

Portfolio Assignment: API Spec

<https://portfolio-skinneem.uc.r.appspot.com/>

Author: Emilio Skinner

Last Update: Dec 5 2022

Change log.....	1
Data Model	2
View all Libraries.....	4
Create a Collection.....	6
View all Collections.....	9
Delete a Collection.....	11
Edit a Collection (Patch).....	12
Edit a Collection (Put).....	15
Create a Media.....	18
View all Media.....	21
Delete a Media.....	23
Edit a Media (Patch).....	24
Edit a Media (Put).....	27
Add Media to Collection.....	30
Delete Media from Collection.....	31
View a Collection's Media.....	32

Change log

Version	Change	Date
1.0	Initial version.	Dec 5, 2022

Data Model

The app stores two non-user entities in Datastore, called Collections and Media. It stores one user entity, called Libraries.

Libraries (User Entity)

Property	Data Type	Notes	Required?
id	Integer	The Library's id. Datastore automatically generates it.	n/a
name	String	Name of the Library.	Yes
type	String	Type of the Library. i.e. Academic, Public, National, etc.	Yes
capacity	Integer	Capacity of the Library. i.e. a number	Yes
owner	String	Sub value of the JWT token corresponding to the Library (aka user) that owns the Collection. Unique identifier of the Library.	Yes

Collections (Protected Non-User Entity)

Property	Data Type	Notes	Required?
id	Integer	The Collection's id. Datastore automatically generates it.	n/a
name	String	Name of the Collection. Between 0 and 100 alphabetical characters. Must be unique.	Yes
capacity	Integer	Capacity of the Collection. Between 0 and 18 digits of type integer.	Yes
last_updated	Date	Date that the Collection was last updated. Must be a string in format MM-DD-YYYY.	Yes
owner	String	Sub value of the JWT token corresponding to the Library (aka user) that owns the Collection. Unique identifier of the Library.	Yes
media	List	List of all Media that the Collection contains.	No

Media (Non-User Entity)

Property	Data Type	Notes	Required?
id	Integer	The Media's id. Datastore automatically generates it.	n/a
name	String	Name of the Media. Between 0 and 100 alphabetical characters. Must be unique.	Yes
type	String	Type of the Media. Between 0 and 50 alphabetical characters.	Yes
release_date	Date	Date that the Media was first released. Must be a string in format MM-DD-YYYY.	Yes
collection	Dictionary	The Collection that the Media belongs to, if any.	No

- **A collection and a media are a 1:M relationship.**
 - **A collection can contain multiple pieces of media but a media can only be attached to one collection at a time.**
- **A library is uniquely identified by the 'sub' value of the JWT token created upon initially signing up and creating an account.**
- **A collection is a protected non-user entity resource. The JWT token must be provided in the auth header by the user making the request to make changes to the 'collection' resource.**
- **The application maps a supplied JWT to the identifier of a user by the 'sub' value of the JWT which is stored in the 'owner' property of the user.**
- **A library and a collection are a 1:M relationship.**
 - **A library can contain multiple collections but a collection can only be attached to one library at a time.**

[View all Libraries](#)

Allows the retrieval of all existing libraries.

GET /libraries

[Request](#)

[Path Parameters](#)

None

[Request Headers](#)

Accept: application/json

[Request Body](#)

None

[Response](#)

[Response Body Format](#)

JSON

[Response Statuses](#)

Outcome	Status Code	Notes
Success	200 OK	

[Response Examples](#)

[Success](#)

Status: 200 OK

```
[
  {
    "id": 123,
    "name": "Noel Memorial Library",
    "type": "Academic",
    "capacity": 1000,
    "owner": "auth0|637028fdf2ea62cc6a5f3de0"
  },
  {
    "id": 456,
    "name": "Magale Library",
    "type": "Public",
    "capacity": 250,
    "owner": "auth0|636f72c31c38a373c603f8ee"
  },
  {
    "id": 789,
    "name": "Library of Congress",
    "type": "National",
```

```
"capacity": 5000,  
"owner": "auth0|636f72c31c38a373c603f8ee"  
}
```

```
]
```

Create a Collection

Allows the creation of a new collection.

