Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт
з лабораторної роботи №3 з дисципліни
«Сучасні технології розробки WEB-застосувань на платформі Microsoft.NET»

«Проектування REST веб-API»

Варіант 2

Виконав студент ІП-13 Дем'янчук Олександр Петрович
(шифр, прізвище, ім'я, по батькові)

Перевірив Бардін В.
(прізвище, ім'я, по батькові)

Київ 2023

Лабораторна робота №3

Варіант 2

Тема: Проектування REST веб-API.

Мета:

1. Ознайомитися з основами створення REST веб-API та методологією C4 для відображення архітектури системи.
2. Ознайомитися з основами створення ER-діаграм для представлення структури бази даних.

Постановка задачі

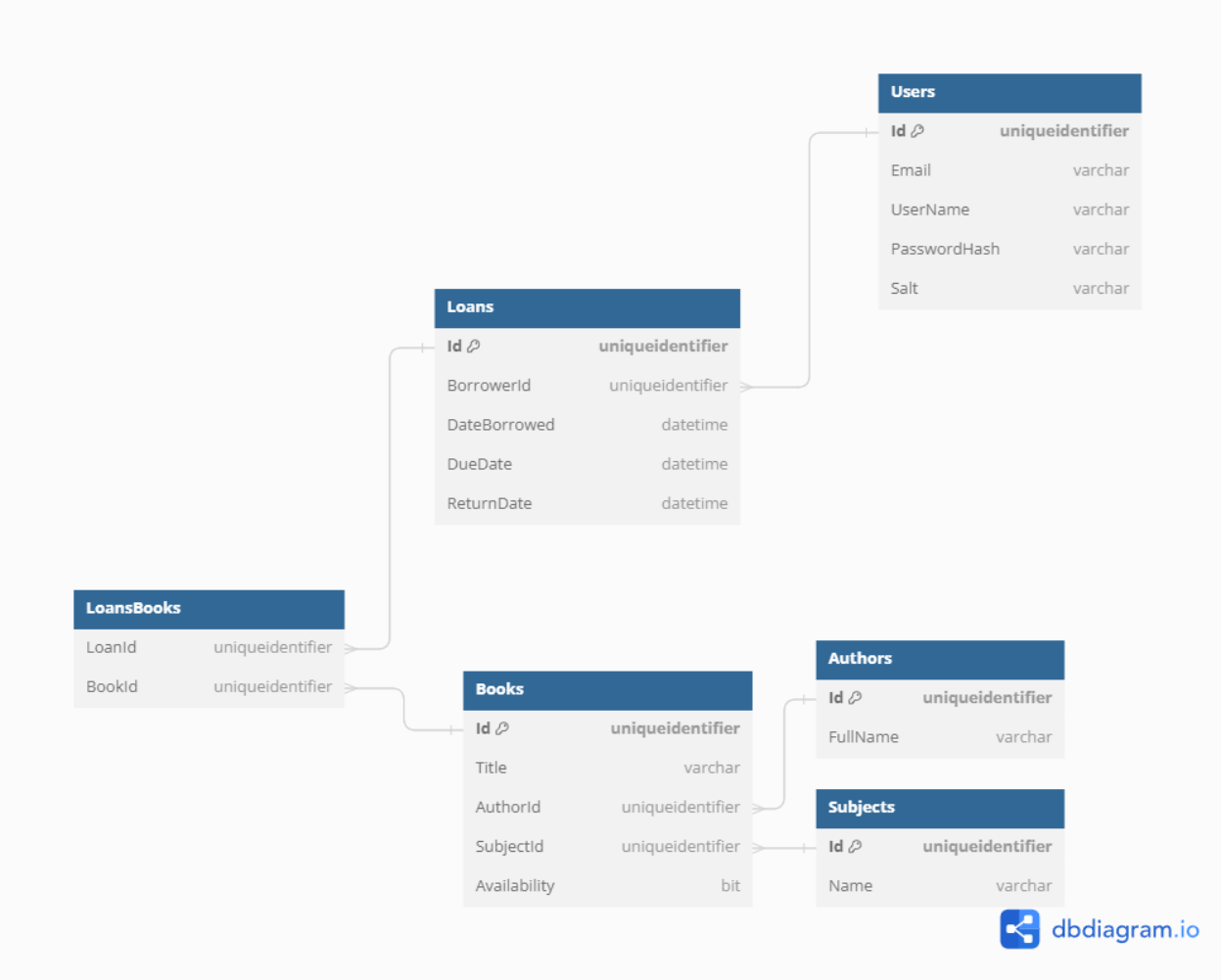
1. З дотриманням вимог REST-у спроектувати веб-API для обраної(згідно варіанту) доменної області, використовуючи методологію C4 для створення діаграми архітектури системи.
2. Створити ER-діаграму для DAL (Data Access Layer), яка відображатиме структуру бази даних веб-API.
3. Оформити спроєктоване рішення у вигляді звіту до лабораторної роботи.

Умова варіанту завдання №2:

2	Бібліотека. Облік формулярів.	1. Каталог літератури надає функції пошуку джерел за назвою, автором та тематикою. 2. Зареєстровані читачі мають можливість користуватись літературою за умови не більше 10 найменувань у формулярі та при наявності екземпляру у сховищі. **Функціональні вимоги:** 1. Ведення каталогу та керування; 2. Користуванням літературою.

Виконання завдань

ER-Діаграма:



Опис сутностей:

User – користувач системи, містить такі поля:

- Id – унікальний ідентифікатор
- Email – Електронна пошта
- Username – ім’я користувача в системі
- PasswordHash – захешований пароль
- Salt – сіль для хешування паролю

Book – сутність книги, містить такі поля:

- Id – унікальний ідентифікатор
- Title – назва книги
- AuthorId – ідентифікатор автора книги
- SubjectId – ідентифікатор тематики книги
- Availability – факт доступності книги в каталозі

Author – автор книги, сутність 3НФ, містить такі поля:

- Id – унікальний ідентифікатор
- FullName – повне ім’я автора

Subject – тематика книги, сутність 3НФ, містить такі поля:

- Id – унікальний ідентифікатор
- Name – назва тематики

Loan –формуляр в бібліотеці, містить такі поля:

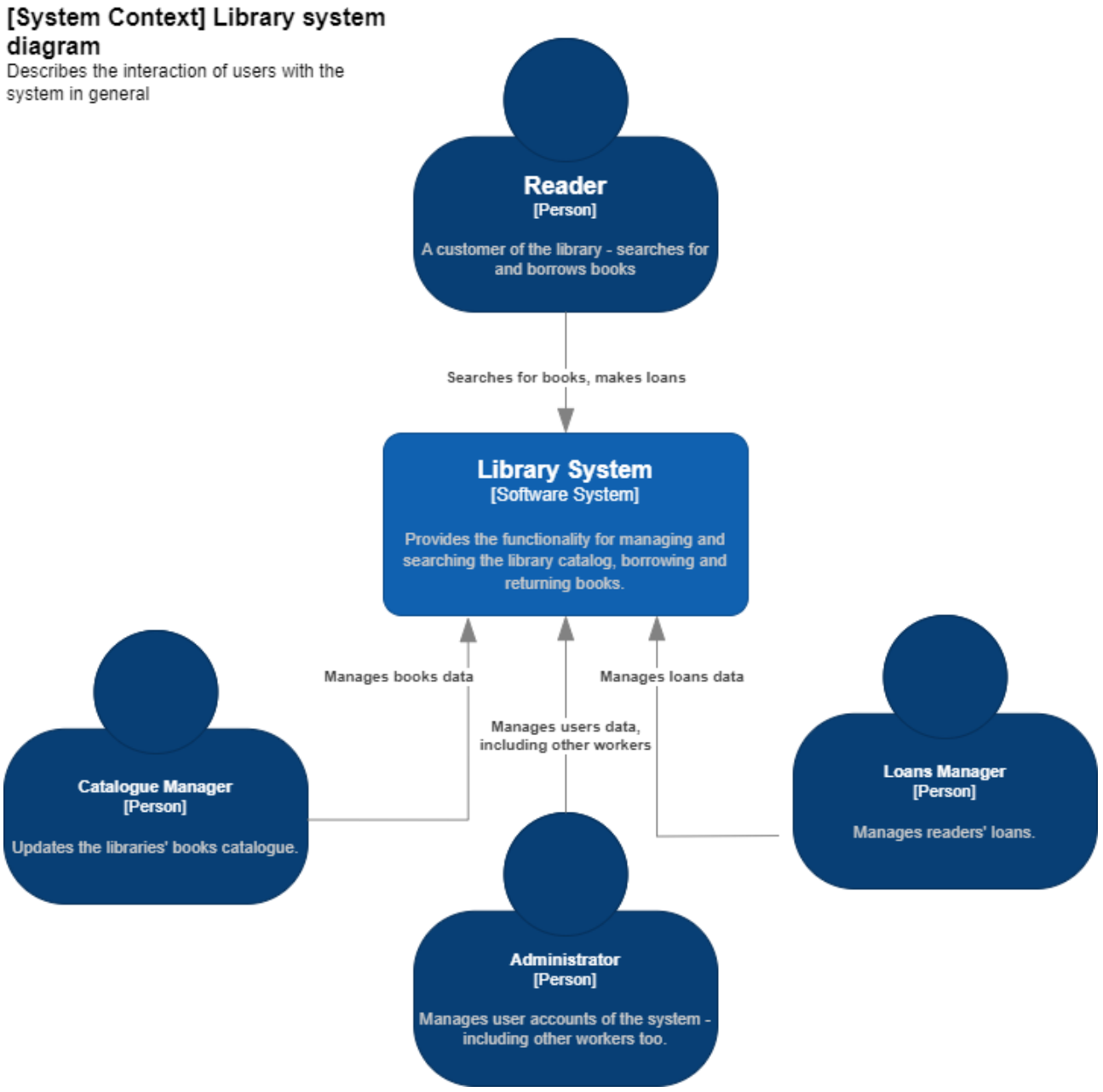
- Id – унікальний ідентифікатор
- BorrowerId – ідентифікатор користувача, що володіє формуляром
- DateBorrowed – дата, коли було оформлено формуляр
- DueDate – дата, до якої потрібно повернути формуляр без заборгованості
- ReturnDate – фактична дата повернення формуляру

