# Backend documentation

## Endpoints:

POST /api/register

Body content: { login: [e-mail], password: [password] }

Responses:

200 – const newUser = {

            eMail: login,

            password: hashPasword,

            pregnanyStart: null,

            photo: null,

            firstName: null,

            lastName: null,

            age: null,

            height: null,

            weight: null

        }

404 – User is already registered

500 – Other problem

POST /api/login

Body content: { login: [e-mail], password: [password] }

Responses:

200 –const token = {

            accessToken: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVmODE2Yjc3ZjBkNzQ2M2YzMDlhZWRlNSIsImlhdCI6MTYwMjMxOTQyMn0.3IUdaTbdLCIvdZfyPbOd6dfGa-hzi0tHb008DOsdfHA"

        }

404 –Bad e-mail or password

GET /api/user

Header type: Authorization

Header content: ‘Bearer [accessToken]’

Responses:

404-User not found or bad id

401 –No token in header

403 –Bad token in header

200-const user = {

    "\_id": "5f816b77f0d7463f309aede5",

    "eMail": "email@gmail.com",

    "password": "$2b$10$2n2hdxRJcTakUilemBjKceVPvKovEL0PSqbfgWux7SOeyKcqiUI0.",

    "pregnanyStart": null,

    "photo": null,

    "firstName": null,

    "lastName": null,

    "age": null,

    "height": null,

    "weight": null

}

PUT /api/user

Header type: Authorization

Header content: ‘Bearer [accessToken]’

Body content: {

“propetryToChange”: “newValue”,

“anotherPropetryToChange”: “newValue”

}

Responses:

404-bad id

401 –No token in header

403 –Bad token in header

200-const updatedContent = {

    "firstName": "loremipsum"

}

## User document schema

const userSchema = {

    \_id: ObjectId,

    eMail: String,

    password: String,

    pregnanyStart: Date Object, *//JS date object*

    photo: String,

    firstName: String,

    lastName: String,

    age: Integer,

    height: Integer, *//[cm]*

    weight: Integer *//[cm]*

}