

Api Documentation

Overview

Api Documentation

Version information

Version : 1.0

License information

License : Apache 2.0

License URL : <http://www.apache.org/licenses/LICENSE-2.0>

Terms of service : urn:tos

URI scheme

Host : esitartulibrary.herokuapp.com

BasePath : /

Tags

- authentication-controller : Authentication Controller
- book-controller : Book Controller
- customer-controller : Customer Controller
- review-controller : Review Controller
- user-controller : User Controller

Paths

Inserts new book

POST /books

Parameters

Type	Name	Description	Schema
Body	createBookRequest <i>required</i>	createBookRequest	CreateBookRequest

Responses

HTTP Code	Description	Schema
200	OK	Book
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- */*

Tags

- book-controller

Retrieves all books

GET /books

Responses

HTTP Code	Description	Schema
200	OK	< Book > array
401	Unauthorized	No Content

HTTP Code	Description	Schema
403	Forbidden	No Content
404	Not Found	No Content

Produces

- */*

Tags

- book-controller

Updates the book

PUT /books

Parameters

Type	Name	Description	Schema
Body	updateBookRequest <i>required</i>	updateBookRequest	UpdateBookRequest

Responses

HTTP Code	Description	Schema
200	OK	Book
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- **/**

Tags

- book-controller

Retrieves books by author

```
GET /books/authors/{author}
```

Parameters

Type	Name	Description	Schema
Path	author <i>required</i>	author	string

Responses

HTTP Code	Description	Schema
200	OK	< Book > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- **/**

Tags

- book-controller

Retrieves available books

```
GET /books/available
```

Responses

HTTP Code	Description	Schema
200	OK	< Book > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- */*

Tags

- book-controller

Retrieves books by language

```
GET /books/languages/{lang}
```

Parameters

Type	Name	Description	Schema
Path	lang <i>required</i>	lang	string

Responses

HTTP Code	Description	Schema
200	OK	< Book > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- **/**

Tags

- book-controller

Retrieves books by status

```
GET /books/statuses/{status}
```

Parameters

Type	Name	Description	Schema
Path	status <i>required</i>	status	enum (AVAILABLE, BOOKED, RENTED)

Responses

HTTP Code	Description	Schema
200	OK	< Book > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- **/**

Tags

- book-controller

Retrieves books by title

```
GET /books/titles/{title}
```

Parameters

Type	Name	Description	Schema
Path	title <i>required</i>	title	string

Responses

HTTP Code	Description	Schema
200	OK	< Book > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- */*

Tags

- book-controller

Retrieves books by year

```
GET /books/years/{year}
```

Parameters

Type	Name	Description	Schema
Path	year <i>required</i>	year	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK	< Book > array

HTTP Code	Description	Schema
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- */*

Tags

- book-controller

Retrieves the book by its id

```
GET /books/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	Book
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- */*

Tags

- book-controller

Delete the book by its id

DELETE /books/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	No Content
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- */*

Tags

- book-controller

Inserts new customer

POST /customers

Parameters

Type	Name	Description	Schema
Body	customerDto <i>required</i>	customerDto	CustomerDto

Responses

HTTP Code	Description	Schema
200	OK	CustomerDto
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- [application/json](#)

Produces

- [*/*](#)

Tags

- customer-controller

Retrieves all customers

GET /customers

Responses

HTTP Code	Description	Schema
200	OK	< Customer > array
401	Unauthorized	No Content

HTTP Code	Description	Schema
403	Forbidden	No Content
404	Not Found	No Content

Produces

- */*

Tags

- customer-controller

Updates the customer by its id

PUT /customers

Parameters

Type	Name	Description	Schema
Body	customerDto <i>required</i>	customerDto	CustomerDto

Responses

HTTP Code	Description	Schema
200	OK	CustomerDto
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- **/**

Tags

- customer-controller

Retrieves the customer by its id

```
GET /customers/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	Customer
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- **/**

Tags

- customer-controller

Delete the customer by its id

```
DELETE /customers/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	No Content
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- **/**

Tags

- customer-controller

Creates a new review for a book

POST /reviews

Parameters

Type	Name	Description	Schema
Body	review <i>required</i>	review	Review

Responses

HTTP Code	Description	Schema
200	OK	Review

HTTP Code	Description	Schema
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- review-controller

Updates review

PUT /reviews

Parameters

Type	Name	Description	Schema
Body	review <i>required</i>	review	Review

Responses

HTTP Code	Description	Schema
200	OK	Review
201	Created	No Content
401	Unauthorized	No Content

HTTP Code	Description	Schema
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- review-controller

Delete the review by its id