LoanBook – таблиця M2M, що зв’язує формуляри та книги в каталозі, містить поля:

- LoanId – ідентифікатор формуляру
- BookId – ідентифікатор книги

Діаграма C4:

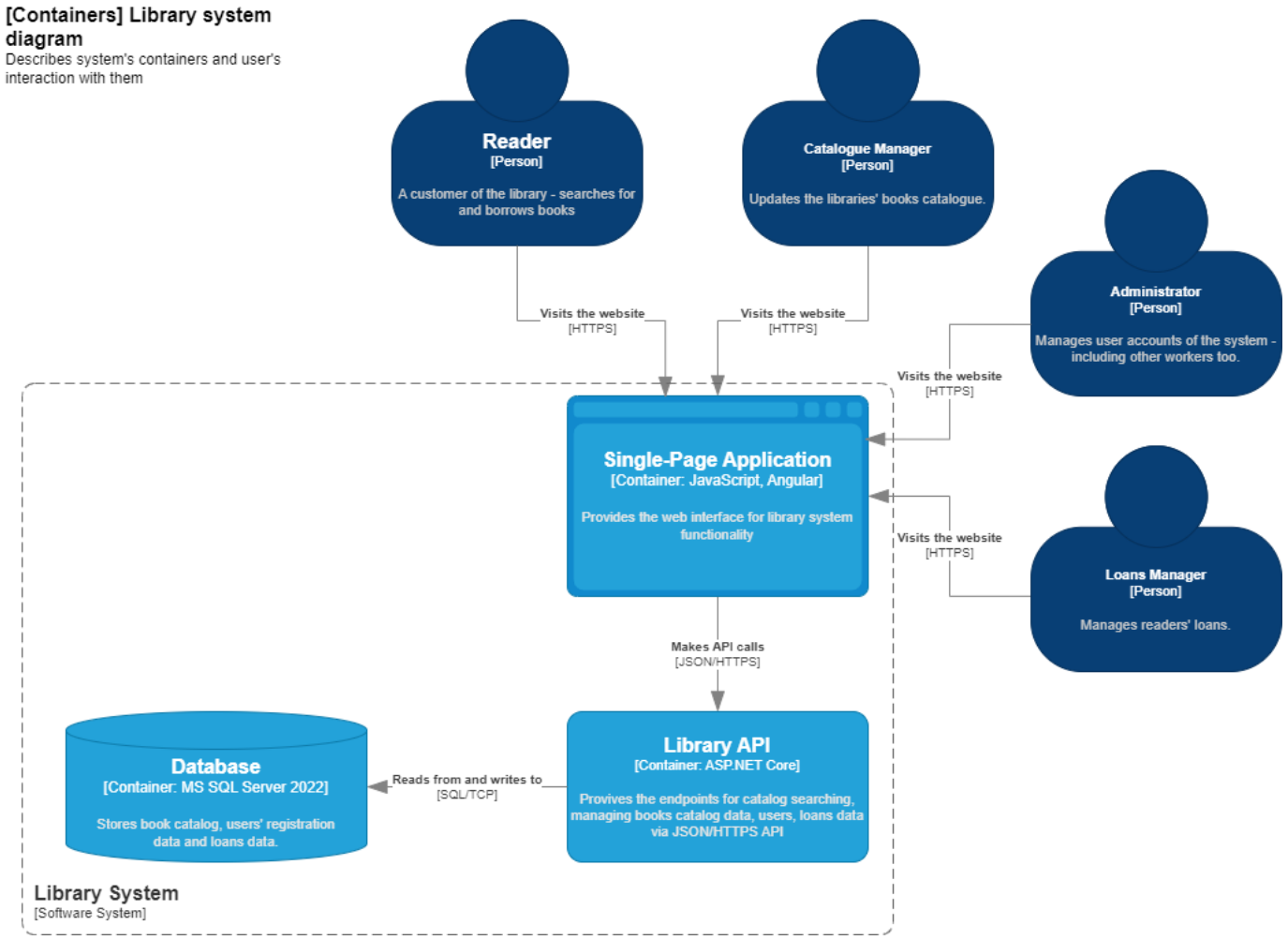
Рівень 1 – Контекст системи:



В системі визначено чотири види авторизованих користувачів:

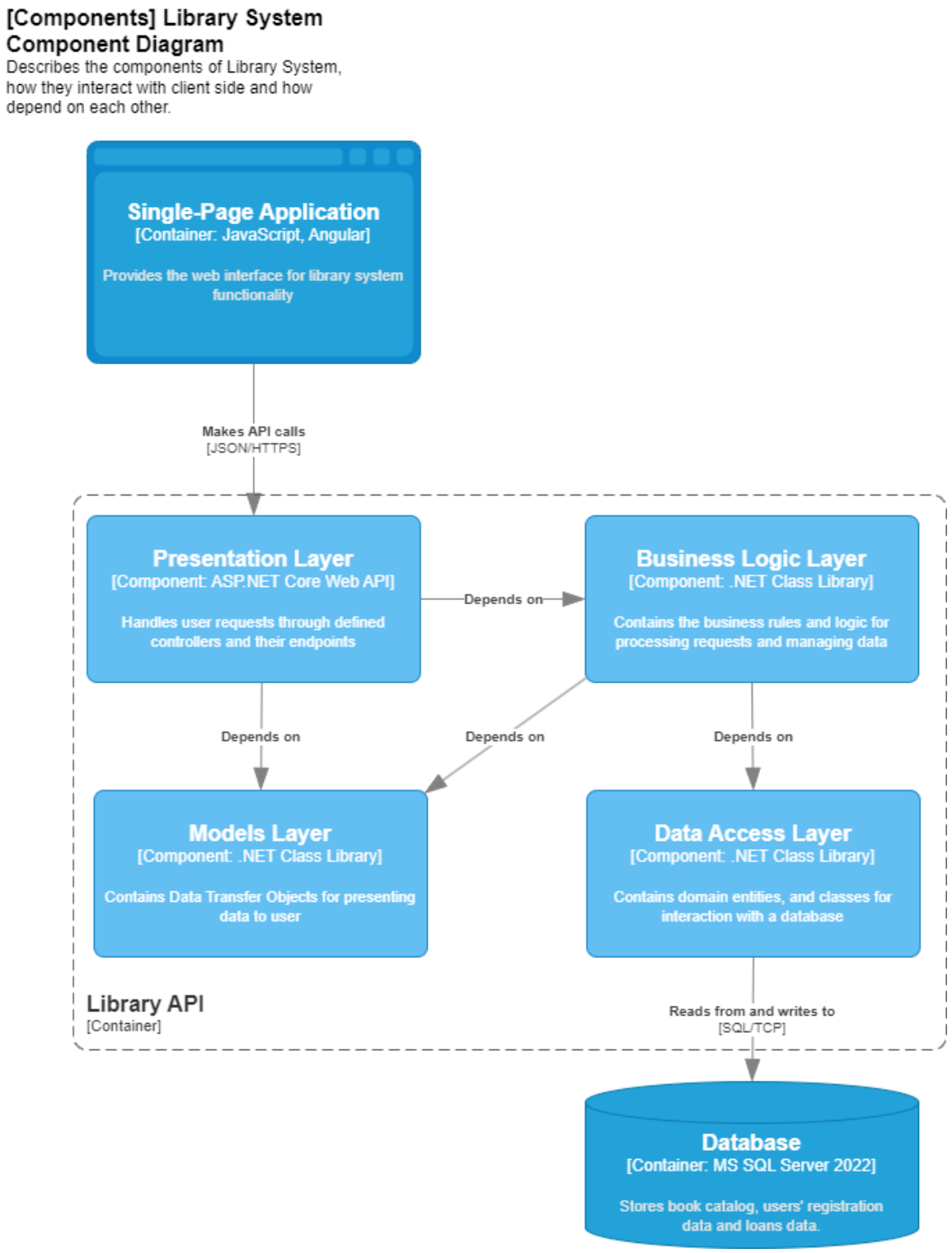
- Читач – клієнт бібліотеки, має можливість пошуку книг у каталозі та створення формулярів на них
- Менеджер каталогу – підтримує вміст каталогу книжок
- Адміністратор – управляє записами про акаунти усіх користувачів системи
- Менеджер формулярів – управляє формулярами в базі даних системи.

