# Bookbuddy API

Bookbuddy API Information

Version: 1.0

All rights reserved

http://apache.org/licenses/LICENSE-2.0.html

## Access

## Methods

[ Jump to [Models](#__Models) ]

### Table of Contents

#### [BookCollectionController](#BookCollectionController)

* [put /collections/{userId}/{collectionName}/add\_book/{bookId}](#addBook)
* [post /collections/{userId}/new/{name}](#addCollection)
* [delete /collections/{collectionId}](#deleteCollection)
* [get /collections/get/{userId}](#getAllCollectionsByUser)
* [get /collections/{collectionId}](#getCollection)
* [put /collections/{collectionId}/remove\_book/{bookId}](#removeBook)
* [put /collections/{collectionId}/rename/{newName}](#renameCollection)

#### [BookController](#BookController)

* [get /books/{bookId}](#findBook)
* [get /books/all](#getAll)
* [get /books/popular](#getFirst20)
* [get /books/search/{searchTerm}](#searchBook)

#### [CartController](#CartController)

* [post /cart/add-item/{cartId}/{bookId}](#addItemToCart)
* [get /cart/get/{userId}](#getCart)
* [put /cart/remove-item/{cartId}/{cartItemId}](#removeItemFromCart)

#### [GuestCheckoutController](#GuestCheckoutController)

* [post /guestcheckout](#saveGuestCheckout)

#### [UserCheckoutController](#UserCheckoutController)

* [post /user-checkout](#createUserCheckout)

#### [UserController](#UserController)

* [get /users/all](#all)
* [delete /users/{userId}](#deleteUser)
* [get /users/{userId}](#findUser)
* [get /users/email](#findUserByEmail)
* [post /users/new](#newUser)
* [put /users/{userId}](#updateUser)

# BookCollectionController

[Up](" \l "__Methods)

put /collections/{userId}/{collectionName}/add\_book/{bookId}

Add a book to a collection (addBook)

### Path parameters

**userId (required)**

Path Parameter — Unique ID corresponding to a user format: int64

**collectionName (required)**

Path Parameter — Name of the collection

**bookId (required)**

Path Parameter — Unique ID corresponding to a book format: int64

### Return type

[BookCollectionDTO](#BookCollectionDTO)

### Example data

Content-Type: application/json

{

"booksInCollection" : [ {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}, {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

} ],

"id" : 1,

"collectionName" : "Favorite Books"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [BookCollectionDTO](#BookCollectionDTO)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

post /collections/{userId}/new/{name}

Add a new collection to the database (addCollection)

### Path parameters

**userId (required)**

Path Parameter — Unique ID corresponding to a user format: int64

**name (required)**

Path Parameter — What the user wants to name the new collection

### Return type

[BookCollectionDTO](#BookCollectionDTO)

### Example data

Content-Type: application/json

{

"booksInCollection" : [ {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}, {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

} ],

"id" : 1,

"collectionName" : "Favorite Books"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 201

Book Collection added successfully [BookCollectionDTO](#BookCollectionDTO)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

delete /collections/{collectionId}

Delete collection by id (deleteCollection)

### Path parameters

**collectionId (required)**

Path Parameter — Unique ID corresponding to a collection format: int64

### Return type

String

### Example data

Content-Type: application/json

""

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [String](#String)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

get /collections/get/{userId}

Get all collections from a user (getAllCollectionsByUser)

### Path parameters

**userId (required)**

Path Parameter — Unique ID corresponding to a user format: int64

### Return type

array[[BookCollectionDTO](#BookCollectionDTO)]

### Example data

Content-Type: application/json

[ {

"booksInCollection" : [ {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}, {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

} ],

"id" : 1,

"collectionName" : "Favorite Books"

}, {

"booksInCollection" : [ {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}, {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

} ],

"id" : 1,

"collectionName" : "Favorite Books"

} ]

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

get /collections/{collectionId}

Get collection by id (getCollection)

### Path parameters

**collectionId (required)**

Path Parameter — Unique ID corresponding to a collection format: int64

### Return type

[BookCollectionDTO](#BookCollectionDTO)

### Example data

Content-Type: application/json

{

"booksInCollection" : [ {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}, {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

} ],

"id" : 1,

"collectionName" : "Favorite Books"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [BookCollectionDTO](#BookCollectionDTO)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

put /collections/{collectionId}/remove\_book/{bookId}

Remove a book from a collection (removeBook)

