

# Portfolio Project

CS 493: Cloud Application Development

Spring 2022

Oregon State University

Kristina Montanez

**The URL where my application is deployed on GCP:**

<https://montanek-portfolio.uw.r.appspot.com/>

**The URL for account creation/login:**

<https://montanek-portfolio.uw.r.appspot.com/>

NOTE: Please place "Bearer " (with a space after) in the "jwt1" and "jwt2" environment variables in the Postman Environment collection. Example JWTs are left in the environment variables to show you the correct format for entering your own JWT in the variables.

## **IMPORTANT NOTE:**

Per Ed Discussion #305, the 405 Status response is captured by Flask and sent as a 500 HTML, and there is no real way around this. Per instructor, if we cannot get around this, it's ok. I have left out Postman tests as there is no way to get them to come back successful. Please see main.py to see each method's setup that would otherwise catch "methods not allowed".

## TABLE OF CONTENTS:

[Change log](#)  
[Data Model](#)  
[List all Users](#)  
[Create a Book](#)  
[List all Books](#)  
[Get a Book](#)  
[Edit a Book \(PATCH\)](#)  
[Edit a Book \(PUT\)](#)  
[Delete a Book](#)  
[Create a Review](#)  
[List all Reviews](#)  
[Get a Review](#)  
[Edit a Review \(PATCH\)](#)  
[Edit a Review \(PUT\)](#)  
[Delete a Review](#)  
[Link a Review to a Book](#)  
[Unlink a Review to a Book](#)  
[Delete all Books and Reviews](#) (For testing purposes only)  
[Delete all Users](#) (For testing purposes only)

## Change log

Version	Change	Date
1.0	Initial version.	June 5, 2022

## Data Model

The app stores three kinds of entities in Datastore: Users, Books, and Reviews.

### Description of the relationship between the non-user entities:

The two non-user entities are Books and Reviews. They are independent of each other, however, a Review may have a Book attached to it, or no Book attached. A Book may have no reviews attached, one review attached, or several reviews attached to it. This allows a User to have multiple reviews for one book, as an opinion of a book may change over time and the User may wish to keep a review of each time they read the Book. The review\_ids are held in an array within the Book entity. The book\_id is a single-value attribute in the Book entity.

### Description of the relationship between the user and non-user entities:

Users will automatically have their account "ids" placed as an attribute in the Book or Review once created, as "owner\_id". Users may create, edit, and delete their Books and Reviews, but they cannot abandon a Book or Review without deleting it. They cannot assign another User to their Book or Review. Users will not keep attributes of either Books or Reviews in their account entities. Only the Books and Reviews will catalog the ownership relationship between User and non-user entities.

### How our Application maps a supplied JWT to a User:

A new User will first visit <https://montanek-portfolio.uw.r.appspot.com/> to see a "Welcome" page. This page will show the following explanation:

This page allows you to test my CS493 Portfolio Project. The "Authorize and Create Account" button below will redirect you to Google's OAuth sign-in page asking for authorization to access your profile information. Only your first and last name will be used in this application. Once you authorize the application to access your name, your own personal account will be created and you may access the account using a Token provided after successful login.

The new User will then click on the button below, which will redirect them to a Google Authentication sign in. The new User will provide their credentials and allow the Application access to their profile information. Upon successful login, the Application will automatically create a new User Account, using the new User's first name, last name, and Unique ID provided by their JWT. This Unique ID becomes a private ID that will never be displayed in the Application. The User will then be redirected to a "User Info" page. Along with the Unique ID and the JWT to copy and paste, the page will show the following explanation:

Your own personal account is created and you may now access the account using a Token provided below. Keep your Unique I.D., this is the I.D. associated with your JWTs and will associate you to your account. Please use the Welcome Page to refresh your JWT.

The new User will use this JWT in all their subsequent posts to the Application, in the "Authorization" header, except for "GET /users" which does not require the header. This JWT will be used to connect the new User to their account via the Unique ID, and retrieve the Books and Reviews that they create.

**NOTE: If the new User runs out of time, the new User may go back to the "Welcome" page, and re-sign in to refresh their JWT. This refreshed JWT will not create a new User Account, and will still link the new User to their existing account Books and Reviews.**

## Users

Property	Data Type	Notes	Required
id	Integer	The Account id of the user. Datastore automatically generates it. This attribute cannot be changed.	automatic
First Name	String	First Name of Account User. This attribute cannot be changed.	automatic
Last Name	String	Last Name of Account User. This attribute cannot be changed.	automatic
<b>PRIVATE ID</b>	<b>Integer</b>	<b>SUB I.D. from JWT provided at account creation. THIS ATTRIBUTE IS STORED BUT NOT VIEWABLE</b>	automatic

## Books

Property	Data Type	Notes	Required
id	Integer	The id of the book. Datastore automatically generates it. This attribute cannot be changed.	automatic
title	String	Title of the book.	yes
author	String	Author of the book.	yes
genre	String	The genre of the book.	yes
owner_id	Integer	The account id of the book owner.	automatic
review_ids	Integer Array	An array of the reviews associated with the book.	yes

## Reviews

Property	Data Type	Notes	Required
id	Integer	The id of the review. Datastore automatically generates it. This attribute cannot be changed.	automatic
date	String	The date that the User last read the book.	yes
rating	Integer	The rating the User gives the book. Rating is 1 through 5, with 1 being the worst, and 5 being the highest.	yes
comment	String	What the User thinks about the book.	yes
owner_id	Integer	The account id of the review owner.	automatic
book_id	Integer	A book associated with the review.	yes

## List all Users

List all the books.