Рівень 2 – Контейнери:



Система складається з трьох контейнерів – фронтенд-додаток, завдяки якому усі користувачі мають зручний доступ до системи; API, яке обробляє запити користувачів з фронтенду; база даних, в якій містяться та оновлюються дані системи.

Рівень 3 – Компоненти:



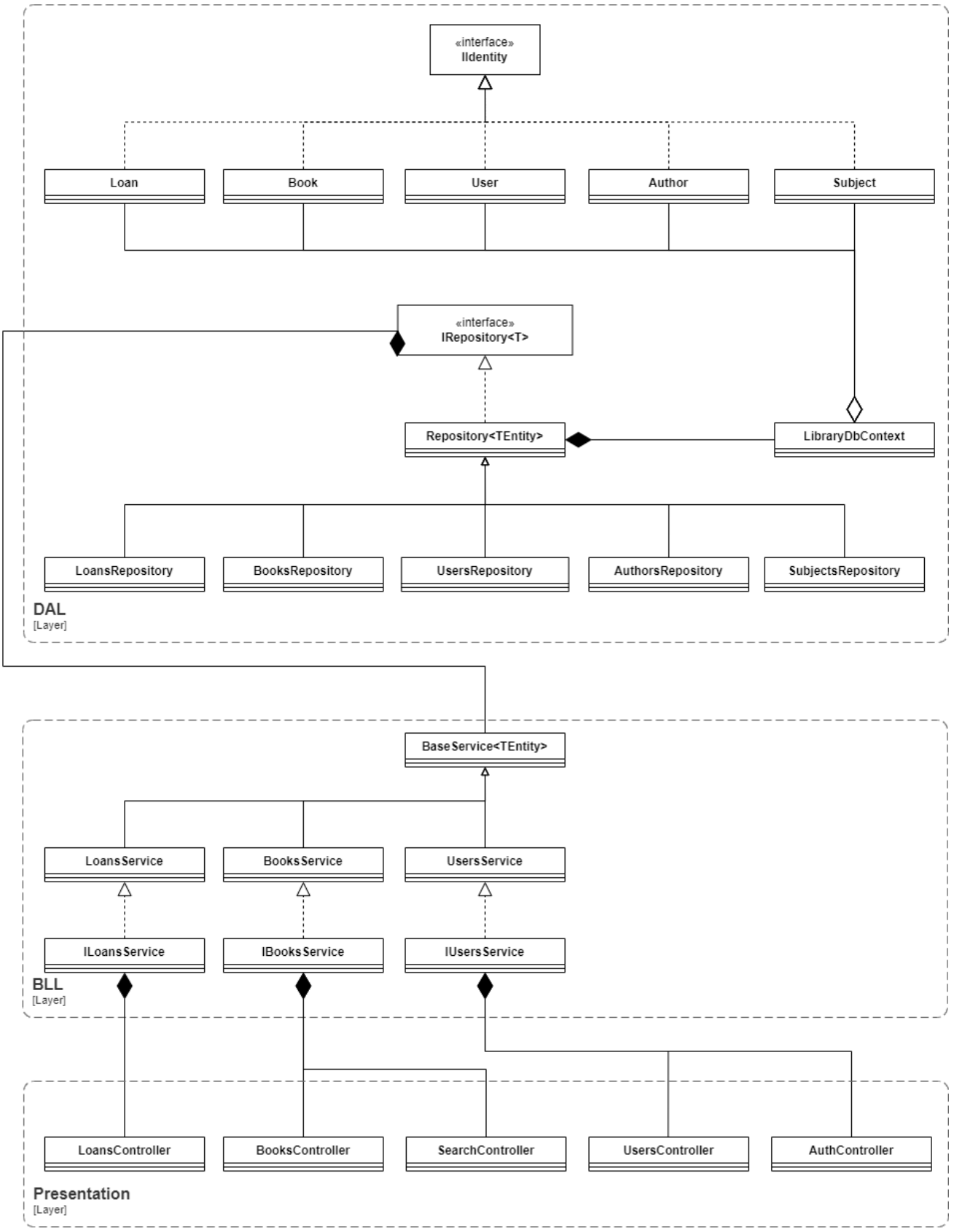
API системи було спроектовано за N-Layer (багатошаровою) архітектурою. Таких шарів в системі 4:

- Презентаційний: ендпоїнти API, через які проходять усі запити користувачів на взаємодію з системою
- Бізнес-логіка: містить бізнес-правила системи, логіку обробки запитів та управління даними
- Доступ до даних: містить доменні сутності, реалізацію

підключення до бази даних, та абстракції для взаємодії з іншими шарами системи

- Моделі – містить Data Transfer Objects, необхідні для передачі даних від користувачів до системи та між шарами системи

Рівень 4 – Діаграма коду:



Система містить інтерфейс для сутностей IIdentity, який гарантує, що сутності системи реалізують унікальний ідентифікатор, такий як GUID.

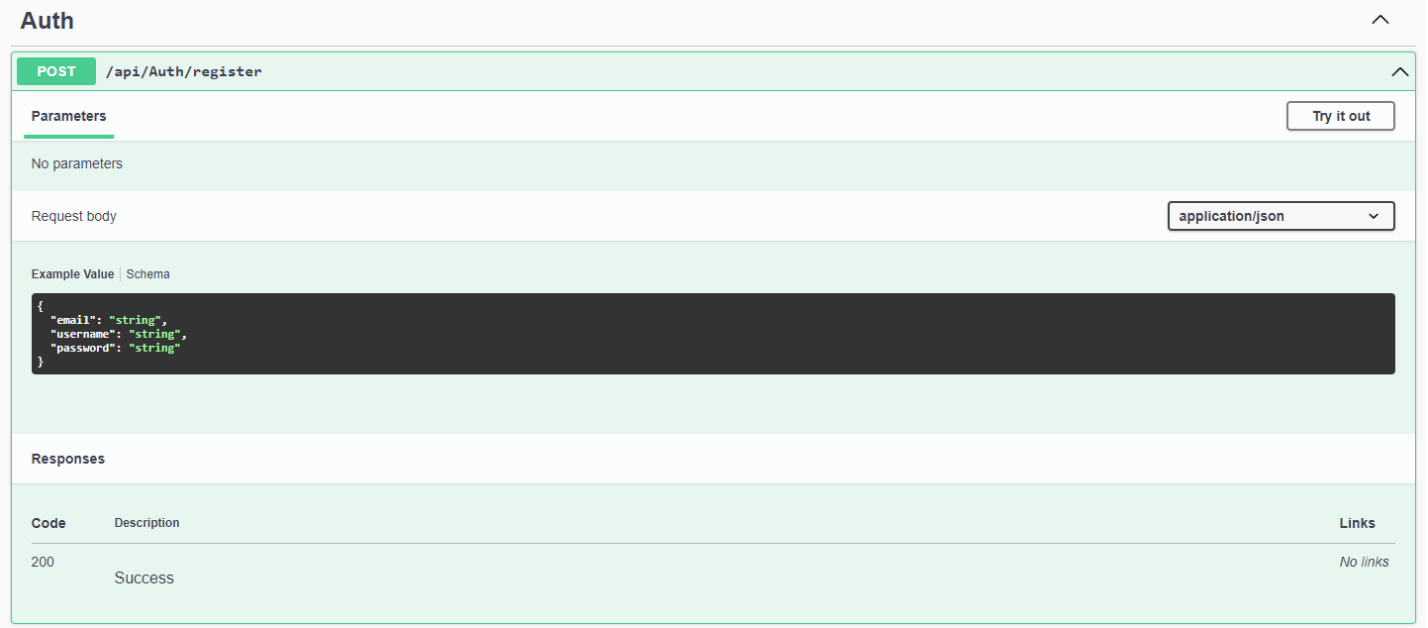
Для з’єднання з базою даних використуватиметься Entity Framework Core, тому маємо клас LibraryDbContext, загорнутий в абстракцію Generic Repository:

