

## User Model

Name	Type	Comment
firstName	String	User first name
lastName	String	User last name
id	UUID	User id

## Create User

### Endpoint

POST /api/users

### Descriptions

Creates user from a given payload and returns created user model

### Payload

Name	Type	Comment
firstName	String	User first name
lastName	String	User last name

### Response

Response code (HTTP status)	Comment
201	User has been created. Returns user model
400	User has not been created because of wrong input

## Update User

### Endpoint

PUT /api/users/<id>

### Description

Updates user with given ID and returns updated user model

### Payload

Name	Type	Comment
firstName	String	User first name
lastName	String	User last name

### Response

Response code (HTTP status)	Comment
200	User has been updated. Returns user model
400	User has not been updated because of wrong input
404	User with given ID not found

## Get User

### Endpoint

GET /api/users/<id>

### Description

Gets user with given ID

### Response

Response code (HTTP status)	Comment
200	User has been returned
404	User with given ID not found

## Get User Collection

### Endpoint

GET /api/users?maxResults=<maxResult>&offset=<offset>&orderBy=<orderBy>&order=<order>

### Description

Gets user collection, supports pagination

### Query Parameters

Name	Type	Comment
maxResults	Integer	Max number of users in collection. Default value is 50
offset	Integer	Offset for the first user in collection. Default value is 0
orderBy	String	Which field to use for sorting
order	String	Possible values: 'asc' or 'desc'. Ascending or descending

### Response

Response code (HTTP status)	Comment
200	Returns JSON structure containing 2 fields: <ul style="list-style-type: none"><li>- total (Integer) – total number of users in the system</li><li>- values (list of users) – list of requested and sorted users according to pagination parameters</li></ul>

## Book Model

Name	Type	Comment
authorName	String	Full name of the author
bookTitle	String	Book title
id	UUID	Book id

## Create Book

### Endpoint

POST /api/books

### Descriptions

Creates book from a given payload and returns created book model

### Payload

Name	Type	Comment
author	String	Full name of the author
bookTitle	String	Book title

### Response

Response code (HTTP status)	Comment
201	Book has been created. Returns book model
400	Book has not been created because of wrong input

## Update Book

### Endpoint

PUT /api/books/<id>

### Description

Updates book with given ID and returns updated book model

### Payload

Name	Type	Comment
authorName	String	Full name of the author
bookTitle	String	Book title

### Response

Response code (HTTP status)	Comment
200	Book has been updated. Returns book model
400	Book has not been updated because of wrong input
404	Book with given ID not found

## Get Book

### Endpoint

GET /api/books/<id>

### Description

Gets book with given ID

### Response

Response code (HTTP status)	Comment
200	Book has been returned
404	Book with given ID not found

## Get Book Collection

### Endpoint

GET /api/books?maxResults=<maxResult>&offset=<offset>&orderBy=<orderBy>&order=<order>

### Description

Gets book collection, supports pagination

### Query Parameters

Name	Type	Comment
maxResults	Integer	Max number of books in collection. Default value is 20
offset	Integer	Offset for the first book in collection. Default value is 0
orderBy	String	Which field to use for sorting
order	String	Possible values: 'asc' or 'desc'. Ascending or descending

### Response

Response code (HTTP status)	Comment
200	Returns JSON structure containing 2 fields: <ul style="list-style-type: none"><li>- total (Integer) – total number of books in the system</li><li>- values (list of books) – list of requested and sorted books according to pagination parameters</li></ul>

## Card Model

Name	Type	Comment
userId	UUID	User id number
[ bookId, startDate, endDate ]	UUID long long	Book id number Date of taking book Date of return book
id	UUID	Card id

## Create Card

### Endpoint

POST /api/cards

### Descriptions

Creates card from a given payload and returns created card model

### Payload

Name	Type	Comment
userId	UUID	User id

### Response

Response code (HTTP status)	Comment
201	Card has been created. Returns card model
400	Card has not been created because of wrong input
404	User with given ID not found

## Update Card (taking book)

### Endpoint

PUT /api/cards/<cardId>

### Description

Adding startDate for book with given Book ID in card with given Card ID and returns updated card model

### Payload

Name	Type	Comment
bookId	UUID	Book id
startDate	Long	Date of taking book

### Response

Response code (HTTP status)	Comment
200	Book has been added. Returns card model
400	Book has not been added because of wrong input

404	Book with given ID not found
409	Returns JSON structure : <ul style="list-style-type: none"> <li>– {"errorMessage": "Oops, book was already taken"}</li> </ul>

## Update Card (return book)

### Endpoint

PUT /api/cards/<cardId>

### Description

Adding endDate for book with given Book ID in card with given Card ID and returns updated card model

### Payload

Name	Type	Comment
bookId	UUID	Book id
endDate	Long	Date of return book

### Response

Response code (HTTP status)	Comment
200	endDate for book has been added. Returns card model
400	endDate for book has not been added because of wrong input
404	Book with given ID not found
409	Returns one JSON structure of the following: <ul style="list-style-type: none"> <li>– {"errorMessage": "Oops, the book has already been returned "}</li> <li>– {"errorMessage": "Oops, the book was not taken "}</li> </ul>

## Get Card

### Endpoint

GET /api/cards/<id>

### Description

Gets card with given ID

### Response

Response code (HTTP status)	Comment
200	Card has been returned (with info about User and Book)
404	Card with given ID not found

## Get Card Collection

### Endpoint

GET /api/cards?bookId=<bookId>&maxResults=<maxResult>&offset=<offset>&orderBy=<orderBy>&order=<order>

### Description

Gets card collection for given Book ID, supports pagination

### Query Parameters

Name	Type	Comment
bookId	UUID	Book ID
maxResults	Integer	Max number of cards in collection. Default value is 10
offset	Integer	Offset for the first card in collection. Default value is 0
orderBy	String	Which field to use for sorting
order	String	Possible values: 'asc' or 'desc'. Ascending or descending

### Response

Response code (HTTP status)	Comment
200	Returns JSON structure containing 2 fields: <ul style="list-style-type: none"><li>- total (Integer) – total number of cards for given Book ID</li><li>- values (list of cards) – list of requested and sorted cards according to pagination parameters</li></ul>

## Get User Card

### Endpoint

GET /api/cards?userId=<userId>

### Description

Gets card for given User ID

### Response

Response code (HTTP status)	Comment
200	Card has been returned (with info about User and Book)
404	Card with given ID not found