### Path parameters

**collectionId (required)**

Path Parameter — Unique ID corresponding to a collection format: int64

**bookId (required)**

Path Parameter — Unique ID corresponding to a book format: int64

### Return type

[BookCollectionDTO](#BookCollectionDTO)

### Example data

Content-Type: application/json

{

"booksInCollection" : [ {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}, {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

} ],

"id" : 1,

"collectionName" : "Favorite Books"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [BookCollectionDTO](#BookCollectionDTO)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

put /collections/{collectionId}/rename/{newName}

Rename a collection (renameCollection)

### Path parameters

**collectionId (required)**

Path Parameter — Unique ID corresponding to a collection format: int64

**newName (required)**

Path Parameter — What the user wants to rename to collection to

### Return type

[BookCollectionDTO](#BookCollectionDTO)

### Example data

Content-Type: application/json

{

"booksInCollection" : [ {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}, {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

} ],

"id" : 1,

"collectionName" : "Favorite Books"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [BookCollectionDTO](#BookCollectionDTO)

#### 404

Not Found [String](#String)

# BookController

[Up](" \l "__Methods)

get /books/{bookId}

Get book by id (findBook)

### Path parameters

**bookId (required)**

Path Parameter — Unique ID corresponding to a book format: int64

### Return type

[Book](#Book)

### Example data

Content-Type: application/json

{

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Book](#Book)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

get /books/all

Get all books (getAll)

### Return type

array[[Book](#Book)]

### Example data

Content-Type: application/json

[ {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}, {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

} ]

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

get /books/popular

Get most popular books (getFirst20)

### Return type

array[[Book](#Book)]

### Example data

Content-Type: application/json

[ {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}, {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

} ]

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

get /books/search/{searchTerm}

Search for book by title (searchBook)

### Path parameters

**searchTerm (required)**

Path Parameter — Title of book to search for

### Return type

[Book](#Book)

### Example data

Content-Type: application/json

{

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Book](#Book)

#### 404

Not Found [String](#String)

# CartController

[Up](" \l "__Methods)

post /cart/add-item/{cartId}/{bookId}

Add an item to a cart (addItemToCart)

### Path parameters

**cartId (required)**

Path Parameter — Unique ID corresponding to a Cart format: int64

**bookId (required)**

Path Parameter — Unique ID corresponding to a Book format: int64

### Return type

[CartDTO](#CartDTO)

### Example data

Content-Type: application/json

{

"totalPrice" : 100,

"cartId" : 1,

"userId" : 1,

"items" : [ {

"userCheckout" : {

"cardCVV" : "cardCVV",

"address" : "address",

"cardExpiry" : "cardExpiry",

"totalPrice" : 1.4658129805029452,

"id" : 6,

"cartItems" : [ null, null ],

"user" : {

"firstName" : "John",

"lastName" : "Doe",

"firebaseUID" : 123456789,

"userDescription" : "I love reading!",

"genres" : [ "Fantasy", "Science Fiction" ],

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

},

"cardNumber" : "cardNumber"

},

"book" : {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

},

"itemPrice" : 0.8008281904610115,

"cartItemId" : 1

}, {

"userCheckout" : {

"cardCVV" : "cardCVV",

"address" : "address",

"cardExpiry" : "cardExpiry",

"totalPrice" : 1.4658129805029452,

"id" : 6,

"cartItems" : [ null, null ],

"user" : {

"firstName" : "John",

"lastName" : "Doe",

"firebaseUID" : 123456789,

"userDescription" : "I love reading!",

"genres" : [ "Fantasy", "Science Fiction" ],

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

},

"cardNumber" : "cardNumber"

},

"book" : {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

},

"itemPrice" : 0.8008281904610115,

"cartItemId" : 1

} ]

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [CartDTO](#CartDTO)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

get /cart/get/{userId}

Get a cart by ID (getCart)

### Path parameters

**userId (required)**

Path Parameter — Unique ID corresponding to a User format: int64

### Return type

[CartDTO](#CartDTO)

### Example data

Content-Type: application/json

{

"totalPrice" : 100,

"cartId" : 1,

"userId" : 1,

"items" : [ {

"userCheckout" : {

"cardCVV" : "cardCVV",

"address" : "address",

"cardExpiry" : "cardExpiry",

"totalPrice" : 1.4658129805029452,

"id" : 6,

"cartItems" : [ null, null ],

"user" : {

"firstName" : "John",

"lastName" : "Doe",

"firebaseUID" : 123456789,

"userDescription" : "I love reading!",

"genres" : [ "Fantasy", "Science Fiction" ],

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

},

"cardNumber" : "cardNumber"