GET /users
------------

### Request

#### Path Parameters

None

#### Protected

No

#### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>· Request Accept header must exist in Request headers.</li><li>· Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for violating the following: <ul style="list-style-type: none"><li>· The Request Method must be "POST".</li></ul>

### Response Examples

- Our App will automatically generate an ID and store it with your user account. However, this is not an attribute that you may alter with POST, PUT, or PATCH. You may not individually delete your account. Please refer to "[Delete all Users](#)" below to reset user accounts administratively.
- Our App will store your Unique I.D., provided to you when you create your account. For security purposes, this Unique I.D. will not be displayed to you or to any other user of this App.
- Your First and Last name are attributes we store from your Google Authorization sign in. These attributes cannot be altered with POST, PUT, or PATCH.

### Success

Status: 200 OK
----------------

```
{
  "Last Name": "Montanez",
  "First Name": "Kristina R",
  "id": 5867703839490048
}, {
  "Last Name": "montanez",
  "id": 6294026856693760,
  "First Name": "kristina"
}]
```

### *Failures*

Status: 406 Missing Accept header.

```
{
  "Error": "The request is missing the Accept Header"
}
```

Status: 406 Unsupported Accept header.

```
{
  "Error": "The request Accept Header must be 'application/json'"
}
```

Status: 405 Invalid Method for Request.

```
{
  "The Request Method is not allowed. Please use 'GET' request method only"
}
```

## Create a Book

Allows you to create a new book.

POST /books
-------------

### Request

#### Path Parameters

None

#### Protected

Yes

### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Description	Required?
title	The title of the book.	Yes
author	The author of the book.	Yes
genre	The author of the book. <i>E.g. "fantasy"</i>	Yes

#### Request Body Example

```
{
  "title": "Dune",
  "author": "Frank Herbert",
  "genre": "sci-fi"
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li></ul>



		<ul style="list-style-type: none"> <li>Invalid JWT Authorization Token.</li> </ul>
Failure	401 Invalid User Account for Request	<p>A 401 status code will be returned for the following:</p> <ul style="list-style-type: none"> <li>Missing user account connected with JWT Authorization Token.</li> </ul>
Failure	405 Invalid Method for Request	<p>A 405 status code will be returned for violating the following:</p> <ul style="list-style-type: none"> <li>The Request Method must either be "POST" or "GET".</li> </ul>
Failure	403 Invalid Request Body	<p>A 403 status code will be returned for violating the following:</p> <ul style="list-style-type: none"> <li>Request body must have the required attributes "title", "author", and "genre".</li> </ul>

### Response Examples

- Our App will automatically generate an ID and store it with the entity being created. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- Our App will provide you with the direct URL address to the book. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- Book attributes must have "title", "author", and "genre". Extraneous attributes provided will be ignored..
- Our App will automatically place your owner ID as an attribute of the book. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- Our App will automatically place an empty list for new Reviews to be associated with the Book. You may connect a Review to this book via the [Link a Review to a Book](#) route below.

### Success

Status: 201 Created

```
{
  "title": "Dune",
  "author": "Frank Herbert",
  "genre": "sci-fi",
  "review_ids": [],
  "owner_id": 5867703839490048,
  "id": 4827211370594304,
  "self": "http://montanek-portfolio.uw.r.appspot.com/books/4827211370594304"
}
```

### Failures

Status: 406 Unsupported Request.  { "Error": "The request is missing the Accept Header" }
Status: 406 Unsupported Request.  { "Error": "The request Accept Header must be 'application/json'" }
Status: 401 Missing Authorization.  { "Error": "The request is missing authorization" }
Status: 401 Invalid Authorization.  { "Error": "The request authorization is invalid" }
Status: 401 Invalid User Account for Request.  { "Error": "No User Account exists for this login" }
Status: 405 Invalid Method for Request.  { "The Request Method is not allowed. Please use 'POST' or 'GET' request methods only" }
Status: 403 Invalid Request Body.  { "Error": "The request object must use the required attributes" }
Status: 403 Invalid Request Body.  { "Error": "The request object is missing at least one of the required attributes" }

## List all Books

List all the books.

GET /books
------------

### Request

Path Parameters

None

Protected

Yes

Request Body

None

### Response

Response Body Format

JSON

### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li><li>Invalid JWT Authorization Token.</li></ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for violating the following: <ul style="list-style-type: none"><li>The Request Method must either be "POST" or "GET".</li></ul>

### Response Examples

- Our App will allow you to view books that you own. You may not view books owned by other users.

- Our App will provide you with the direct URL address to each book. However, this is not an attribute that you may alter with POST, PUT, or PATCH.

#### Success

Status: 200 OK

```
{
  "total items": 6,
  "reviews": {
    "reviews": [{
      "owner_id": 5867703839490048,
      "book_id": null,
      "rating": 5,
      "date": "3/11/2005",
      "comment": "Iconic 'enemies to lovers' trope where an entitled @$$hat is repeatedly offended by a stick-in-the-mud. Sorry if i broke hearts with that one.",
      "id": 4750879735414784,
      "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/4750879735414784"
    }, {
      "owner_id": 5867703839490048,
      "rating": 1,
      "date": "1/05/2011",
      "book_id": null,
      "comment": "Don't judge, I had a really boring job after law school and needed audio that would keep me awake..",
      "id": 5084899610460160,
      "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/5084899610460160"
    }, {
      "date": "7/4/2000",
      "owner_id": 5867703839490048,
      "comment": "Young, wealthy extraterrestrial immigrant starts a sex cult and is eventually killed by a mob.",
      "book_id": null,
      "rating": 2,
      "id": 5367335988756480,
      "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/5367335988756480"
    }, {
      "comment": "Bro-mance 'enemies to lovers' are tasked with babysitting a kid so the world doesn't end. I will forever ship David Tennant and Michael Sheen.",
      "owner_id": 5867703839490048,
      "rating": 5,
      "date": "2/25/2019",
      "book_id": null,
      "id": 5420191735349248,
      "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/5420191735349248"
    }, {
      "date": "6/10/2022",
```

```
"comment": "Out in the dusty middle of nowhere, political posturing, everything is extremely boring until the main dude rides a giant sand worm.",
"rating": 4,
"owner_id": 5867703839490048,
"book_id": null,
"id": 5930285942177792,
"self": "http://montanek-portfolio.uw.r.appspot.com/reviews/5930285942177792"
}},
"next": "http://montanek-portfolio.uw.r.appspot.com/reviews?limit=5&offset=5"
}
}
```

### Failures

Status: 406 Unsupported Request.