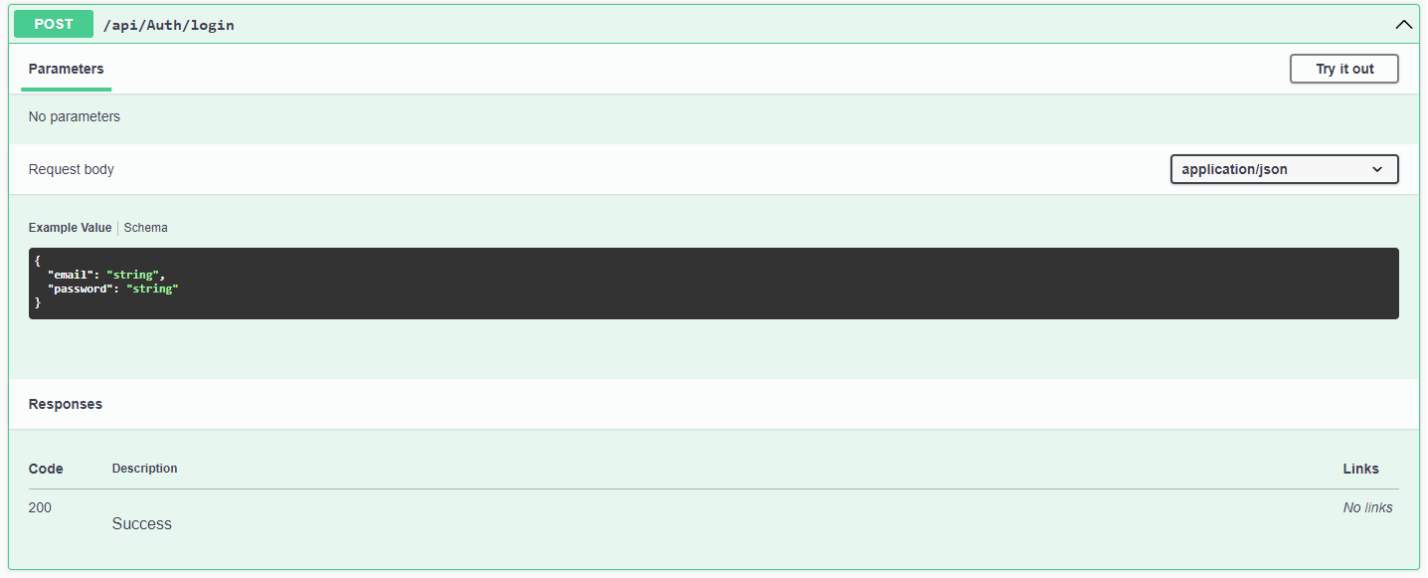
Інтерфейс IRepository<TEntity> декларує набір CRUD-операцій для сутностей типу TEntity, Reporitory<TEntity> реалізує ці операції з використанням LibraryDbContext, і від цього базового репозиторію наслідуються репозиторії для усіх сутностей системи.

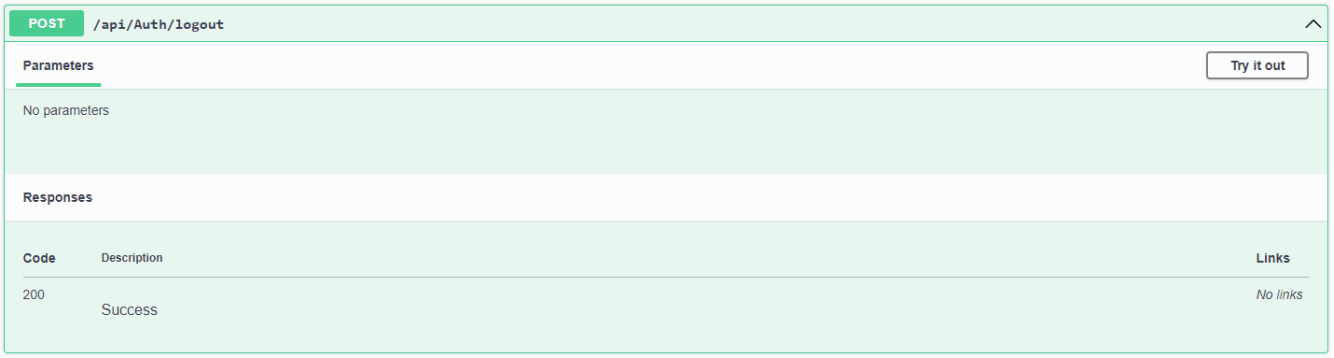
BaseService<TEntity> є базовим класом для сервісів в бізнес-логіці системи, декларує наявність репозиторію необхідних сутностей для усіх наслідуваних сервісів.

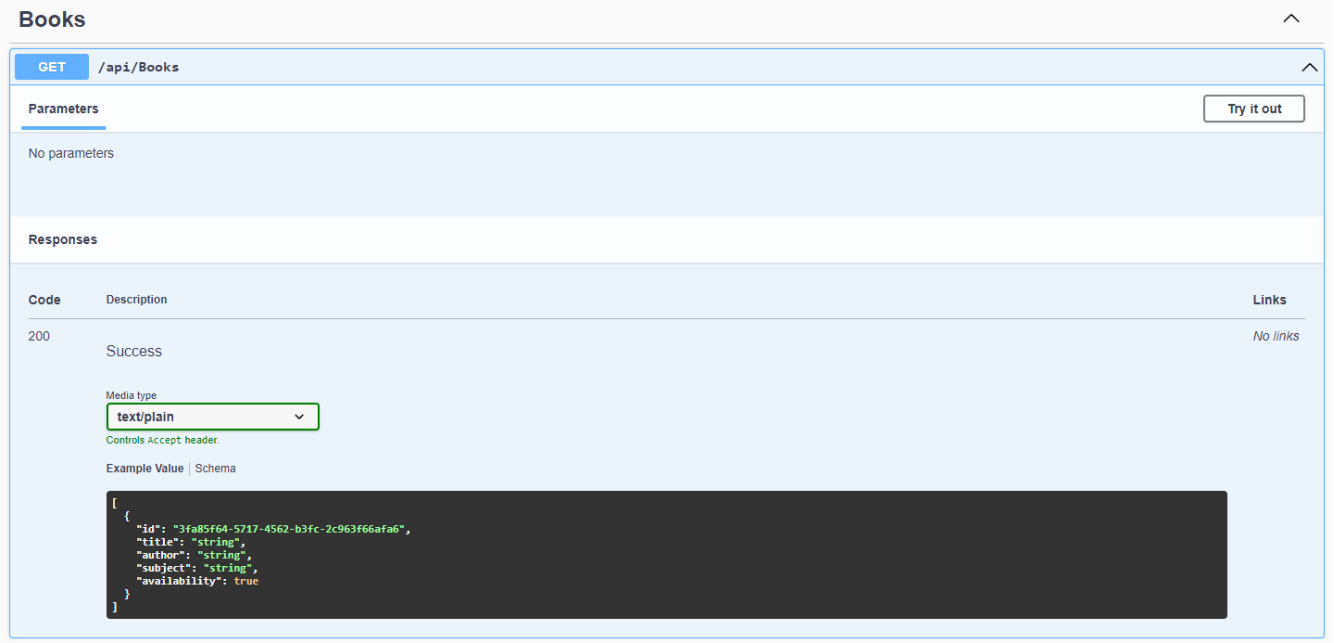
Сервіси бізнес-логіки обробляють запити з контролерів WebAPI.