POST /collections

Request

Path Parameters

None

Request Headers

Content-Type: application/json

Accept: application/json

Authorization: Bearer-Token (id_token value of the JWT corresponding to the Library that owns this Collection)

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Data Type	Description	Required?
name	String	Name of the collection. Between 0 and 100 alphabetical characters. Must be unique.	Yes
capacity	Integer	Capacity of the collection. Between 0 and 18 digits of type integer.	Yes
last_updated	Date	Date that the Collection was last updated. Must be a string in format MM-DD-YYYY.	Yes

Request Body Example

```
{
  "name": "Main",
  "capacity": 100,
  "last_updated": "12-01-2022"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
---------	-------------	-------

Success	201 Created	
Failure	400 Bad Request	<p>If an extraneous attribute (such as ID) is included in the request body, 400 status code is returned with the error message: "The request object contains an extraneous attribute."</p> <p>If a required attribute is missing, 400 status code is returned with the error message: "The request object is missing at least one of the required attributes".</p> <p>If an attribute is invalid, 400 status code is returned with the error message: "The request object contained an invalid value."</p>
Failure	403 Forbidden	If the name already exists, 403 status code is returned with the error message: "The name is already taken by an existing collection."
Failure	415 Unsupported Media Type	If the request header's Content-Type is not set to 'application/json', 415 status code is returned with the error message: "The server only supports JSON format."

Response Examples

- Datastore automatically generates an ID and stores it with the entity being created. Your app must send back in the response body: the ID and self-link to the resource as shown below.

Response Body Example

Success

```
Status: 201 Created
{
  "id": 123,
  "name": "Main",
  "capacity": 100,
  "last_updated": "12-01-2022",
  "media": [ ],
  "self": "https://portfolio-skinneem.uc.r.appspot.com/collections/123"
}
```

Failure

```
Status: 400 Bad Request
{
  "400 Bad Request": "The request object contains an extraneous attribute."
}
```

Status: 403 Forbidden

```
{  
  "403 Forbidden": "The name is already taken by an existing collection."  
}
```

Status: 415 Unsupported Media Type

```
{  
  "415 Unsupported Media Type": "The server only supports JSON format."  
}
```


View all Collections

Allows the retrieval of all existing collections.

GET /collections

Request

Path Parameters

None

Request Headers

Accept: application/json

Authorization: Bearer-Token (id_token value of the JWT corresponding to the Library that owns this Collection)

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If the request does not accept JSON, 406 status code is returned with the error message: "The media type is not accepted by the server." This is because the server only handles JSON, but the request failed to indicate it could accept that format.

Response Examples

Success

```
Status: 200 OK
[
  collections: {
    "id": 123,
    "name": "Main",
    "capacity": 100,
    "last_updated": "12-01-2022",
    "media": [ ],
    "self": "https://portfolio-skinneem.uc.r.appspot.com/collections/123"
  }
  {
    "id": 456,
```

```
"name": "Periodicals",
"capacity": 25,
"last_updated": "11-01-2022",
"media": [ ],
"self": "https://portfolio-skinneem.uc.r.appspot.com/collections/456"
}
{
  "id": 789,
  "name": "References",
  "capacity": 250,
  "last_updated": "10-01-2022",
  "media": [ ],
  "self": "https://portfolio-skinneem.uc.r.appspot.com/collections/789"
}
},
total_items: 3
next:
]
```

Status: 406 Not Acceptable

```
{
  "406 Not Acceptable": "The media type is not accepted by the server."
}
```

Delete a Collection

Allows the deletion of an existing collection.

```
DELETE /collections/:collection_id
```

Request

Path Parameters

Name	Description
collection_id	The collection's ID

Request Headers

Authorization: Bearer-Token (id_token value of the JWT corresponding to the Library that owns this Collection)

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	403 Forbidden	If the collection is owned by another library, 403 status code is returned with the error message: "Collection is owned by someone else."
Failure	404 Not Found	If the collection does not exist, 404 status code is returned with the error message: "No collection with this collection_id exists".

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 403 Forbidden
```

```
{
  "403 Forbidden": "Collection is owned by someone else"
}
```

```
Status: 404 Not Found
```

```
{
```

```
"404 Not Found": "No collection with this collection_id exists"
}
```

Edit a Collection (Patch)

