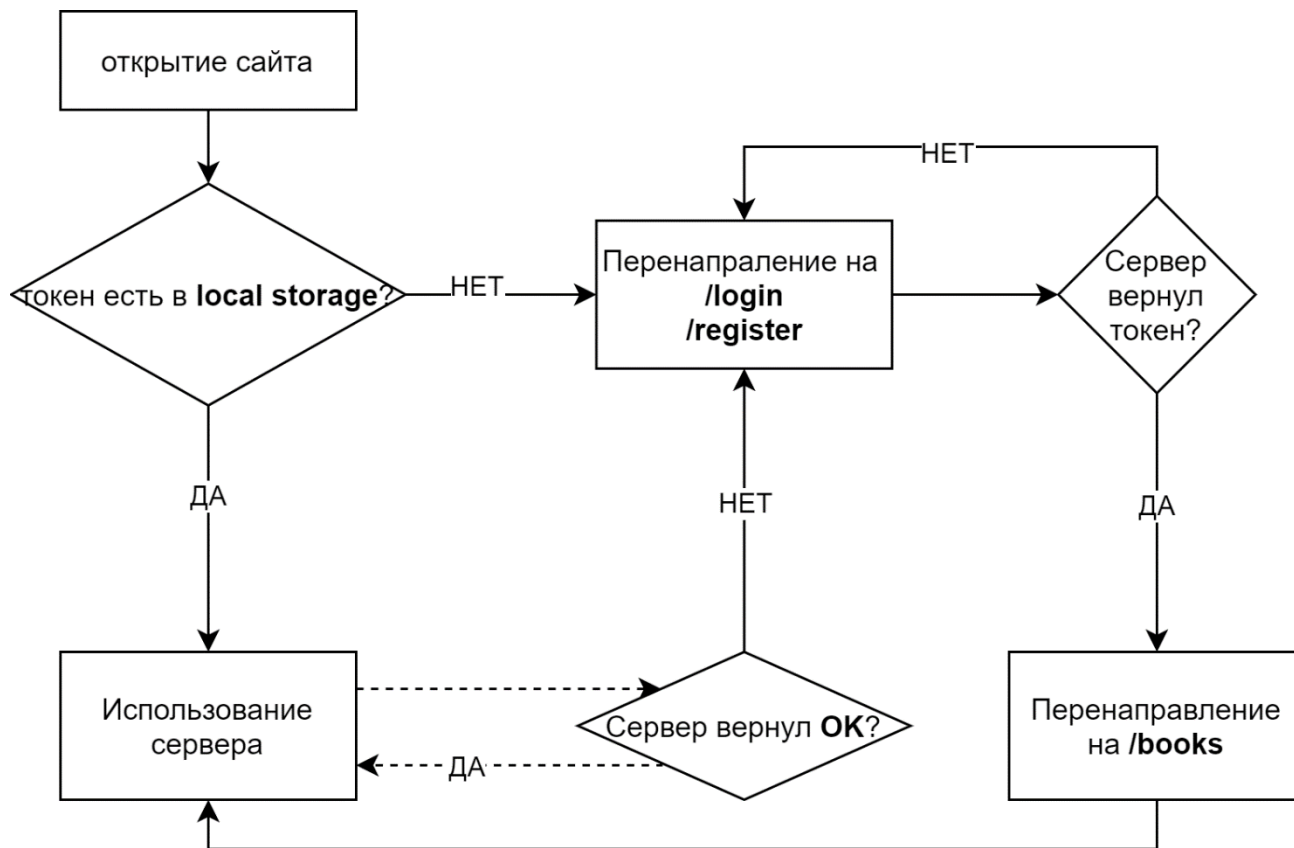


## Авторизация:

Схема авторизации:



## Описания API:

### Авторизация и работа с пользователями:

Контроллер: UserController.cs

Сервис: UserService.cs

### Регистрация:

POST /api/user/register

JSON Body:

```
{
  "name": "user",
  "password": "123"
}
```

### Логин:

POST /api/user/login

JSON Body

```
{
  "name": "user",
  "password": "123"
}
```

### Обновление токена:

POST /api/user/refresh

JSON Body

```
{
    "refreshToken": " token"
}
```

При успешном логине, регистрации или обновлении токена сервер возвращает пару токенов для дальнейшего доступа к Api.

Response:

```
{
    "refreshToken": " token "
    "accessToken": " token "
}
```

#### Работа с книгами:

Контроллер: BookController.cs

Сервис: BookService.cs

#### Создание и загрузка книги:

POST /api/book/create

Header: Authorization: Bearer { токен }

Multipart form Body:

Ключ	Значение
book_info	JSON: { "title": "Book Title", "author": "Book Author", "tags": "tag1, tag2 , tag3", "private": false }
file	File API

#### Обновление книги:

PUT /api/book/update

Header: Authorization: Bearer { токен }

Multipart form Body:

Ключ	Значение
book_info	JSON: { "title": "Book Title", "author": "Book Author", "tags": "tag1, tag2 , tag3", "private": false }
file	File API

#### Загрузка всех публичных книг:

GET /api/book/list

Header: Authorization: Bearer { токен }

#### Загрузка всех частных книг пользователя:

GET /api/book/listmy

Header: Authorization: Bearer { токен }

#### Загрузка файла книги:

GET /api/book/ download/{ ID книги }

Header: Authorization: Bearer { токен }

#### Удаление книги:

DELETE /api/book/delete/{ ID книги }

Header: Authorization: Bearer { токен }

При запуске сервер генерирует случайный ключ для подписи токенов. Access токены живут 60 секунд, а Refresh токены – 1 час.

Запросы **listmy**, **create**, **update**, **delete** требуют Id пользователя чтобы определить владельца книги. Идентификатор пользователя содержится в теле токена.

Структуры данных:

#### USER

```
{
  "id": { ID } ,
  "name": { Name } ,
  "password": " { Pass } "
```

#### BOOK

```
{
  "id": " { Book Id } ",
  "title": " { Book Title } ",
  "author": " { Book Author } ",
  "tags": [ " { tag1 } ", " { tag2 }" ... ],
  "private": { true / false }
```

## Приложение:

Интерфейс логина и регистрации:

Library

My books

Login

Register

Username

nikita

Password

pass123

Login

Login

Register

Username

nikita

Password

pass123

Password confirm

pass123

Register

Интерфейс публичных книг:

## Learn ASP.NET Core 3

Kenneth Yamikani Fukizi

Tags: [.net](#) [asp](#) [c#](#)

[Download](#)

## Mastering Ubuntu Server

Jay LaCroix

Tags: [linux](#) [ubuntu](#)

[Download](#)

Интерфейс книг, принадлежащих пользователю:

[Library](#) [My books](#)

Add book

### Learn ASP.NET Core 3

**Kenneth Yamikani Fukizi**  
Tags: [.net](#) [asp](#) [c#](#)  
[Delete](#) [Edit](#)

### Mastering Ubuntu Server

**Jay LaCroix**  
Tags: [linux](#) [ubuntu](#)  
[Delete](#) [Edit](#)

### My private book

**Nikita**  
Tags: [private](#) [book](#)  
[Delete](#) [Edit](#)

Интерфейс загрузки, обновления книги:

[Library](#) [My books](#)

## Upload book

Title

Author

Tags

☒ Private

File  
 Get Programming with Go.pdf