Опис WebAPI:

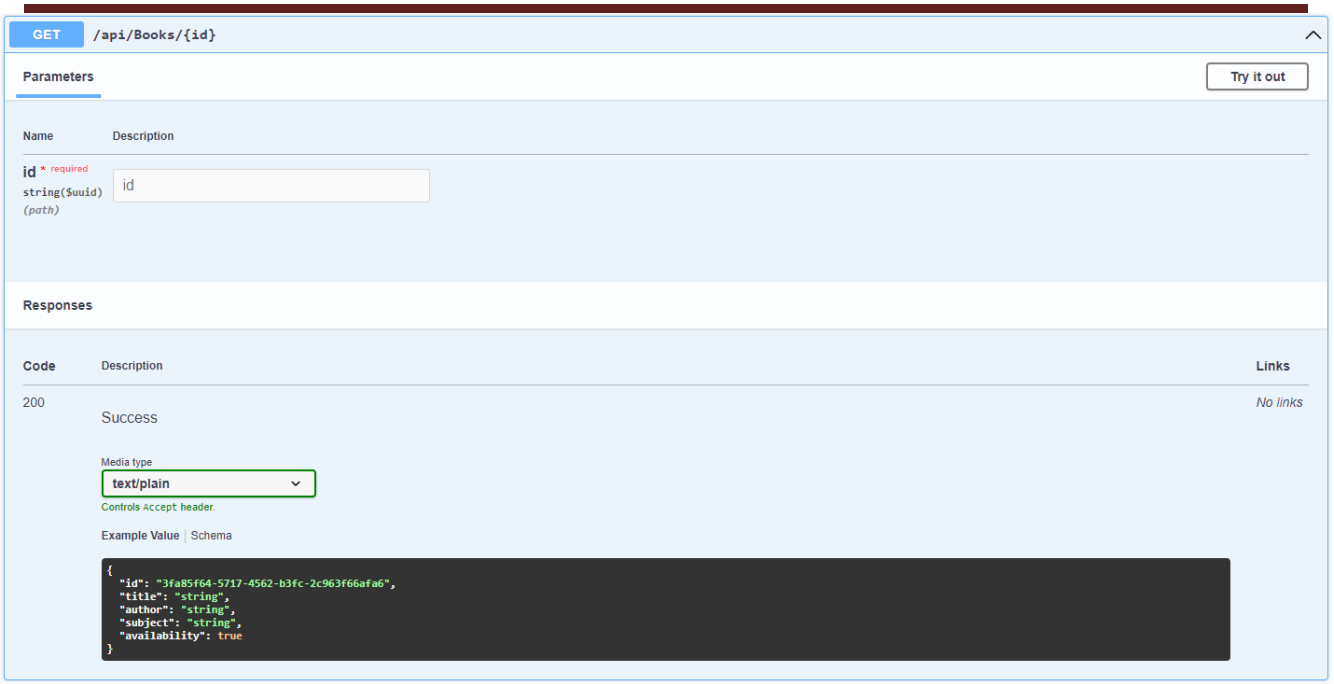


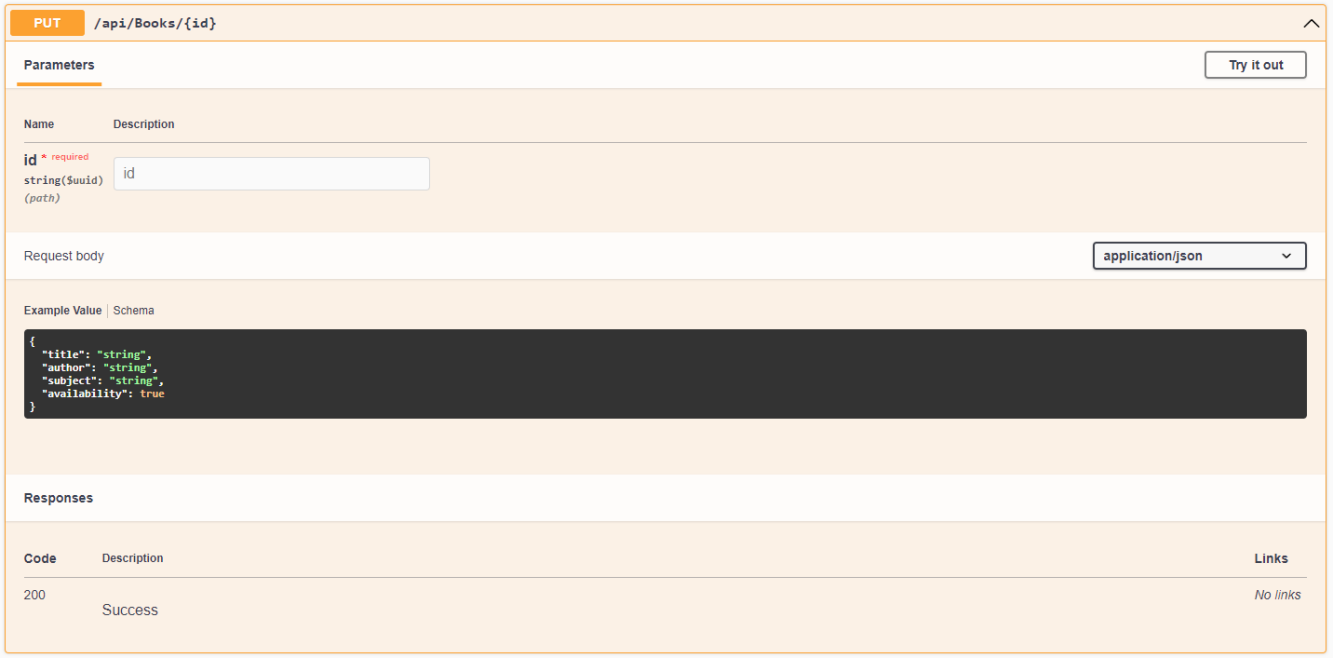


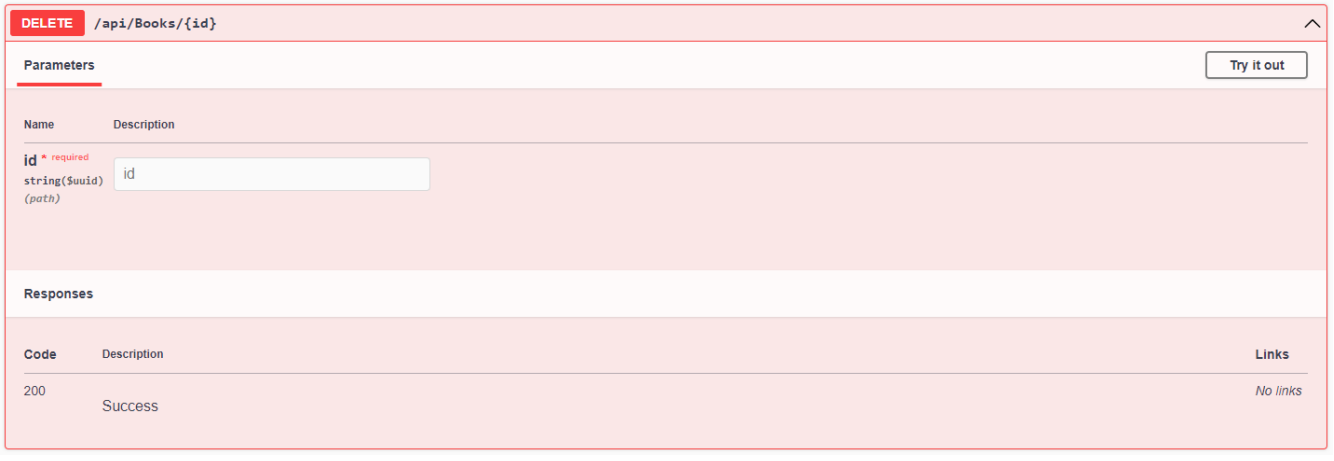

POST
/api/Auth/logout
Parameters
Try it out
No parameters
Responses
Code
Description
Links
200
Success
No links


Books
GET
/api/Books
Parameters
Try it out
No parameters
Responses
Code
Description
Links
200
Success
No links
Media type
text/plain
Controls Accept header.
Example Value | Schema
[
{
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"title": "string",
"author": "string",
"subject": "string",
"availability": true
}
]