```
DELETE /reviews/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	No Content
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- */*

Tags

- review-controller

Inserts new user

POST /users

Parameters

Type	Name	Description	Schema
Body	user <i>required</i>	user	User

Responses

HTTP Code	Description	Schema
200	OK	User
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- */*

Tags

- user-controller

Retrieves all users

GET /users

Responses

HTTP Code	Description	Schema
200	OK	< User > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

• */*

Tags

• user-controller

Updates the user by its id

PUT /users

Parameters

Type	Name	Description	Schema
Body	user <i>required</i>	user	User

Responses

HTTP Code	Description	Schema
200	OK	User
201	Created	No Content

HTTP Code	Description	Schema
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- user-controller

authorize

POST /users/authenticate

Parameters

Type	Name	Description	Schema
Body	loginDto <i>required</i>	loginDto	LoginDTO

Responses

HTTP Code	Description	Schema
200	OK	JWTToken
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

HTTP Code	Description	Schema
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- authentication-controller

getActualUser

GET /users/current

Responses

HTTP Code	Description	Schema
200	OK	User
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`

Tags

- user-controller

Retrieves the user by its id

GET /users/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	User
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- **/**

Tags

- user-controller

Delete the user by its id

```
DELETE /users/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	No Content

HTTP Code	Description	Schema
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- */*

Tags

- user-controller

Definitions

Authority

Name	Schema
id <i>optional</i>	integer (int64)
role <i>optional</i>	enum (CUSTOMER, LIBRARIAN)

Book

Name	Description	Schema
author <i>optional</i>		string

Name	Description	Schema
categories optional		< enum (ACTION_AND_ADV ENTURE, ALTERNATE_HISTO RY, ANTHOLOGY, ART_ARCHITECTUR E, AUTOBIOGRAPHY, BIOGRAPHY, BUSINESS_ECONOM ICS, CHICK_LIT, CHILDREN_S, CLASSIC, COMIC_BOOK, COMING_OF_AGE, COOKBOOK, CRAFTS_HOBBIES, CRIME, DIARY, DICTIONARY, DRAMA, ENCYCLOPEDIA, FAIRYTALE, FANTASY, GRAPHIC_NOVEL, GUIDE, HEALTH_FITNESS, HISTORICAL_FICTIO N, HISTORY, HOME_AND_GARDE N, HORROR, HUMOR, JOURNAL, MATH, MEMOIR, MYSTERY, PARANORMAL_ROM ANCE, PHILOSOPHY, PICTURE_BOOK, POETRY, POLITICAL_THRILL ER, PRAYER, RELIGION_SPIRITUA LITY_AND_NEW_AG E, REVIEW, ROMANCE, SATIRE, SCIENCE, SCIENCE_FICTION, SELF_HELP, SHORT_STORY, SPORTS_AND_LEISU RE, SUSPENSE,

Name	Description	Schema
id <i>optional</i>		integer (int64)
language <i>optional</i>		string
releaseDate <i>optional</i>	Example : "yyyy"	string
status <i>optional</i>		enum (AVAILABLE, BOOKED, RENTED)
title <i>required</i>		string

BookRentingRequest

Name	Schema
book <i>optional</i>	Book
customer <i>optional</i>	Customer
fineAmount <i>optional</i>	number
id <i>optional</i>	integer (int64)
rentedAt <i>optional</i>	string (date-time)
rentedUntil <i>optional</i>	string (date-time)
status <i>optional</i>	enum (CANCELLED, DECLINED, EXPIRED, OPEN, RENTED)

CreateBookRequest

Name	Description	Schema
author <i>required</i>	Length : 2 - 2147483647	string

Name	Description	Schema