```
{
"Error": "The request is missing the Accept Header"
}
```

Status: 406 Unsupported Request.

```
{
"Error": "The request Accept Header must be 'application/json'"
}
```

Status: 401 Missing Authorization.

```
{
"Error": "The request is missing authorization"
}
```

Status: 401 Invalid Authorization.

```
{
"Error": "The request authorization is invalid"
}
```

Status: 401 Invalid User Account for Request.

```
{
"Error": "No User Account exists for this login"
}
```

Status: 405 Invalid Method for Request.

```
{
"The Request Method is not allowed. Please use 'POST' or 'GET' request methods only"
}
```

## Get a Book

Allows you to get an existing book

GET /books/:book_id
---------------------

### Request

#### Path Parameters

Name	Description
book_id	ID of the book

### Request Body

None

### Protected

Yes

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li><li>Invalid JWT Authorization Token.</li></ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for the following: <ul style="list-style-type: none"><li>The Request Method will not support "POST".</li></ul>
Failure	403 Invalid Authorization for Request Object	A 403 status code will be returned for violating the following: <ul style="list-style-type: none"><li>The User must be the owner of the Book.</li></ul>

		· The Book must exist in the database.
--	--	--

### Response Examples

- Our App will allow you to view books that you own. You may not create, view, edit, or delete books owned by other users.
- Our App will provide you with the direct URL address to the book. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- You may connect a Review to this book via the [Link a Review to a Book](#) route below.

### Success

Status: 200 OK

```
{
  "title": "Pride and Prejudice",
  "review_ids": [],
  "author": "Jane Austen",
  "genre": "romance",
  "owner_id": 5867703839490048,
  "id": 4684183624482816,
  "self": "http://montanek-portfolio.uw.r.appspot.com/books/4684183624482816"
}
```

### Failures

Status: 406 Unsupported Request.

```
{
  "Error": "The request is missing the Accept Header"
}
```

Status: 406 Unsupported Request.

```
{
  "Error": "The request Accept Header must be 'application/json'"
}
```

Status: 401 Missing Authorization.

```
{
  "Error": "The request is missing authorization"
}
```

Status: 401 Invalid Authorization.

```
{
  "Error": "The request authorization is invalid"
}
```

Status: 401 Invalid User Account for Request.  <pre>{   "Error": "No User Account exists for this login" }</pre>
Status: 405 Invalid Method for Request.  <pre>{   "The Request Method is not allowed. Please use 'GET', 'PUT', 'PATCH', or 'DELETE' request methods only" }</pre>
Status: 403 Missing Object.  <pre>{   "Error": "No book with this book_id exists" }</pre>
Status: 403 Invalid Authorization for Object.  <pre>{   "Error": "The book is not owned by you" }</pre>



## Edit a Book

Allows you to edit a book.

PATCH /books/:book_id
-----------------------

### Request

#### Request JSON Attributes

Name	Description	Required?
title	The title of the book.	No
author	The author of the book.	No
genre	The author of the book. <i>E.g. "fantasy"</i>	No

### Request Body

Required

### Protected

Yes

### Request Body Format

JSON

### Request Body Example

```
{
  "title": "The Name of the Wind",
  "author": "Patrick Rothfuss",
  "genre": "fantasy"
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li><li>Invalid JWT Authorization Token.</li></ul>

Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>· Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for the following: <ul style="list-style-type: none"><li>· The Request Method will not support "POST".</li></ul>
Failure	403 Invalid Authorization for Request Object	A 403 status code will be returned for violating the following: <ul style="list-style-type: none"><li>· The User must be the owner of the Book.</li><li>· The Book must exist in the database.</li></ul>

### Response Examples

- Our App will allow you to edit a book that you own. You may not create, view, edit, or delete books owned by other users.
- Our App will return the Book as edited, and provide you with the direct URL address to the Book. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- For account purposes, you may not edit the owner ID of the Book, or the ID given to the Book.
- You may connect a Review to this book via the [Link a Review to a Book](#) route below.
- Attributes that you edit include the "title", "author", and "genre".
- For PATCH edits, you may leave out any attributes that you do not wish to change. This will not replace or override any prior attributes unspecified in your request.

### Success

Status: 201 Created

```
{
  "title": "The Name of the Wind",
  "author": "Patrick Rothfuss",
  "genre": "fantasy",
  "review_ids": [],
  "owner_id": 6294026856693760,
  "id": 6582719123292160,
  "self": "http://montanek-portfolio.uw.r.appspot.com/books/6582719123292160"
}
```

### Failures

Status: 406 Unsupported Request.

<pre>{ "Error": "The request is missing the Accept Header" }</pre>
Status: 406 Unsupported Request.  <pre>{ "Error": "The request Accept Header must be 'application/json'" }</pre>
Status: 401 Missing Authorization.  <pre>{ "Error": "The request is missing authorization" }</pre>
Status: 401 Invalid Authorization.  <pre>{ "Error": "The request authorization is invalid" }</pre>
Status: 401 Invalid User Account for Request.  <pre>{ "Error": "No User Account exists for this login" }</pre>
Status: 405 Invalid Method for Request.  <pre>{ "The Request Method is not allowed. Please use 'GET', 'PUT', 'PATCH', or 'DELETE' request methods only" }</pre>
Status: 403 Missing Object.  <pre>{ "Error": "No book with this book_id exists" }</pre>
Status: 403 Invalid Authorization for Object.  <pre>{ "Error": "The book is not owned by you" }</pre>

## Edit a Book

Allows you to edit a book.