},

"book" : {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

},

"itemPrice" : 0.8008281904610115,

"cartItemId" : 1

}, {

"userCheckout" : {

"cardCVV" : "cardCVV",

"address" : "address",

"cardExpiry" : "cardExpiry",

"totalPrice" : 1.4658129805029452,

"id" : 6,

"cartItems" : [ null, null ],

"user" : {

"firstName" : "John",

"lastName" : "Doe",

"firebaseUID" : 123456789,

"userDescription" : "I love reading!",

"genres" : [ "Fantasy", "Science Fiction" ],

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

},

"cardNumber" : "cardNumber"

},

"book" : {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

},

"itemPrice" : 0.8008281904610115,

"cartItemId" : 1

} ]

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [CartDTO](#CartDTO)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

put /cart/remove-item/{cartId}/{cartItemId}

Remove an item from a cart (removeItemFromCart)

### Path parameters

**cartId (required)**

Path Parameter — Unique ID corresponding to a Cart format: int64

**cartItemId (required)**

Path Parameter — Unique ID corresponding to a CartItem format: int64

### Return type

[CartDTO](#CartDTO)

### Example data

Content-Type: application/json

{

"totalPrice" : 100,

"cartId" : 1,

"userId" : 1,

"items" : [ {

"userCheckout" : {

"cardCVV" : "cardCVV",

"address" : "address",

"cardExpiry" : "cardExpiry",

"totalPrice" : 1.4658129805029452,

"id" : 6,

"cartItems" : [ null, null ],

"user" : {

"firstName" : "John",

"lastName" : "Doe",

"firebaseUID" : 123456789,

"userDescription" : "I love reading!",

"genres" : [ "Fantasy", "Science Fiction" ],

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

},

"cardNumber" : "cardNumber"

},

"book" : {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

},

"itemPrice" : 0.8008281904610115,

"cartItemId" : 1

}, {

"userCheckout" : {

"cardCVV" : "cardCVV",

"address" : "address",

"cardExpiry" : "cardExpiry",

"totalPrice" : 1.4658129805029452,

"id" : 6,

"cartItems" : [ null, null ],

"user" : {

"firstName" : "John",

"lastName" : "Doe",

"firebaseUID" : 123456789,

"userDescription" : "I love reading!",

"genres" : [ "Fantasy", "Science Fiction" ],

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

},

"cardNumber" : "cardNumber"

},

"book" : {

"author" : "F. Scott Fitzgerald",

"price" : 12.99,

"description" : "A novel about the American Dream",

"id" : 1,

"title" : "The Great Gatsby"

},

"itemPrice" : 0.8008281904610115,

"cartItemId" : 1

} ]

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [CartDTO](#CartDTO)

#### 404

Not Found [String](#String)

# GuestCheckoutController

[Up](" \l "__Methods)

post /guestcheckout

(saveGuestCheckout)

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

**body** [**GuestCheckout**](#GuestCheckout) **(required)**

Body Parameter —

### Return type

[GuestCheckout](#GuestCheckout)

### Example data

Content-Type: application/json

{

"cardCVV" : "cardCVV",

"address" : "address",

"cardExpiry" : "cardExpiry",

"price" : "price",

"name" : "name",

"id" : 0,

"bookName" : "bookName",

"email" : "email",

"cardNumber" : "cardNumber"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [GuestCheckout](#GuestCheckout)

#### 404

Not Found [String](#String)

# UserCheckoutController

[Up](" \l "__Methods)

post /user-checkout

(createUserCheckout)

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

**body** [**UserCheckout**](#UserCheckout) **(required)**

Body Parameter —

### Return type

[UserCheckout](#UserCheckout)

### Example data

Content-Type: application/json

{

"cardCVV" : "cardCVV",

"address" : "address",

"cardExpiry" : "cardExpiry",

"totalPrice" : 1.4658129805029452,

"id" : 6,

"cartItems" : [ null, null ],

"user" : {

"firstName" : "John",

"lastName" : "Doe",

"firebaseUID" : 123456789,

"userDescription" : "I love reading!",

"genres" : [ "Fantasy", "Science Fiction" ],

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

},

"cardNumber" : "cardNumber"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [UserCheckout](#UserCheckout)

#### 404

Not Found [String](#String)

# UserController

[Up](" \l "__Methods)

get /users/all

Admin route to list all users (all)

### Return type

array[[UserDTO](#UserDTO)]