categories optional		< enum (ACTION_AND_ADV ENTURE, ALTERNATE_HISTO RY, ANTHOLOGY, ART_ARCHITECTUR E, AUTOBIOGRAPHY, BIOGRAPHY, BUSINESS_ECONOM ICS, CHICK_LIT, CHILDREN_S, CLASSIC, COMIC_BOOK, COMING_OF_AGE, COOKBOOK, CRAFTS_HOBBIES, CRIME, DIARY, DICTIONARY, DRAMA, ENCYCLOPEDIA, FAIRYTALE, FANTASY, GRAPHIC_NOVEL, GUIDE, HEALTH_FITNESS, HISTORICAL_FICTIO N, HISTORY, HOME_AND_GARDE N, HORROR, HUMOR, JOURNAL, MATH, MEMOIR, MYSTERY, PARANORMAL_ROM ANCE, PHILOSOPHY, PICTURE_BOOK, POETRY, POLITICAL_THRILL ER, PRAYER, RELIGION_SPIRITUA LITY_AND_NEW_AG E, REVIEW, ROMANCE, SATIRE, SCIENCE, SCIENCE_FICTION, SELF_HELP, SHORT_STORY, SPORTS_AND_LEISU RE, SUSPENSE,

Name	Description	Schema
language <i>required</i>		string
title <i>required</i>	Length : 2 - 2147483647	string
year <i>required</i>		integer (int32)

Customer

Name	Schema
bookRentingRequests <i>optional</i>	< BookRentingRequest > array
fineAmount <i>optional</i>	number
id <i>optional</i>	integer (int64)
idCode <i>required</i>	string
user <i>optional</i>	User

CustomerDto

Name	Description	Schema
activated <i>optional</i>		boolean
address <i>optional</i>		string
email <i>optional</i>		string

Name	Description	Schema
fineAmount <i>optional</i>		number
firstName <i>required</i>	Length : 2 - 50	string
idCode <i>optional</i>		string
lastName <i>required</i>	Length : 2 - 50	string
passWord <i>required</i>	Length : 4 - 50	string
phoneNumber <i>optional</i>		string
roles <i>optional</i>		< enum (CUSTOMER, LIBRARIAN) > array
userId <i>optional</i>		integer (int64)
userName <i>required</i>	Length : 4 - 50	string

JWTToken

Name	Schema
token <i>optional</i>	string

LoginDTO

Name	Description	Schema
password <i>required</i>	Length : 4 - 100	string

Name	Description	Schema
rememberMe <i>optional</i>		boolean
username <i>required</i>	Length : 1 - 50	string

Review

Name	Schema
book <i>optional</i>	Book
comment <i>optional</i>	string
id <i>optional</i>	integer (int64)
score <i>optional</i>	integer (int32)

UpdateBookRequest

Name	Description	Schema
author <i>required</i>	Length : 2 - 2147483647	string

Name	Description	Schema
categories optional		< enum (ACTION_AND_ADV ENTURE, ALTERNATE_HISTO RY, ANTHOLOGY, ART_ARCHITECTUR E, AUTOBIOGRAPHY, BIOGRAPHY, BUSINESS_ECONOM ICS, CHICK_LIT, CHILDREN_S, CLASSIC, COMIC_BOOK, COMING_OF_AGE, COOKBOOK, CRAFTS_HOBBIES, CRIME, DIARY, DICTIONARY, DRAMA, ENCYCLOPEDIA, FAIRYTALE, FANTASY, GRAPHIC_NOVEL, GUIDE, HEALTH_FITNESS, HISTORICAL_FICTIO N, HISTORY, HOME_AND_GARDE N, HORROR, HUMOR, JOURNAL, MATH, MEMOIR, MYSTERY, PARANORMAL_ROM ANCE, PHILOSOPHY, PICTURE_BOOK, POETRY, POLITICAL_THRILL ER, PRAYER, RELIGION_SPIRITUA LITY_AND_NEW_AG E, REVIEW, ROMANCE, SATIRE, SCIENCE, SCIENCE_FICTION, SELF_HELP, SHORT_STORY, SPORTS_AND_LEISU RE, SUSPENSE,

Name	Description	Schema
id <i>required</i>		integer (int64)
language <i>required</i>		string
title <i>required</i>	Length : 2 - 2147483647	string
year <i>required</i>		integer (int32)

User

Name	Description	Schema
activated <i>optional</i>		boolean
address <i>optional</i>		string
authorities <i>optional</i>		< Authority > array
email <i>optional</i>		string
firstName <i>required</i>	Length : 2 - 50	string
id <i>optional</i>		integer (int64)
lastName <i>required</i>	Length : 2 - 50	string
password <i>required</i>	Length : 4 - 100	string
phoneNumber <i>optional</i>		string

Name	Description	Schema
type <i>optional</i>		enum (CUSTOMER, LIBRARIAN)
username <i>required</i>	Length : 4 - 50	string