POST
/api/Books
Parameters
Try it out
No parameters
Request body
application/json
Example Value | Schema
{
"title": "string",
"author": "string",
"subject": "string"
}
Responses
Code
Description
Links
200
Success
No links


GET
/api/Books/{id}
Parameters
Try it out
Name
Description
id * required
string($uuid)
(path)
id
Responses
Code
Description
Links
200
Success
No links
Media type
text/plain
Controls Accept header.
Example Value | Schema
{
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"title": "string",
"author": "string",
"subject": "string",
"availability": true
}


PUT
/api/Books/{id}
Parameters
Try it out
Name
Description
id * required
string($uuid)
(path)
id
Request body
application/json
Example Value | Schema
{
"title": "string",
"author": "string",
"subject": "string",
"availability": true
}
Responses
Code
Description
Links
200
Success
No links


DELETE
/api/Books/{id}
Parameters
Try it out
Name
Description
id * required
string($uuid)
(path)
id
Responses
Code
Description
Links
200
Success
No links

Loans

GET `/api/Loans`

Parameters | Try it out

No parameters

Responses

Code	Description	Links
200	Success Media type: text/plain Controls Accept header. Example Value | Schema	*No links*


```
[
  {
    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "borrower": {
      "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "email": "string",
      "username": "string"
    },
    "books": [
      {
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "title": "string",
        "author": "string",
        "subject": "string",
        "availability": true
      }
    ],
    "dateBorrowed": "2023-11-19T14:45:00.555Z",
    "dueDate": "2023-11-19T14:45:00.555Z",
    "returnDate": "2023-11-19T14:45:00.555Z"
  }
]
```