### Example data

Content-Type: application/json

[ {

"firstName" : "John",

"lastName" : "Doe",

"userDescription" : "I love reading!",

"genres" : [ {

"name" : "name",

"id" : 0

}, {

"name" : "name",

"id" : 0

} ],

"cartId" : 1,

"description" : "description",

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

}, {

"firstName" : "John",

"lastName" : "Doe",

"userDescription" : "I love reading!",

"genres" : [ {

"name" : "name",

"id" : 0

}, {

"name" : "name",

"id" : 0

} ],

"cartId" : 1,

"description" : "description",

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

} ]

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

delete /users/{userId}

Delete user by id (deleteUser)

### Path parameters

**userId (required)**

Path Parameter — Unique ID corresponding to a user format: int64

### Return type

String

### Example data

Content-Type: application/json

""

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [String](#String)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

get /users/{userId}

Get user details by id (findUser)

### Path parameters

**userId (required)**

Path Parameter — Unique ID corresponding to a user format: int64

### Return type

[UserDTO](#UserDTO)

### Example data

Content-Type: application/json

{

"firstName" : "John",

"lastName" : "Doe",

"userDescription" : "I love reading!",

"genres" : [ {

"name" : "name",

"id" : 0

}, {

"name" : "name",

"id" : 0

} ],

"cartId" : 1,

"description" : "description",

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [UserDTO](#UserDTO)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

get /users/email

Get user details by email (findUserByEmail)

### Query parameters

**email (required)**

Query Parameter —

### Return type

[UserDTO](#UserDTO)

### Example data

Content-Type: application/json

{

"firstName" : "John",

"lastName" : "Doe",

"userDescription" : "I love reading!",

"genres" : [ {

"name" : "name",

"id" : 0

}, {

"name" : "name",

"id" : 0

} ],

"cartId" : 1,

"description" : "description",

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [UserDTO](#UserDTO)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

post /users/new

Add a new user to the database (newUser)

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

**body** [**CreateUserDTO**](#CreateUserDTO) **(required)**

Body Parameter —

### Return type

[UserDTO](#UserDTO)

### Example data

Content-Type: application/json

{

"firstName" : "John",

"lastName" : "Doe",

"userDescription" : "I love reading!",

"genres" : [ {

"name" : "name",

"id" : 0

}, {

"name" : "name",

"id" : 0

} ],

"cartId" : 1,

"description" : "description",

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 201

User added successfully [UserDTO](#UserDTO)

#### 404

Not Found [String](#String)

[Up](" \l "__Methods)

put /users/{userId}

Update user details by id (updateUser)

### Path parameters

**userId (required)**

Path Parameter — Unique ID corresponding to a user format: int64

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

**body** [**UserDTO**](#UserDTO) **(required)**

Body Parameter —

### Return type

[UserDTO](#UserDTO)

### Example data

Content-Type: application/json

{

"firstName" : "John",

"lastName" : "Doe",

"userDescription" : "I love reading!",

"genres" : [ {

"name" : "name",

"id" : 0

}, {

"name" : "name",

"id" : 0

} ],

"cartId" : 1,

"description" : "description",

"dateOfBirth" : "2000-01-23",

"id" : 1,

"email" : "test@email.com"

}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [UserDTO](#UserDTO)

#### 404

Not Found [String](#String)

## Models

[ Jump to [Methods](#__Methods) ]

### Table of Contents

1. [Book](#Book)
2. [BookCollectionDTO](#BookCollectionDTO)
3. [CartDTO](#CartDTO)
4. [CartItem](#CartItem)
5. [CreateUserDTO](#CreateUserDTO)
6. [Genre](#Genre)
7. [GenreDTO](#GenreDTO)
8. [GuestCheckout](#GuestCheckout)
9. [User](#User)
10. [UserCheckout](#UserCheckout)
11. [UserDTO](#UserDTO)

### Book [Up](#__Models)

A book entity

**id (optional)**

[*Long*](#long) Unique ID generated by database corresponding to a book format: int64

example: 1

**title (optional)**

[*String*](#string) Title of the book

example: The Great Gatsby

**author (optional)**

[*String*](#string) Author of the book

example: F. Scott Fitzgerald

**price (optional)**

[*BigDecimal*](#BigDecimal) Price of the book

example: 12.99

**description (optional)**

[*String*](#string) Description of the book

example: A novel about the American Dream

### BookCollectionDTO [Up](#__Models)

A DTO for a book collection

**id (optional)**

[*Long*](#long) Unique ID generated by database corresponding to a book collection format: int64