Allows the partial update of an existing collection.

```
PATCH /collections/:collection_id
```

Request

Path Parameters

Name	Description
collection_id	The collection's ID

Request Headers

Content-Type: application/json

Accept: application/json

Authorization: Bearer-Token (id_token value of the JWT corresponding to the Library that owns this Collection)

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Data Type	Description	Required?
name	String	Name of the collection. Between 0 and 100 alphabetical characters. Must be unique.	Yes
capacity	Integer	Capacity of the collection. Between 0 and 18 digits of type integer.	Yes
last_updated	Date	Date that the Collection was last updated. Must be a string in format MM-DD-YYYY.	Yes

Request Body Example

- Partial updates can consist of: name & capacity, name & last_updated, capacity & last_updated, or just one of: name, capacity, or last_updated. Attributes must be valid. Name must be unique.

```
{
  "name": "Audio/Visual",
  "capacity": 80
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If an extraneous attribute (such as ID, or having more than 2 attributes) is included in the request body, 400 status code is returned with the error message: "The request object contains an extraneous attribute." If an attribute is invalid, 400 status code is returned with the error message: "The request object contained an invalid value."
Failure	403 Forbidden	If the name already exists, 403 status code is returned with the error message: "The name is already taken by an existing collection."
Failure	404 Not Found	If the collection does not exist, 404 status code is returned with the error message: "No collection with this collection_id exists".
Failure	415 Unsupported Media Type	If the request header's Content-Type is not set to 'application/json', 415 status code is returned with the error message: "The server only supports JSON format."

Response Examples

- Datastore automatically generates an ID and stores it with the entity being created. Your app must send back in the response body: the ID and self-link to the resource as shown below.
- Partial updates cannot contain all 3 attributes. It will return 400 status code.

Success

Status: 200 OK

```
{
  "name": "Audio/Visual",
  "capacity": 80
}
```

Failure

Status: 400 Bad Request

```
{
  "400 Bad Request": "The request object contained an extraneous attribute."
}
```

Status: 403 Forbidden

```
{  
  "403 Forbidden": "The name is already taken by an existing collection."  
}
```

Status: 404 Not Found

```
{  
  "404 Not Found": "No collection with this collection_id exists"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "415 Unsupported Media Type": "The server only accepts JSON format."  
}
```

Edit a Collection (Put)

Replaces the attributes of an existing collection.

PUT /collections/:collection_id

Request

Path Parameters

Name	Description
collection_id	The collection's ID

Request Headers

Content-Type: application/json

Accept: application/json

Authorization: Bearer-Token (id_token value of the JWT corresponding to the Library that owns this Collection)

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Data Type	Description	Required?
name	String	Name of the collection. Between 0 and 100 alphabetical characters. Must be unique.	Yes
capacity	Integer	Capacity of the collection. Between 0 and 18 digits of type integer.	Yes
last_updated	Date	Date that the Collection was last updated. Must be a string in format MM-DD-YYYY.	Yes

Request Body Example

- All 3 attributes are included in the request body when updating an existing boat with PUT.

```
{
  "name": "Main",
  "capacity": 70,
  "last_updated": "12-01-2021"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
---------	-------------	-------

Success	303 See Other	
Failure	400 Bad Request	<p>If an extraneous attribute (such as ID, or having more than 2 attributes) is included in the request body, 400 status code is returned with the error message: "The request object contains an extraneous attribute."</p> <p>If an attribute is invalid, 400 status code is returned with the error message: "The request object contained an invalid value."</p>
Failure	403 Forbidden	If the name already exists, 403 status code is returned with the error message: "The name is already taken by an existing collection."
Failure	404 Not Found	If the collection does not exist, 404 status code is returned with the error message: "No collection with this collection_id exists".
Failure	415 Unsupported Media Type	If the request header's Content-Type is not set to 'application/json', 415 status code is returned with the error message: "The server only supports JSON format."

Response Examples

- Datastore automatically generates an ID and stores it with the entity being created. Your app must send back in the response body: the ID and self-link to the resource as shown below.
- 303 See Other: The Location header of the response must be set to the self-link of the resource.