POST `/api/Loans`

Parameters | Try it out

No parameters

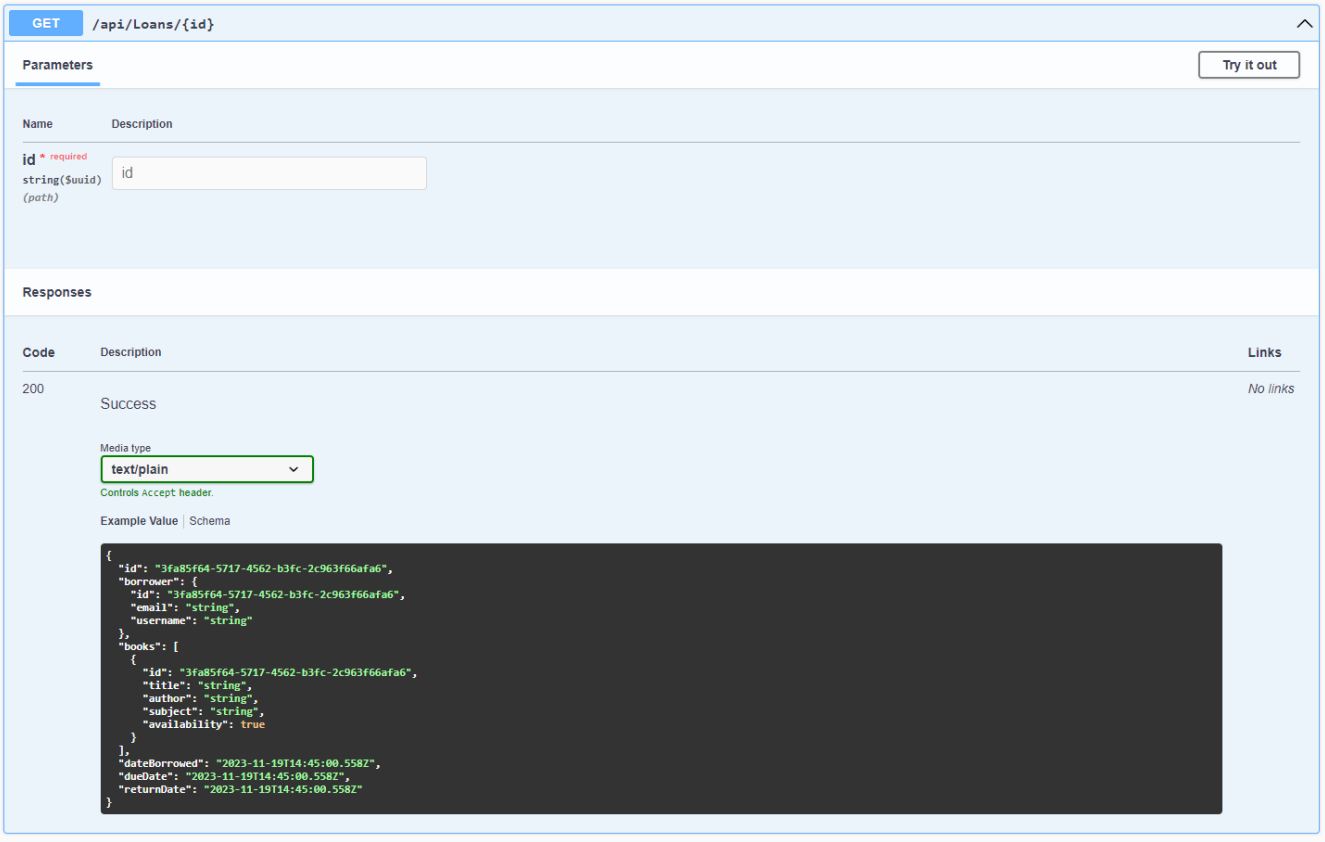
Request body: application/json

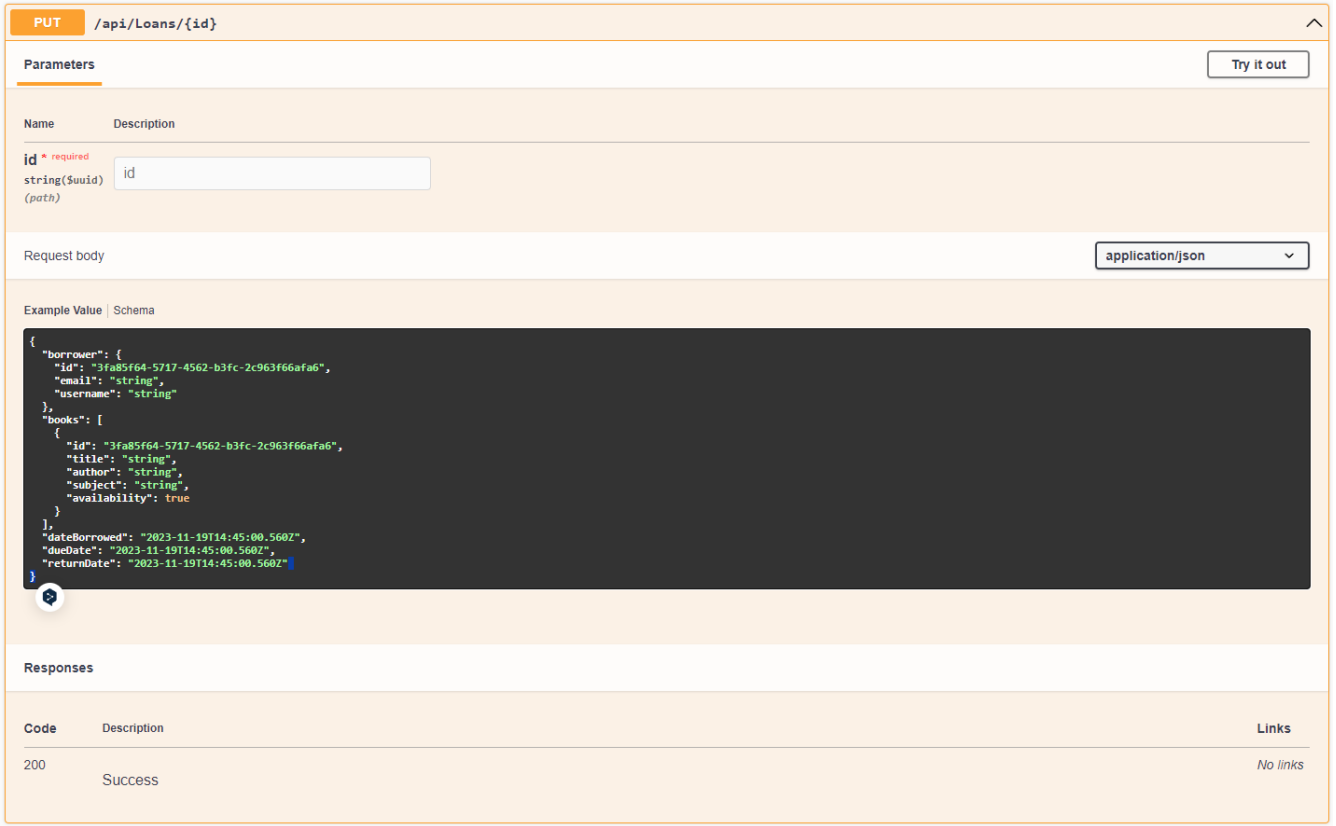
Example Value | Schema

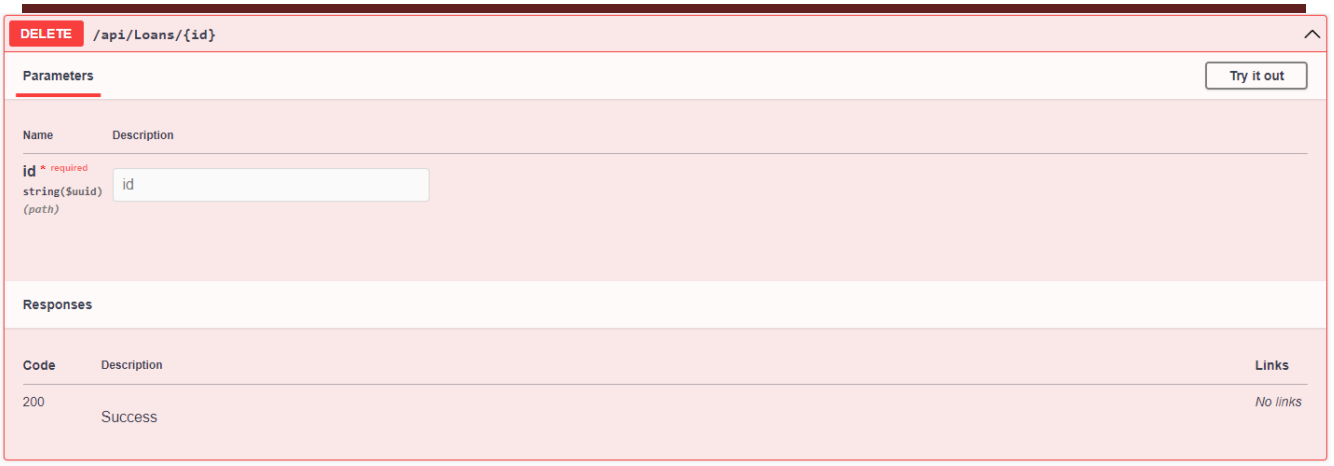

```
{
  "borrower": {
    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "email": "string",
    "username": "string"
  },
  "booksIds": [
    "3fa85f64-5717-4562-b3fc-2c963f66afa6"
  ]
}
```