PUT /books/:book\_id

### Request

#### Path Parameters

Name	Description
book_id	ID of the book

#### Request Body

Required

#### Protected

Yes

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Description	Required?
name	The name of the book.	Yes
type	The type of the book. E.g., Sailbook, Catamaran, etc.	Yes
length	Length of the book in feet.	Yes

#### Request Body Example

```
{
  "title": "Good Omens",
  "author": "Neil Gaiman & Terry Pratchett",
  "genre": "fantasy"
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li></ul>

		<ul style="list-style-type: none"> <li>Invalid JWT Authorization Token.</li> </ul>
Failure	401 Invalid User Account for Request	<p>A 401 status code will be returned for the following:</p> <ul style="list-style-type: none"> <li>Missing user account connected with JWT Authorization Token.</li> </ul>
Failure	405 Invalid Method for Request	<p>A 405 status code will be returned for the following:</p> <ul style="list-style-type: none"> <li>The Request Method will not support "POST".</li> </ul>
Failure	403 Invalid Authorization for Request Object	<p>A 403 status code will be returned for violating the following:</p> <ul style="list-style-type: none"> <li>The User must be the owner of the Book.</li> <li>The Book must exist in the database.</li> </ul>

### Response Examples

- Our App will allow you to edit a book that you own. You may not create, view, edit, or delete books owned by other users.
- Our App will return the Book as edited, and provide you with the direct URL address to the Book. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- For account purposes, you may not edit the owner ID of the Book, or the ID given to the Book.
- You may connect a Review to this book via the [Link a Review to a Book](#) route below.
- Attributes that you edit include the "title", "author", and "genre".
- WARNING: For PUT edits, if you leave out any attributes that you do not wish to change, this will replace any prior attributes unspecified in your request. Because [Link a Review to a Book](#) and [Unlink a Review to a Book](#) provide you a way to edit the relationship between Books and Reviews, "review\_ids" attribute will remain unaffected by PUT.

### Success

Status: 201 Created

```
{
  "title": "Good Omens",
  "author": "Neil Gaiman & Terry Pratchett",
  "genre": "fantasy",
  "review_ids": [],
  "owner_id": 5867703839490048,
  "id": 5778115385098240,
  "self": "http://montanek-portfolio.uw.r.appspot.com/books/5778115385098240"
}
```

### Failures

Status: 406 Unsupported Request.  { "Error": "The request is missing the Accept Header" }
Status: 406 Unsupported Request.  { "Error": "The request Accept Header must be 'application/json'" }
Status: 401 Missing Authorization.  { "Error": "The request is missing authorization" }
Status: 401 Invalid Authorization.  { "Error": "The request authorization is invalid" }
Status: 401 Invalid User Account for Request.  { "Error": "No User Account exists for this login" }
Status: 405 Invalid Method for Request.  { "The Request Method is not allowed. Please use 'GET', 'PUT', 'PATCH', or 'DELETE' request methods only" }
Status: 403 Missing Object.  { "Error": "No book with this book_id exists" }
Status: 403 Invalid Authorization for Object.  { "Error": "The book is not owned by you" }

## Delete a Book

Allows you to delete a book. Note that if the book is currently in a review, deleting the book makes any connected review's "book\_id" attribute empty.

DELETE /books/:book_id
------------------------

### Request

#### Path Parameters

Name	Description
book_id	ID of the book

### Request Body

None

### Protected

Yes

### Response

No body

### Response Body Format

Success: No body

Failure: JSON

### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li><li>Invalid JWT Authorization Token.</li></ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for the following: <ul style="list-style-type: none"><li>The Request Method will not support "POST".</li></ul>

Failure	403 Invalid Authorization for Request Object	A 403 status code will be returned for violating the following: <ul style="list-style-type: none"><li>· The User must be the owner of the Book.</li><li>· The Book must exist in the database.</li></ul>
---------	--	--

### Response Examples

- Deleting the Book will also delete the “book\_id” attribute from any Reviews currently connected with the Book.

### Success

Status: 204 No Content
------------------------

### Failures

Status: 406 Unsupported Request.  <pre>{   "Error": "The request is missing the Accept Header" }</pre>
Status: 406 Unsupported Request.  <pre>{   "Error": "The request Accept Header must be 'application/json'" }</pre>
Status: 401 Missing Authorization.  <pre>{   "Error": "The request is missing authorization" }</pre>
Status: 401 Invalid Authorization.  <pre>{   "Error": "The request authorization is invalid" }</pre>
Status: 401 Invalid User Account for Request.  <pre>{   "Error": "No User Account exists for this login" }</pre>
Status: 405 Invalid Method for Request.  <pre>{   "The Request Method is not allowed. Please use 'GET', 'PUT', 'PATCH', or 'DELETE' request methods only" }</pre>
Status: 403 Missing Object.



```
{  
  "Error": "No book with this book_id exists"  
}
```

Status: 403 Invalid Authorization for Object.

```
{  
  "Error": "The book is not owned by you"  
}
```

## Create a Review

Allows you to create a new review.

POST /reviews
---------------

### Request

#### Path Parameters

None

#### Protected

Yes

#### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Description	Required?
date	The date of the last time a User has read the Book..	Yes
rating	The rating of the Book, with 1 being the lowest, and 5 being the highest.	Yes
comment	The comment from the User about the Book.	Yes

#### Request Body Example

```
{
  "date": "2/25/2019",
  "rating": 5,
  "comment": "Bro-mance 'enemies to lovers' are tasked with babysitting a kid so the world doesn't end. I will forever ship David Tennant and Michael Sheen."
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>

Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>· Missing JWT Authorization Token.</li><li>· Invalid JWT Authorization Token.</li></ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>· Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for violating the following: <ul style="list-style-type: none"><li>· The Request Method must either be "POST" or "GET".</li></ul>
Failure	403 Invalid Request Body	A 403 status code will be returned for violating the following: <ul style="list-style-type: none"><li>· Request body must have the required attributes "date", "rating", and "comment".</li></ul>

### Response Examples

- Our App will automatically generate an ID and store it with the entity being created. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- Our App will provide you with the direct URL address to the Review. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- Review attributes must have "date", "rating", and "comment". Extraneous attributes provided will be ignored.
- Our App will automatically place your owner ID as an attribute of the Review. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- Our App will automatically place an empty ID attribute for new books to be associated with the Review. You may connect a Book to this Review via the [Link a Review to a Book](#) route below.