Success

```
Status: 303 See Other
{
  "id": 123,
  "name": "Main",
  "capacity": 70,
  "last_updated": "12-01-2021",
  "self": "https://portfolio-skinneem.uc.r.appspot.com/collections/123"
}
```

Failure

```
Status: 400 Bad Request
{
  "400 Bad Request": "The request object contained an extraneous attribute."
}
```


Status: 403 Forbidden

```
{  
  "403 Forbidden": "The name is already taken by an existing collection."  
}
```

Status: 404 Not Found

```
{  
  "404 Not Found": "No collection with this collection_id exists"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "415 Unsupported Media Type": "The server only accepts JSON format."  
}
```

Create a Media

Allows the creation of a new media.

POST /media

Request

Path Parameters

None

Request Headers

Content-Type: application/json

Accept: application/json

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Data Type	Description	Required?
name	String	Name of the Media. Between 0 and 100 alphabetical characters. Must be unique.	Yes
type	String	Type of the Media. Between 0 and 50 alphabetical characters.	Yes
release_date	Date	Date that the Media was first released. Must be a string in format MM-DD-YYYY.	Yes
collection	Dictionary	The Collection that the Media belongs to, if any.	No

Request Body Example

```
{
  "name": "Fundamentals of JavaScript",
  "type": "Textbook",
  "release_date": "07-31-2005"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	

Failure	400 Bad Request	<p>If an extraneous attribute (such as ID) is included in the request body, 400 status code is returned with the error message: “The request object contains an extraneous attribute.”</p> <p>If a required attribute is missing, 400 status code is returned with the error message: “The request object is missing at least one of the required attributes”.</p> <p>If an attribute is invalid, 400 status code is returned with the error message: “The request object contained an invalid value.”</p>
Failure	403 Forbidden	If the name already exists, 403 status code is returned with the error message: “The name is already taken by an existing media.”
Failure	415 Unsupported Media Type	If the request header’s Content-Type is not set to ‘application/json’, 415 status code is returned with the error message: “The server only supports JSON format.”

Response Examples

- Datastore automatically generates an ID and stores it with the entity being created. Your app must send back in the response body: the ID and self-link to the resource as shown below.

Response Body Example

Success

```
Status: 201 Created
{
  "id": 123,
  "name": "Fundamentals of JavaScript",
  "type": "Textbook",
  "release_date": "07-31-2005"
  "self": "https://portfolio-skinneem.uc.r.appspot.com/media/123"
}
```

Failure

```
Status: 400 Bad Request
{
  "400 Bad Request": "The request object contains an extraneous attribute."
}
```

```
Status: 403 Forbidden
{
```

```
"403 Forbidden": "The name is already taken by an existing media."
```

```
}
```

```
Status: 415 Unsupported Media Type
```

```
{
```

```
"415 Unsupported Media Type": "The server only supports JSON format."
```

```
}
```

View all Media

Allows the retrieval of all existing media.

GET /media

Request

Path Parameters

None

Request Headers

Accept: application/json

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If the request does not accept JSON, 406 status code is returned with the error message: "The media type is not accepted by the server." This is because the server only handles JSON, but the request failed to indicate it could accept that format.

Response Examples

Success

```
Status: 200 OK
[
  media: {
    "id": 123,
    "name": "Fundamentals of JavaScript",
    "type": "Textbook",
    "release_date": "07-31-2005",
    "collection": { }
    "self": "https://portfolio-skinneem.uc.r.appspot.com/media/123"
  }
  {
    "id": 456,
    "name": "The Basics of Python: 101",
```

```
"type": "Textbook",
"release_date": "08-31-2005",
"collection": { }
"self": "https://portfolio-skinneem.uc.r.appspot.com/media/456"
}
{
"id": 789,
"name": "C++ A Hero's Journey",
"type": "Textbook",
"release_date": "02-14-2009",
"collection": { }
"self": "https://portfolio-skinneem.uc.r.appspot.com/media/789"
}
},
total_items: 3
next:
]
```

Status: 406 Not Acceptable

```
{
"406 Not Acceptable": "The media type is not accepted by the server."
}
```

Delete a Media

Allows the deletion of an existing media.

```
DELETE /media/:media_id
```

Request

Path Parameters

Name	Description
media_id	The media's ID

Request Headers

None

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	If the media does not exist, 404 status code is returned with the error message: "No media with this media_id exists".

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found
```

```
{
  "404 Not Found": "No media with this media_id exists"
}
```

Edit a Media (Patch)

Allows the partial update of an existing media.