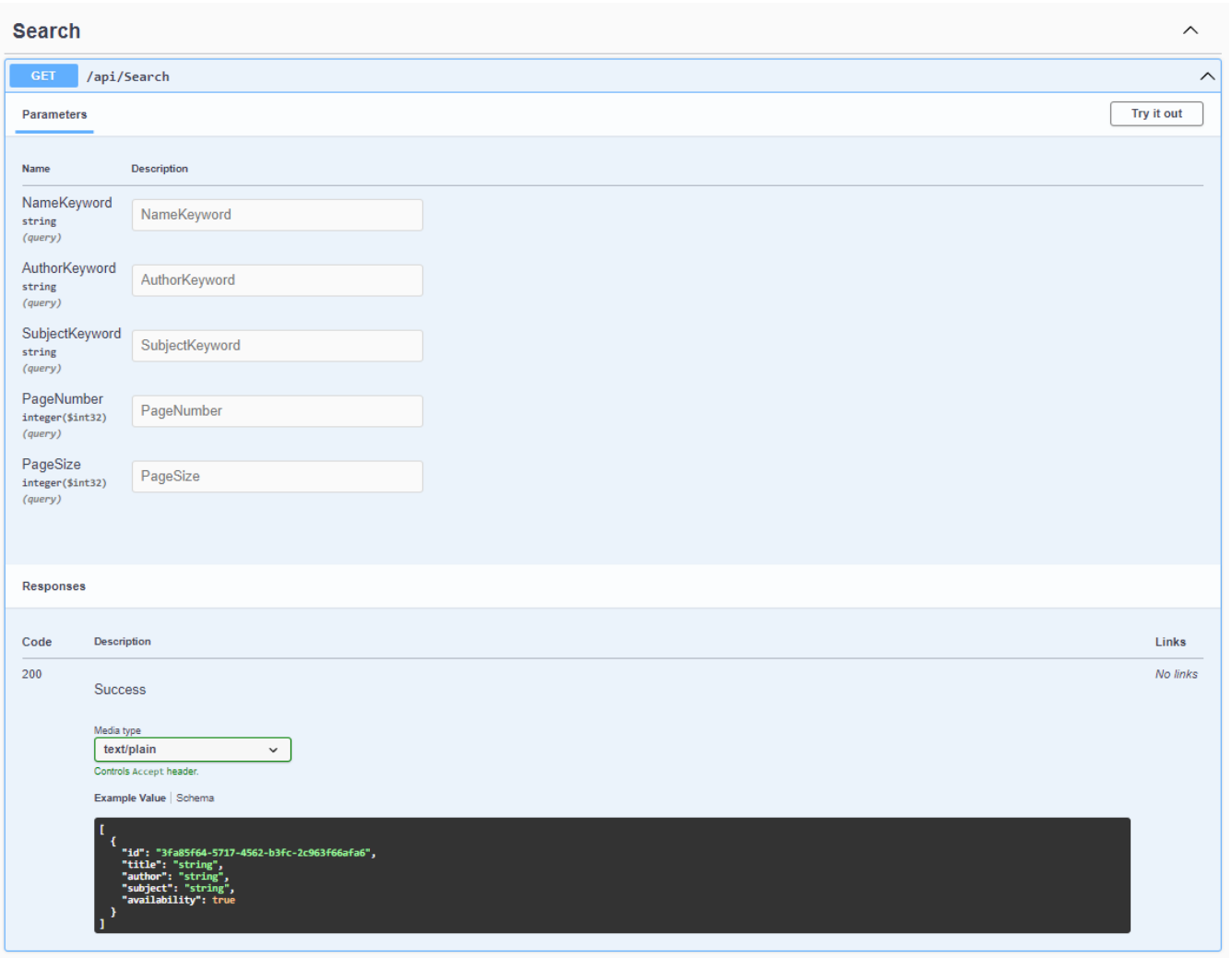

Responses

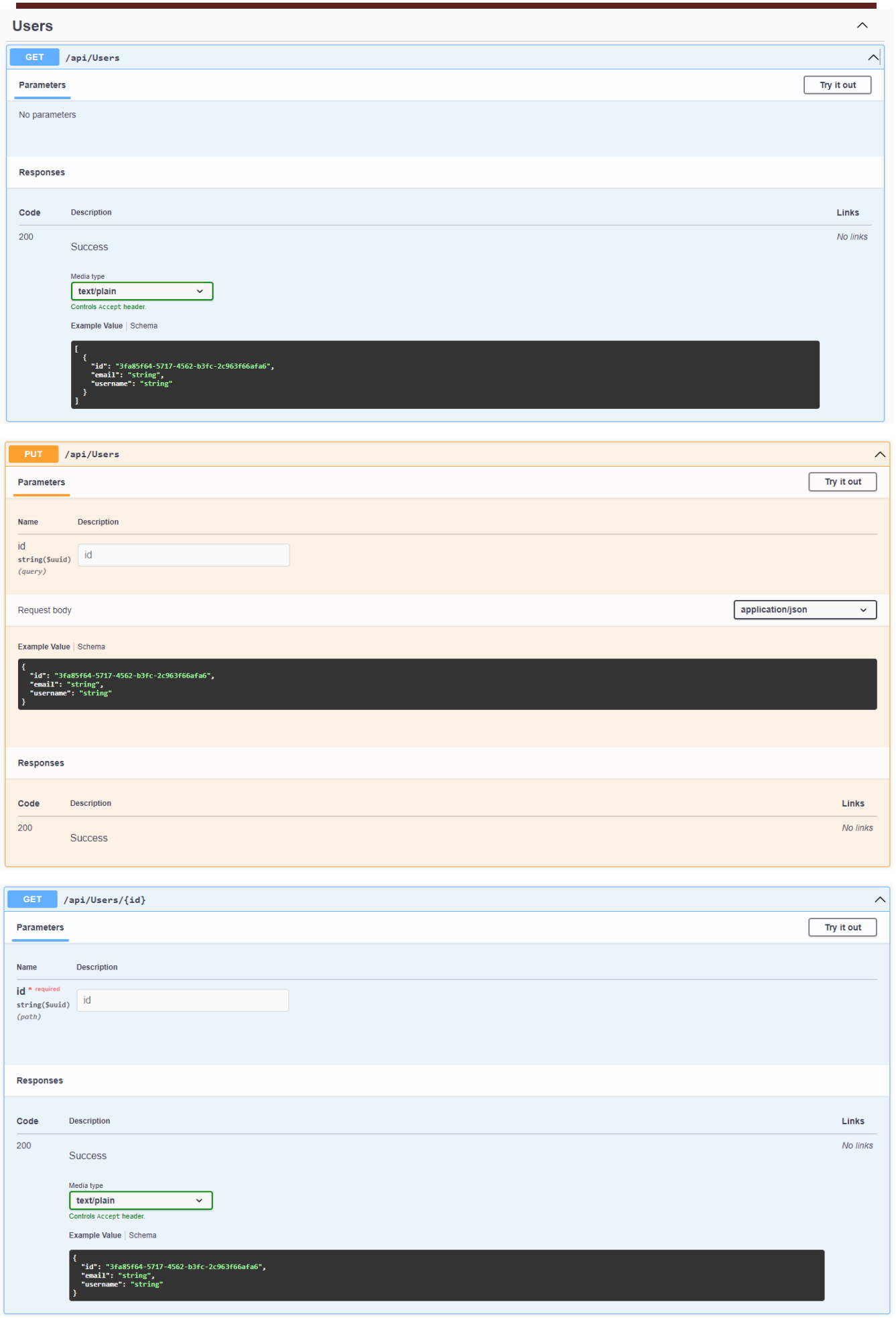
Code	Description	Links
200	Success	*No links*

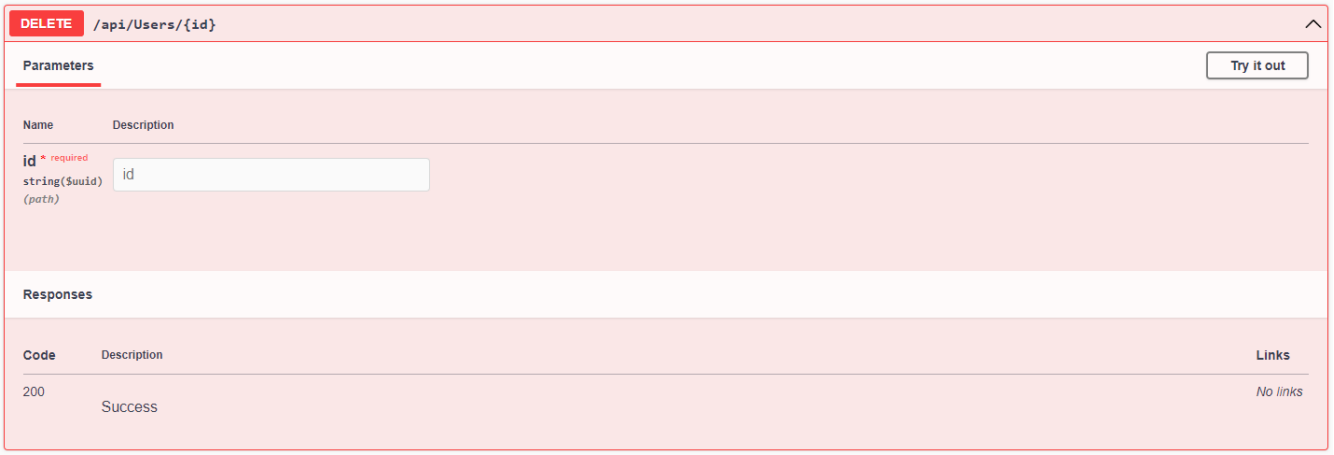

GET
/api/Loans/{id}
Parameters
Try it out
Name
Description
id * required
string($uuid)
(path)
id
Responses
Code
Description
Links
200
Success
No links
Media type
text/plain
Controls Accept header.
Example Value | Schema
{
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"borrower": {
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"email": "string",
"username": "string"
},
"books": [
{
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"title": "string",
"author": "string",
"subject": "string",
"availability": true
}
],
"dateBorrowed": "2023-11-19T14:45:00.558Z",
"dueDate": "2023-11-19T14:45:00.558Z",
"returnDate": "2023-11-19T14:45:00.558Z"
}


PUT
/api/Loans/{id}
Parameters
Try it out
Name
Description
id * required
string($uuid)
(path)
id
Request body
application/json
Example Value | Schema
{
"borrower": {
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"email": "string",
"username": "string"
},
"books": [
{
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"title": "string",
"author": "string",
"subject": "string",
"availability": true
}
],
"dateBorrowed": "2023-11-19T14:45:00.560Z",
"dueDate": "2023-11-19T14:45:00.560Z",
"returnDate": "2023-11-19T14:45:00.560Z"
}
Responses
Code
Description
Links
200
Success
No links


DELETE
/api/Loans/{id}
Parameters
Try it out
Name
Description
id * required
string($uuid)
(path)
id
Responses
Code
Description
Links
200
Success
No links


Search
GET
/api/Search
Parameters
Try it out
Name
Description
NameKeyword
string
(query)
NameKeyword
AuthorKeyword
string
(query)
AuthorKeyword
SubjectKeyword
string
(query)
SubjectKeyword
PageNumber
integer($int32)
(query)
PageNumber
PageSize
integer($int32)
(query)
PageSize
Responses
Code
Description
Links
200
Success
No links
Media type
text/plain
Controls Accept header.
Example Value | Schema
[
{
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"title": "string",
"author": "string",
"subject": "string",
"availability": true
}
]


Users
GET /api/Users
Parameters
Try it out
No parameters
Responses
Code
Description
Links
200
Success
No links
Media type
text/plain
Controls Accept header.
Example Value | Schema
[
{
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"email": "string",
"username": "string"
}
]
PUT /api/Users
Parameters
Try it out
Name
Description
id
string($uuid)
(query)
id
Request body
application/json
Example Value | Schema
{
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"email": "string",
"username": "string"
}
Responses
Code
Description
Links
200
Success
No links
GET /api/Users/{id}
Parameters
Try it out
Name
Description
id * required
string($uuid)
(path)
id
Responses
Code
Description
Links
200
Success
No links
Media type
text/plain
Controls Accept header.
Example Value | Schema
{
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"email": "string",
"username": "string"
}



Schemas

BookCreatedDto {
- title: string, nullable: true
- author: string, nullable: true
- subject: string, nullable: true

}

BookDto {
- id: string($uuid)
- title: string, nullable: true
- author: string, nullable: true
- subject: string, nullable: true
- availability: boolean

}

BookUpdatedDto {
- title: string, nullable: true
- author: string, nullable: true
- subject: string, nullable: true
- availability: boolean

}

LoanCreatedDto {
- borrower: **UserDto** {
  - id: string($uuid)
  - email: string, nullable: true
  - username: string, nullable: true

  }
- booksIds: [ ] nullable: true

}

```
LoanDto ⌄ {
    id                  string($uuid)
    borrower            UserDto ⌄ {
                            id                  string($uuid)
                            email               string
                                                nullable: true
                            username            string
                                                nullable: true
                        }
    books                ⌄ [
                        nullable: true
                        BookDto › {...}]
    dateBorrowed        string($date-time)
    dueDate             string($date-time)
    returnDate          string($date-time)
                        nullable: true
}
```

```
LoanUpdatedDto ⌄ {
    borrower            UserDto ⌄ {
                            id                  › [...]
                            email               › [...]
                            username            › [...]
                        }
    books                ⌄ [
                        nullable: true
                        BookDto ⌄ {
                            id                  string($uuid)
                            title               string
                                                nullable: true
                            author              string
                                                nullable: true
                            subject             string
                                                nullable: true
                            availability        boolean
                         }]
    dateBorrowed        string($date-time)
    dueDate             string($date-time)
    returnDate          string($date-time)
}
```

Висновки:

Висновок: В ході виконання лабораторної роботи №3 було спроектовано WebAPI-систему для бібліотеки, побудувавши діаграму C4, ER-діаграму та спроектувавши документацію API Swagger json.