### Success

Status: 201 Created
<pre>{   "date": "2/25/2019",   "rating": 5,   "comment": "Bro-mance 'enemies to lovers' are tasked with babysitting a kid so the world doesn't end. I will forever ship David Tennant and Michael Sheen.",   "book_id": null,   "owner_id": 5867703839490048,   "id": 5420191735349248,</pre>

```
"self": "http://montanek-portfolio.uw.r.appspot.com/reviews/5420191735349248"
}
```

### Failures

Status: 406 Unsupported Request.

```
{
  "Error": "The request is missing the Accept Header"
}
```

Status: 406 Unsupported Request.

```
{
  "Error": "The request Accept Header must be 'application/json'"
}
```

Status: 401 Missing Authorization.

```
{
  "Error": "The request is missing authorization"
}
```

Status: 401 Invalid Authorization.

```
{
  "Error": "The request authorization is invalid"
}
```

Status: 401 Invalid User Account for Request.

```
{
  "Error": "No User Account exists for this login"
}
```

Status: 405 Invalid Method for Request.

```
{
  "The Request Method is not allowed. Please use 'POST' or 'GET' request methods only"
}
```

Status: 403 Invalid Request Body.

```
{
  "Error": "The request object must use the required attributes"
}
```

Status: 403 Invalid Request Body.

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

## List all Reviews

List all the reviews.

GET /reviews
--------------

### Request

Path Parameters

None

Protected

Yes

Request Body

None

### Response

Response Body Format

JSON

### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>· Request Accept header must exist in Request headers.</li><li>· Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>· Missing JWT Authorization Token.</li><li>· Invalid JWT Authorization Token.</li></ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>· Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for violating the following: <ul style="list-style-type: none"><li>· The Request Method must either be "POST" or "GET".</li></ul>

## Response Examples

- Our App will allow you to view reviews that you own. You may not view reviews owned by other users.
- Our App will provide you with the direct URL address to each review. However, this is not an attribute that you may alter with POST, PUT, or PATCH.

## Success

Status: 200 OK

```
{
  "total items": 6,
  "reviews": {
    "reviews": [{
      "book_id": null,
      "rating": 5,
      "comment": "An overly optimistic portrayal of how humans would react to a deadly pandemic. Don't forget to read his zombie survival guide.",
      "owner_id": 6294026856693760,
      "date": "8/16/2005",
      "id": 5215165431676928,
      "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/5215165431676928"
    }, {
      "comment": "What would you do if you had the ability to create an entire world with the stroke of a pen? Extra nerd points for anyone who's ever played the video game.",
      "date": "6/1/2000",
      "book_id": null,
      "owner_id": 6294026856693760,
      "rating": 5,
      "id": 5456603260125184,
      "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/5456603260125184"
    }, {
      "owner_id": 6294026856693760,
      "date": "9/3/2007",
      "comment": "Old man with a self-importance the size of my TBR list reminiscing on his college years. Well written but rather annoying main character. Harry Potter for adults.",
      "book_id": null,
      "rating": 3,
      "id": 5459213929152512,
      "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/5459213929152512"
    }, {
      "comment": "Nerd Utopia for those of us elder Millennials that are stuck in the 80s and 90s. Dystopian novel about what happens if everyone buys their kids Oculus VR headsets.",
      "book_id": null,
      "owner_id": 6294026856693760,
```

```
{
  "rating": 4,
  "date": "8/16/2011",
  "id": 6019553213546496,
  "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/6019553213546496"
}, {
  "owner_id": 6294026856693760,
  "comment": "You want a daring heist? But make it Game of Thrones style? With a grimy cut-throat atmosphere in a shanty port town? Look no further! Don't forget to read the sequel.",
  "rating": 5,
  "book_id": null,
  "date": "11/2/2016",
  "id": 6314695447281664,
  "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/6314695447281664"
}],
"next": "http://montanek-portfolio.uw.r.appspot.com/reviews?limit=5&offset=5"
}
```

#### Failures

Status: 406 Unsupported Request.

```
{
  "Error": "The request is missing the Accept Header"
}
```

Status: 406 Unsupported Request.

```
{
  "Error": "The request Accept Header must be 'application/json'"
}
```

Status: 401 Missing Authorization.

```
{
  "Error": "The request is missing authorization"
}
```

Status: 401 Invalid Authorization.

```
{
  "Error": "The request authorization is invalid"
}
```

Status: 401 Invalid User Account for Request.

```
{
  "Error": "No User Account exists for this login"
}
```

Status: 405 Invalid Method for Request.

```
{  
  "The Request Method is not allowed. Please use 'POST' or 'GET' request methods  
  only"  
}
```



## Get a Review

Allows you to get an existing review.

GET /reviews/:review_id
-------------------------

### Request

#### Path Parameters

Name	Description
review_id	ID of the review

### Request Body

None

### Protected

Yes

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li><li>Invalid JWT Authorization Token.</li></ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for the following: <ul style="list-style-type: none"><li>The Request Method will not support "POST".</li></ul>
Failure	403 Invalid Authorization for Request Object	A 403 status code will be returned for violating the following: <ul style="list-style-type: none"><li>The User must be the owner of the Review.</li></ul>

		· The Review must exist in the database.
--	--	--

### Response Examples

- Our App will allow you to view reviews that you own. You may not create, view, edit, or delete reviews owned by other users.
- Our App will provide you with the direct URL address to the Review. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- You may connect a Book to this Review via the [Link a Review to a Book](#) route below.

