

Microservices Documentation

Book Service

Base URL: `http://localhost:8081`

Endpoints:

GET `/api/Books`:

Retrieves all books.

Response:

```
[  
  {  
    "id": "integer",  
    "name": "string",  
    "author": "string",  
    "quantity": "integer"  
  }  
]
```

POST `/api/Books`:

Creates a new book.

Response:

```
{  
  "id": "integer",  
  "name": "string",  
  "author": "string",  
  "quantity": "integer"  
}
```

GET /api/Books/{id}:

Retrieves a specific book by its ID.

Response:

```
{  
  "id": "integer",  
  "name": "string",  
  "author": "string",  
  "quantity": "integer"  
}
```

PUT /api/Books/{id}:

Updates a book by its ID.

Response:

```
{  
  "id": "integer",  
  "name": "string",  
  "author": "string",  
  "quantity": "integer"  
}
```

DELETE /api/Books/{id}:

Deletes a book by its ID.

Response: 200 OK

DTO Structure:

```
{  
  "id": "integer",  
  "name": "string",
```

```
"author": "string",  
"quantity": "integer"  
}
```

User Service

Base URL: <http://localhost:8082>

Endpoints:

GET /api/users:

Retrieves all users.

Response:

```
[  
  {  
    "id": "integer",  
    "name": "string",  
    "email": "string"  
  }  
]
```

POST /api/users:

Creates a new user.

Response:

```
{  
  "id": "integer",  
  "name": "string",  
  "email": "string"  
}
```

GET /api/users/{id}:

Retrieves a user by their ID.

Response:

```
{  
  "id": "integer",  
  "name": "string",  
  "email": "string"  
}
```

PUT /api/users/{id}:

Updates a user by their ID.

Response:

```
{  
  "id": "integer",  
  "name": "string",  
  "email": "string"  
}
```

DELETE /api/users/{id}:

Deletes a user by their ID.

Response: 200 OK

DTO Structure:

```
{  
  "id": "integer",  
  "name": "string",  
  "email": "string"  
}
```

Borrowing Service

Base URL: `http://localhost:8083`

Endpoints:

GET `/borrows`:

Retrieves all borrowing records.

Response:

```
[  
  {  
    "id": "integer",  
    "userId": "integer",  
    "bookId": "integer",  
    "borrowDate": "string",  
    "returnDate": "string",  
    "returned": "boolean"  
  }  
]
```

POST `/borrows`:

Creates a new borrowing record.

Response:

```
{  
  "id": "integer",  
  "userId": "integer",  
  "bookId": "integer",  
  "borrowDate": "string",  
  "returnDate": "string",  
  "returned": "boolean"  
}
```

```
}
```

GET /borrows/{id}:

Retrieves a specific borrowing record by its ID.

Response:

```
{  
  "id": "integer",  
  "userId": "integer",  
  "bookId": "integer",  
  "borrowDate": "string",  
  "returnDate": "string",  
  "returned": "boolean"  
}
```

PUT /borrows/{id}:

Updates a borrowing record by its ID.

Response:

```
{  
  "id": "integer",  
  "userId": "integer",  
  "bookId": "integer",  
  "borrowDate": "string",  
  "returnDate": "string",  
  "returned": "boolean"  
}
```

PUT /borrows/{id}/return:

Marks a borrowing record as returned.

Response:

```
{  
  "id": "integer",  
  "userId": "integer",  
  "bookId": "integer",  
  "borrowDate": "string",  
  "returnDate": "string",  
  "returned": "boolean"  
}
```

DELETE /borrows/{id}:

Deletes a borrowing record by its ID.

Response: 200 OK

DTO Structure:

```
{  
  "id": "integer",  
  "userId": "integer",  
  "bookId": "integer",  
  "borrowDate": "string",  
  "returnDate": "string",  
  "returned": "boolean"  
}
```

Notification Service

Base URL: <http://localhost:8084>

Endpoints:

GET /api/notifications:

Retrieves all notifications.

Response:

```
[  
  {  
    "id": "integer",  
    "message": "string",  
    "recipient": "string"  
  }  
]
```

POST /api/notifications:

Creates a new notification.

Response:

```
{  
  "id": "integer",  
  "message": "string",  
  "recipient": "string"  
}
```

GET /api/notifications/{id}:

Retrieves a specific notification by its ID.

Response:

```
{  
  "id": "integer",  
  "message": "string",  
  "recipient": "string"  
}
```



```
}
```

PUT /api/notifications/{id}:

Updates a notification by its ID.

Response:

```
{  
  "id": "integer",  
  "message": "string",  
  "recipient": "string"  
}
```

DELETE /api/notifications/{id}:

Deletes a notification by its ID.

Response: 200 OK

DTO Structure:

```
{  
  "id": "integer",  
  "message": "string",  
  "recipient": "string"  
}
```