example: 1

**collectionName (optional)**

[*String*](#string) Name of the collection

example: Favorite Books

**booksInCollection (optional)**

[*array[Book]*](#Book) List of books in the collection

### CartDTO [Up](#__Models)

Data Transfer Object for Cart

**cartId (optional)**

[*Long*](#long) Unique ID corresponding to a Cart format: int64

example: 1

**userId (optional)**

[*Long*](#long) Unique ID corresponding to a User format: int64

example: 1

**items (optional)**

[*array[CartItem]*](#CartItem) List of items in the cart

**totalPrice (optional)**

[*BigDecimal*](#BigDecimal) Total price of all items in the cart

example: 100

### CartItem [Up](#__Models)

An item present in the cart

**cartItemId (optional)**

[*Long*](#long) Unique ID generated by database corresponding to a user format: int64

example: 1

**book (optional)**

[*Book*](#Book)

**itemPrice (optional)**

[*BigDecimal*](#BigDecimal) price of the item

**userCheckout (optional)**

[*UserCheckout*](#UserCheckout)

### CreateUserDTO [Up](#__Models)

A DTO for creating a user

**firstName (optional)**

[*String*](#string) First name of the user

example: John

**lastName (optional)**

[*String*](#string) Last name of the user

example: Doe

**email (optional)**

[*String*](#string) Email of the user

example: test@email.com

**dateOfBirth (optional)**

[*date*](#date) Date of birth of the user format: date

**description (optional)**

[*String*](#string) Description of the user

example: I love reading!

**genres (optional)**

[*String*](#string)

**genresList (optional)**

[*array[String]*](#string)

### Genre [Up](#__Models)

List of favorite genres of the user

**id (optional)**

[*Long*](#long) format: int64

**name (optional)**

[*String*](#string)

### GenreDTO [Up](#__Models)

List of genres that the user is interested in

**id (optional)**

[*Long*](#long) format: int64

**name (optional)**

[*String*](#string)

### GuestCheckout [Up](#__Models)

**id (optional)**

[*Long*](#long) format: int64

**name (optional)**

[*String*](#string)

**email (optional)**

[*String*](#string)

**address (optional)**

[*String*](#string)

**cardNumber (optional)**

[*String*](#string)

**cardExpiry (optional)**

[*String*](#string)

**cardCVV (optional)**

[*String*](#string)

**price (optional)**

[*String*](#string)

**bookName (optional)**

[*String*](#string)

### User [Up](#__Models)

A user of the application

**id (optional)**

[*Long*](#long) Unique ID generated by database corresponding to a user format: int64

example: 1

**firebaseUID (optional)**

[*Long*](#long) Unique ID generated by Firebase corresponding to a user format: int64

example: 123456789

**firstName (optional)**

[*String*](#string) First name of the user

example: John

**lastName (optional)**

[*String*](#string) Last name of the user

example: Doe

**email (optional)**

[*String*](#string) Email of the user

example: test@email.com

**dateOfBirth (optional)**

[*date*](#date) Date of birth of the user format: date

**userDescription (optional)**

[*String*](#string) Description of the user

example: I love reading!

**genres (optional)**

[*array[Genre]*](#Genre) List of favorite genres of the user

example: ["Fantasy","Science Fiction"]

### UserCheckout [Up](#__Models)

**id (optional)**

[*Long*](#long) format: int64

**user (optional)**

[*User*](#User)

**cartItems (optional)**

[*array[CartItem]*](#CartItem)

**totalPrice (optional)**

[*BigDecimal*](#BigDecimal)

**address (optional)**

[*String*](#string)

**cardNumber (optional)**

[*String*](#string)

**cardExpiry (optional)**

[*String*](#string)

**cardCVV (optional)**

[*String*](#string)

### UserDTO [Up](#__Models)

A DTO for a user

**id (optional)**

[*Long*](#long) Unique ID generated by database corresponding to a user format: int64

example: 1

**firstName (optional)**

[*String*](#string) First name of the user

example: John

**lastName (optional)**

[*String*](#string) Last name of the user

example: Doe

**email (optional)**

[*String*](#string) Email of the user

example: test@email.com

**dateOfBirth (optional)**

[*date*](#date) Date of birth of the user format: date

**cartId (optional)**

[*Long*](#long) Unique ID corresponding to a user's cart format: int64

example: 1

**userDescription (optional)**

[*String*](#string) Description of the user

example: I love reading!

**genres (optional)**

[*array[GenreDTO]*](#GenreDTO) List of genres that the user is interested in

**description (optional)**

[*String*](#string)