### Success

Status: 200 OK
<pre>{   "comment": "Wizard Highschool drama, horcruxes, and a deeper look into who Severus Snape is.",   "date": "8/16/2005",   "rating": 5,   "owner_id": 6294026856693760,   "book_id": null,   "id": 6552400143843328,   "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/6552400143843328" }</pre>

### Failures

Status: 406 Unsupported Request.
<pre>{   "Error": "The request is missing the Accept Header" }</pre>
Status: 406 Unsupported Request.
<pre>{   "Error": "The request Accept Header must be 'application/json'" }</pre>
Status: 401 Missing Authorization.
<pre>{   "Error": "The request is missing authorization" }</pre>
Status: 401 Invalid Authorization.
<pre>{   "Error": "The request authorization is invalid" }</pre>

Status: 401 Invalid User Account for Request.  <pre>{   "Error": "No User Account exists for this login" }</pre>
Status: 405 Invalid Method for Request.  <pre>{   "The Request Method is not allowed. Please use 'GET', 'PUT', 'PATCH', or 'DELETE' request methods only" }</pre>
Status: 403 Missing Object.  <pre>{   "Error": "No review with this review_id exists" }</pre>
Status: 403 Invalid Authorization for Object.  <pre>{   "Error": "The review is not owned by you" }</pre>

## Edit a Review

Allows you to edit a review.

PATCH /reviews/:review_id
---------------------------

### Request

#### Path Parameters

Name	Description
review_id	ID of the review

### Request Body

Required

### Protected

Yes

### Request Body Format

JSON

#### Request JSON Attributes

Name	Description	Required?
date	The date of the last time a User has read the Book..	No
rating	The rating of the Book, with 1 being the lowest, and 5 being the highest.	No
comment	The comment from the User about the Book.	No

### Request Body Example

```
{
  "date": "4/06/2018",
  "rating": 4,
  "comment": "I know we've all heard this before. Die Hard isn't a Christmas movie. John McClane sneaks around a tower at night avoiding Alan Rickman. It's a Harry Potter movie."
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following:

		<ul style="list-style-type: none"> <li>Request Accept header must exist in Request headers.</li> <li>Request Accept head must be "application/json" or "*/*".</li> </ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"> <li>Missing JWT Authorization Token.</li> <li>Invalid JWT Authorization Token.</li> </ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"> <li>Missing user account connected with JWT Authorization Token.</li> </ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for the following: <ul style="list-style-type: none"> <li>The Request Method will not support "POST".</li> </ul>
Failure	403 Invalid Authorization for Request Object	A 403 status code will be returned for violating the following: <ul style="list-style-type: none"> <li>The User must be the owner of the Review.</li> <li>The Review must exist in the database.</li> </ul>

### Response Examples

- Our App will allow you to edit a Review that you own. You may not create, view, edit, or delete Reviews owned by other users.
- Our App will return the Review as edited, and provide you with the direct URL address to the Review. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- For account purposes, you may not edit the owner ID of the Review, or the ID given to the Review.
- You may connect a Book to this Review via the [Link a Review to a Book](#) route below.
- Attributes that you edit include the "date", "rating", and "comment".
- For PATCH edits, you may leave out any attributes that you do not wish to change. This will not replace or override any prior attributes unspecified in your request.
- .

### Success

Status: 201 Created  <pre>{   "rating": 4,   "date": "4/06/2018",   "book_id": null,   "owner_id": 6294026856693760,</pre>
--

```
"comment": "I know we've all heard this before. Die Hard isn't a Christmas movie. John McClane sneaks around a tower at night avoiding Alan Rickman. It's a Harry Potter movie.",
"id": 6552400143843328,
"self": "http://montanek-portfolio.uw.r.appspot.com/reviews/6552400143843328"
}
```

### Failures

Status: 406 Unsupported Request.

```
{
  "Error": "The request is missing the Accept Header"
}
```

Status: 406 Unsupported Request.

```
{
  "Error": "The request Accept Header must be 'application/json'"
}
```

Status: 401 Missing Authorization.

```
{
  "Error": "The request is missing authorization"
}
```

Status: 401 Invalid Authorization.

```
{
  "Error": "The request authorization is invalid"
}
```

Status: 401 Invalid User Account for Request.

```
{
  "Error": "No User Account exists for this login"
}
```

Status: 405 Invalid Method for Request.

```
{
  "The Request Method is not allowed. Please use 'GET', 'PUT', 'PATCH', or 'DELETE' request methods only"
}
```

Status: 403 Missing Object.

```
{
  "Error": "No review with this review_id exists"
}
```

Status: 403 Invalid Authorization for Object.

```
{
  "Error": "The review is not owned by you"
}
```

}
---

## Edit a Review

Allows you to edit a review.

PUT /reviews/:review\_id

### Request

#### Path Parameters

Name	Description
review_id	ID of the review

### Request Body

Required

### Protected

Yes

### Request Body Format

JSON

#### Request JSON Attributes

Name	Description	Required?
date	The date of the last time a User has read the Book..	No
rating	The rating of the Book, with 1 being the lowest, and 5 being the highest.	No
comment	The comment from the User about the Book.	No

### Request Body Example

```
{
  "rating": 5,
  "comment": "Nevermind I really liked this book."
}
```

## Response

### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li></ul>



		<ul style="list-style-type: none"><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li><li>Invalid JWT Authorization Token.</li></ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for the following: <ul style="list-style-type: none"><li>The Request Method will not support "POST".</li></ul>
Failure	403 Invalid Authorization for Request Object	A 403 status code will be returned for violating the following: <ul style="list-style-type: none"><li>The User must be the owner of the Review.</li><li>The Review must exist in the database.</li></ul>

### Response Examples

- Our App will allow you to edit a Review that you own. You may not create, view, edit, or delete Reviews owned by other users.
- Our App will return the Review as edited, and provide you with the direct URL address to the Review. However, this is not an attribute that you may alter with POST, PUT, or PATCH.
- For account purposes, you may not edit the owner ID of the Review, or the ID given to the Review.
- You may connect a Book to this Review via the [Link a Review to a Book](#) route below.
- Attributes that you edit include the "date", "rating", and "comment".
- WARNING: For PUT edits, if you leave out any attributes that you do not wish to change, this will replace any prior attributes unspecified in your request. Because [Link a Review to a Book](#) and [Unlink a Review to a Book](#) provide you a way to edit the relationship between Books and Reviews, "book\_id" attribute will remain unaffected by PUT.

### Success

Status: 201 Created

```
{  
  "comment": "Nevermind I really liked this book.",  
  "book_id": null,  
  "rating": 5,
```

```
"date": "2/13/2006",  
"owner_id": 5867703839490048,  
"id": 4750879735414784,  
"self": "http://montanek-portfolio.uw.r.appspot.com/reviews/4750879735414784"  
}
```

### Failures

Status: 406 Unsupported Request.

