**Server Configuration:**

Use application.properties file to configure server:

server.port - *TomCat port on which API will be listened in.*

spring.datasource.\* *- external data base configuration (driver, port, etc)*

*NOTE: server was configured to use MySQL, if you have another one please add it in* pom.xml (or remove comments)

spring.liquibase.\* - *configuration for tool that will fill database and create nesessary tables.*

jwt.token.expiredms - *expired time for token in miliseconds*

**Rest Api:**

(default server address is localhost:8080)

**/api/auth/login**

used for user’s login.

Request:

POST JSON msg

*{*

*"username": "test",*

*"password": "test"*

*}*

*(note: user must be in database (liquibase will create defoult user test/test))*

Response:

JSON msg

*{*

*"username": "test",*

*"token": "[some token]"*

*}*

*(***note***: use this token for authorization to other services*.

*As example in Postman add in Headers use:*

*Key: Authorization*

*Value: Bearer\_TOKEN*

*(as excemple value - Bearer\_eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ0ZXN0Iiwicm9sZXMiOlsiUk9MRV9VU0VSIiwiUk9MRV9BRE1JTiJdLCJpYXQiOjE2MTI3NzQ5OTgsImV4cCI6MTYxMjc3ODU5OH0.oiWvpcHXs0POrHulCUbbbBi0Sx37t5rTc2YcokZd\_tQ)*

*)*

**/api/admin/createadmin**

used for create new user with administrator;s rights

Request:

*POST JSON msg*

*{*

*"username": "name",*

*"password": "p@ssword"*

*}*

Response:

*JSON msg with created user*

*{*

*"id": 3,*

*"username": "test44",*

*"firstName": "first",*

*"lastName": "second",*

*"email": "some@email",*

*"status": "ACTIVE"*

*}*

*(NOTE: some info is for TODO :) )*

**/api/admin/users**

show all users in database

Request:

*GET HTTP msg*

Response

*JSON msg with all users in format as in admin creation*

**/api/admin/users/{ID}**

get user by id

Request:

GET HTTP msg

Response

JSON msg with user ID = user’s ID in DataBase

**/api/todo/note**

how all notes in database

Request:

*GET HTTP msg*

Response

*JSON msg with all todo notes in format*

*{*

*"id": 1,*

*"text": "note text",*

*"autor": "autor",*

*"status": "ACTIVE"*

*}*

*(NOTE: status could be only “ACTIVE” or “NOT\_ACTIVE”)*

**/api/todo/activenotes**

how all notes in database with ACTIVE status

Request:

GET HTTP msg

**/api/todo/note/{id}**

GET HTTP msg

show note by ID

**/api/todo/createnote** and  **/api/todo/updatenote/{id}**

create new todo note or if it exist update old one (for update you don’t need to fill all JSON fields)

Request:

*POST JSON msg*

*{*

*"text": "some text",*

*"autor": "autor",*

*"status": "ACTIVE"*

*}*

Response

*JSON msg with new or updated note*

**/api/todo/updatenote/{id}**

delete note by ID