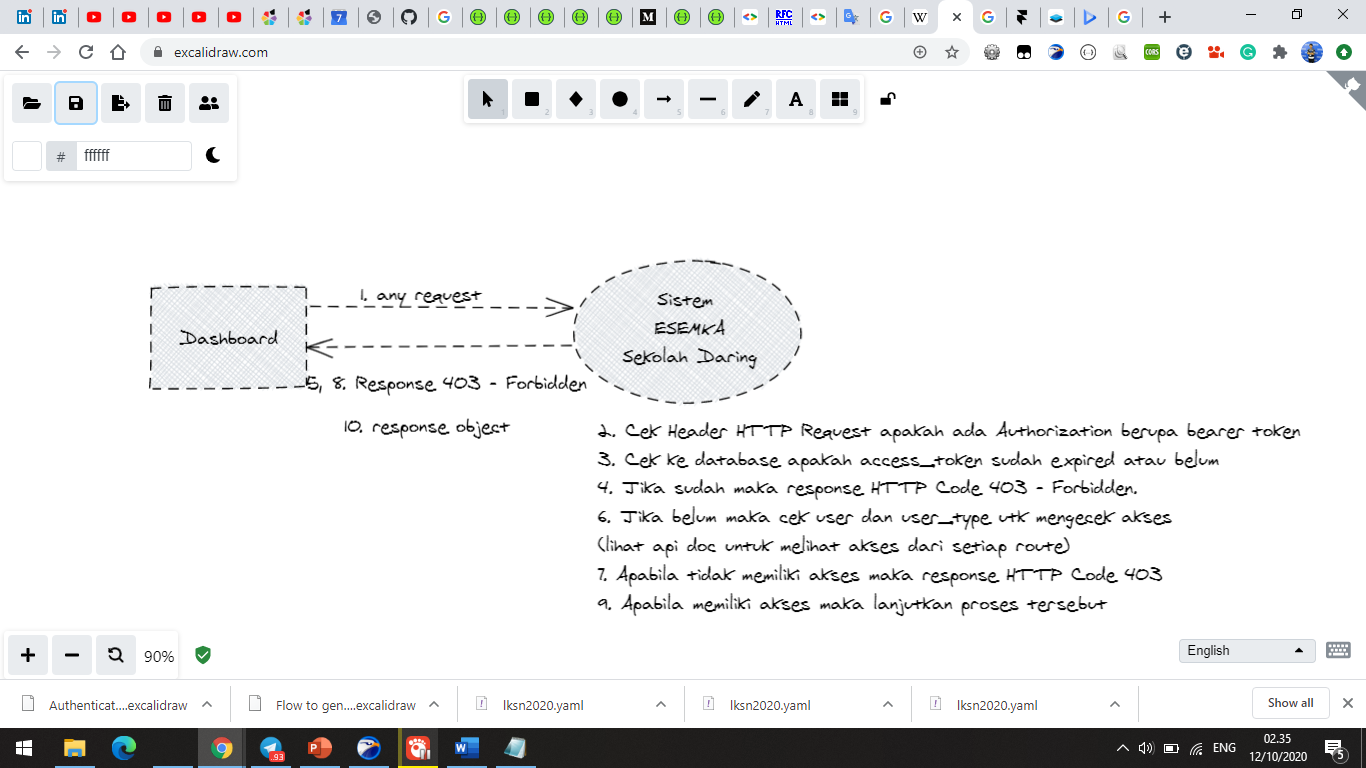


The authentication process will be proceed following these steps:

1. Every HTTP Request coming through will be trigger the authentication process to be proceed. (The authorization will be put in request header with Bearer token standard, e.g.

Header: [Authorization: Bearer *<<access\_token>>*]

1. If the request header contains authorization in it then check the access\_token. If the access token still valid which mean the token is found in access\_token table and still not expired yet, then the process can be continue to be proceed to the url given. But if the token is not the same with any token in access\_token table or already expired then return with “Forbidden Message” with HTTP Code 403. (See API Documentation regarding this the response for every URL route)



1. **Forgot Password**

Forgot password can be proceed by generating the activation key. Activation key is produced of encryption using bcrypt with the following format.

{email}

For example:

“budi@sekolah.daring.org”

Link that going to be generated would be like

[https://sekolah.daring.org/v1/reset/{hasil\_enkripsi}](https://sekolah.daring.org/v1/reset/%7bhasil_enkripsi%7d)

The link later on will be sent to the email which provided on request body with POST Method.

The email will follow this template.

Halo {{nama}},

Berikut link untuk *reset password* anda.

[https://sekolah.daring.org/v1/reset/{hasil\_enkripsi}](https://sekolah.daring.org/v1/reset/%7bhasil_enkripsi%7d)

Terima kasih.

1. **Reset Password**

*Reset password* will be proceed with following steps:

1. Check the key string provided on the query params which is encrypted key generated on *forgot password*.
2. If the match with the email of active user (whether student or teacher) the check the request body if the password value matches the confirmation password value then proceed to reset the password and sending the email following the template down below. But if the password value does not match the confirmation password value then return error message “Password tidak sama dengan Konfirmasi Password”.

Halo {{nama}},

Untuk informasi anda, baru saja ada pengubahan kata sandi untuk akun anda.

Sekian.  
Terima kasih.

1. **The Rest of the modules**

For the rest of the modules, the details can be seen on API Documentation given. If there is any question about the modules can be asked later when the session started.

😊 Good Luck 😊