```
{  
"Error": "The request is missing the Accept Header"  
}
```

Status: 406 Unsupported Request.

```
{  
"Error": "The request Accept Header must be 'application/json'"  
}
```

Status: 401 Missing Authorization.

```
{  
"Error": "The request is missing authorization"  
}
```

Status: 401 Invalid Authorization.

```
{  
"Error": "The request authorization is invalid"  
}
```

Status: 401 Invalid User Account for Request.

```
{  
"Error": "No User Account exists for this login"  
}
```

Status: 405 Invalid Method for Request.

```
{  
"The Request Method is not allowed. Please use 'GET', 'PUT', 'PATCH', or 'DELETE'  
request methods only"  
}
```

Status: 403 Missing Object.

```
{  
"Error": "No review with this review_id exists"  
}
```

Status: 403 Invalid Authorization for Object.

```
{  
  "Error": "The review is not owned by you"  
}
```

## Delete a Review

Allows you to delete a review. If the review being deleted has a book, the book "review\_ids" array will no longer include the deleted Review in the array list.

DELETE /reviews/:review_id
----------------------------

### Request

#### Path Parameters

Name	Description
review_id	ID of the review

### Request Body

None

### Protected

Yes

### Response

No body

### Response Body Format

Success: No body

Failure: JSON

### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li><li>Invalid JWT Authorization Token.</li></ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for the following: <ul style="list-style-type: none"><li>The Request Method will not support "POST".</li></ul>

Failure	403 Invalid Authorization for Request Object	A 403 status code will be returned for violating the following: <ul style="list-style-type: none"> <li>· The User must be the owner of the Review.</li> <li>· The Review must exist in the database.</li> </ul>
---------	--	---

### Response Examples

- Deleting the Review will also delete the entry in the “review\_ids” attribute array list from any Book currently connected with the Review.

### Success

Status: 204 No Content
------------------------

### Failures

Status: 406 Unsupported Request.  <pre>{   "Error": "The request is missing the Accept Header" }</pre>
Status: 406 Unsupported Request.  <pre>{   "Error": "The request Accept Header must be 'application/json'" }</pre>
Status: 401 Missing Authorization.  <pre>{   "Error": "The request is missing authorization" }</pre>
Status: 401 Invalid Authorization.  <pre>{   "Error": "The request authorization is invalid" }</pre>
Status: 401 Invalid User Account for Request.  <pre>{   "Error": "No User Account exists for this login" }</pre>
Status: 405 Invalid Method for Request.  <pre>{   "The Request Method is not allowed. Please use 'GET', 'PUT', 'PATCH', or 'DELETE' request methods only" }</pre>
Status: 403 Missing Object.

```
{  
  "Error": "No review with this review_id exists"  
}
```

Status: 403 Invalid Authorization for Object.

```
{  
  "Error": "The review is not owned by you"  
}
```

## Link a Review to a Book

Book is connected with a Review.

PUT /books/:book_id/reviews/:review_id
--

### Request

#### Path Parameters

Name	Description
review_id	ID of the review
book_id	ID of the book

### Request Body

None

### Protected

Yes

### Response

No body

### Response Body Format

Success: No body

Failure: JSON

### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li><li>Invalid JWT Authorization Token.</li></ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for the following:

		<ul style="list-style-type: none"> <li>The Request Method will not support "POST", "PATCH", OR "GET".</li> </ul>
Failure	403 Invalid Authorization for Request Object	<p>A 403 status code will be returned for violating the following:</p> <ul style="list-style-type: none"> <li>The User must be the owner of the Review.</li> <li>The Review must exist in the database.</li> <li>The User must be the owner of the Book.</li> <li>The Book must exist in the database.</li> </ul>
Failure	403 Request Object Already Connected	<p>A 403 status code will be returned for violating the following:</p> <ul style="list-style-type: none"> <li>The Book is already connected with the Review.</li> </ul>

## Response Examples

- Our App will allow you to link a Book with a Review. You may not link Books or Reviews owned by other users.
- Our App will return the Review and the Book as edited, and provide you with the direct URL address to both. However, these attributes cannot be altered with POST, PUT, or PATCH.
- For account purposes, you may not edit the owner ID of the Review, or the ID given to the Review.
- You may sever the connection between this Book and this Review via the [Unlink a Review to a Book](#) route below.
- A Review cannot be assigned to multiple Books.
- A Book may be assigned to multiple Reviews.

## Response Examples

### Success

<p>Status: 201 Created</p> <pre>{   "book": {     "title": "Dune",     "author": "Frank Herbert",     "id": 4827211370594304,     "owner_id": 5867703839490048,     "review_ids": [5930285942177792],     "genre": "sci-fi",     "self": "http://montanek-portfolio.uw.r.appspot.com/books/4827211370594304"   },   "review": {     "date": "6/10/2022",     "rating": 4,</pre>
---



```
{
  "id": 5930285942177792,
  "comment": "Out in the dusty middle of nowhere, political posturing, everything is extremely boring until the main dude rides a giant sand worm.",
  "book_id": 4827211370594304,
  "owner_id": 5867703839490048,
  "self": "http://montanek-portfolio.uw.r.appspot.com/reviews/5930285942177792"
}
```

### Failures

Status: 406 Unsupported Request.

```
{
  "Error": "The request is missing the Accept Header"
}
```

Status: 406 Unsupported Request.

```
{
  "Error": "The request Accept Header must be 'application/json'"
}
```

Status: 401 Missing Authorization.

```
{
  "Error": "The request is missing authorization"
}
```

Status: 401 Invalid Authorization.

```
{
  "Error": "The request authorization is invalid"
}
```

Status: 401 Invalid User Account for Request.

```
{
  "Error": "No User Account exists for this login"
}
```

Status: 405 Invalid Method for Request.

```
{
  "The Request Method is not allowed. Please use 'PUT' or 'DELETE' request methods only"
}
```

Status: 403 Missing Object - Review.

```
{
  "Error": "No review with this review_id exists"
}
```

Status: 403 Invalid Authorization for Object - Review.

<pre>{   "Error": "The review is not owned by you" }</pre>
Status: 403 Missing Object- Book.  <pre>{   "Error": "No book with this book_id exists" }</pre>
Status: 403 Invalid Authorization for Object- Book.  <pre>{   "Error": "The book is not owned by you" }</pre>
Status: 403 Request Object Already Connected  <pre>{   "Error": "The book and review are already connected" }</pre>

## Unlink a Review to a Book

Book is disconnected from the Review.

DELETE /books/:book_id/reviews/:review_id
---

### Request

#### Path Parameters

Name	Description
review_id	ID of the review
book_id	ID of the book

### Request Body

None

### Protected

Yes

### Response

No body

### Response Body Format

Success: No body

Failure: JSON

### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	406 Unsupported Request	A 406 status code will be returned for violating the following: <ul style="list-style-type: none"><li>Request Accept header must exist in Request headers.</li><li>Request Accept head must be "application/json" or "*/*".</li></ul>
Failure	401 Invalid Authorization in Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing JWT Authorization Token.</li><li>Invalid JWT Authorization Token.</li></ul>
Failure	401 Invalid User Account for Request	A 401 status code will be returned for the following: <ul style="list-style-type: none"><li>Missing user account connected with JWT Authorization Token.</li></ul>
Failure	405 Invalid Method for Request	A 405 status code will be returned for the following:

		<ul style="list-style-type: none"> <li>The Request Method will not support "POST", "PATCH", OR "GET".</li> </ul>
Failure	403 Invalid Authorization for Request Object	<p>A 403 status code will be returned for violating the following:</p> <ul style="list-style-type: none"> <li>The User must be the owner of the Review.</li> <li>The Review must exist in the database.</li> <li>The User must be the owner of the Book.</li> <li>The Book must exist in the database.</li> </ul>
Failure	403 Request Object Already Connected	<p>A 403 status code will be returned for violating the following:</p> <ul style="list-style-type: none"> <li>The Book is not currently connected with the Review.</li> </ul>

### Response Examples

- Our App will allow you to unlink a Book with a Review. You may not unlink Books or Reviews owned by other users.
- You may reconnect the connection between this Book and this Review via the [Link a Review to a Book](#) route below.
- Unlinking a Review cannot sever relationships of other reviews to the Book.

### Success

Status: 204 No Content
------------------------

### Failures

<p>Status: 406 Unsupported Request.</p> <pre>{   "Error": "The request is missing the Accept Header" }</pre>
<p>Status: 406 Unsupported Request.</p> <pre>{   "Error": "The request Accept Header must be 'application/json'" }</pre>
<p>Status: 401 Missing Authorization.</p> <pre>{   "Error": "The request is missing authorization" }</pre>
<p>Status: 401 Invalid Authorization.</p> <pre>{</pre>

<pre>"Error": "The request authorization is invalid" }</pre>
<p>Status: 401 Invalid User Account for Request.</p> <pre>{ "Error": "No User Account exists for this login" }</pre>
<p>Status: 405 Invalid Method for Request.</p> <pre>{ "The Request Method is not allowed. Please use 'PUT' or 'DELETE' request methods only" }</pre>
<p>Status: 403 Missing Object - Review.</p> <pre>{ "Error": "No review with this review_id exists" }</pre>
<p>Status: 403 Invalid Authorization for Object - Review.</p> <pre>{ "Error": "The review is not owned by you" }</pre>
<p>Status: 403 Missing Object- Book.</p> <pre>{ "Error": "No book with this book_id exists" }</pre>
<p>Status: 403 Invalid Authorization for Object- Book.</p> <pre>{ "Error": "The book is not owned by you" }</pre>
<p>Status: 403 Request Object Already Connected</p> <pre>{ "Error": "The book and review are already disconnected" }</pre>

## Delete all Books and Reviews

NOTE: This is a route for testing purposes only. This will delete all Books and Reviews from the database as a total reset.

DELETE /reset/booksreviews
----------------------------

### Request

#### Path Parameters

None

#### Protected

No

#### Request Body

None

### Response

No body

#### Response Body Format

Success: No body

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	

### Response Examples

- This route is for testing purposes only. You will not need to apply “application/json” to your Accept header or provide a JWT.
- This route resets Book and Reviews for ALL USERS. Please be aware of this reset scope.
- This route does not provide any error status codes.

#### Success

Status: 204 No Content
------------------------

## Delete all Users

NOTE: This is a route for testing purposes only. This will delete all Users from the database as a total reset.

DELETE /reset/users
---------------------

### Request

#### Path Parameters

None

#### Protected

No

#### Request Body

None

### Response

No body

#### Response Body Format

Success: No body

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	

### Response Examples

- This route is for testing purposes only. You will not need to apply “application/json” to your Accept header or provide a JWT.
- This route resets Users for ALL ACCOUNTS. Please be aware of this reset scope.
- This route does not provide any error status codes.

#### Success

Status: 204 No Content
------------------------