PATCH /media/:media_id

Request

Path Parameters

Name	Description
media_id	The media's ID

Request Headers

Content-Type: application/json

Accept: application/json

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Data Type	Description	Required?
name	String	Name of the Media. Between 0 and 100 alphabetical characters. Must be unique.	Yes
type	String	Type of the Media. Between 0 and 50 alphabetical characters. Must be unique.	Yes
release_date	Date	Date that the Media was first released. Must be a string in format MM-DD-YYYY.	Yes
collection	Dictionary	The Collection that the Media belongs to, if any.	No

Request Body Example

- Partial updates can consist of: name & type, name & release_date, type & release_date, or just one of: name, type, or release_date. Attributes must be valid. Name must be unique.

```
{
  "name": "The Basics of Python: 101",
  "type": "Children's Book"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	<p>If an extraneous attribute (such as ID, or having more than 2 attributes) is included in the request body, 400 status code is returned with the error message: "The request object contains an extraneous attribute."</p> <p>If an attribute is invalid, 400 status code is returned with the error message: "The request object contained an invalid value."</p>
Failure	403 Forbidden	If the name already exists, 403 status code is returned with the error message: "The name is already taken by an existing media."
Failure	404 Not Found	If the collection does not exist, 404 status code is returned with the error message: "No media with this media_id exists".
Failure	415 Unsupported Media Type	If the request header's Content-Type is not set to 'application/json', 415 status code is returned with the error message: "The server only supports JSON format."

Response Examples

- Datastore automatically generates an ID and stores it with the entity being created. Your app must send back in the response body: the ID and self-link to the resource as shown below.
- Partial updates cannot contain all 3 attributes. It will return 400 status code.

Success

```
Status: 200 OK
{
  "name": "The Basics of Python: 101",
  "type": "Children's Book"
}
```

Failure

```
Status: 400 Bad Request
{
  "400 Bad Request": "The request object contained an extraneous attribute."
}
```

```
Status: 403 Forbidden
```

```
{  
  "403 Forbidden": "The name is already taken by an existing media."  
}
```

Status: 404 Not Found

```
{  
  "404 Not Found": "No media with this media_id exists"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "415 Unsupported Media Type": "The server only accepts JSON format."  
}
```

Edit a Media (Put)

Replaces the attributes of an existing media.

PUT /media/:media_id

Request

Path Parameters

Name	Description
media_id	The media's ID

Request Headers

Content-Type: application/json

Accept: application/json

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Data Type	Description	Required?
name	String	Name of the Media. Between 0 and 100 alphabetical characters. Must be unique.	Yes
type	String	Type of the Media. Between 0 and 50 alphabetical characters.	Yes
release_date	Date	Date that the Media was first released. Must be a string in format MM-DD-YYYY.	Yes

Request Body Example

- All 3 attributes are included in the request body when updating an existing boat with PUT.

```
{
  "name": "C++ A Hero's Journey",
  "type": "Audiobook",
  "last_updated": "11-08-2020"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	

Failure	400 Bad Request	<p>If an extraneous attribute (such as ID, or having more than 2 attributes) is included in the request body, 400 status code is returned with the error message: "The request object contains an extraneous attribute."</p> <p>If an attribute is invalid, 400 status code is returned with the error message: "The request object contained an invalid value."</p>
Failure	403 Forbidden	If the name already exists, 403 status code is returned with the error message: "The name is already taken by an existing media."
Failure	404 Not Found	If the collection does not exist, 404 status code is returned with the error message: "No media with this media_id exists".
Failure	415 Unsupported Media Type	If the request header's Content-Type is not set to 'application/json', 415 status code is returned with the error message: "The server only supports JSON format."

Response Examples

- Datastore automatically generates an ID and stores it with the entity being created. Your app must send back in the response body: the ID and self-link to the resource as shown below.
- 303 See Other: The Location header of the response must be set to the self-link of the resource.

Success

```
Status: 303 See Other
{
  "id": 123,
  "name": "C++ A Hero's Journey",
  "type": "Audiobook",
  "last_updated": "11-08-2020",
  "self": "https://portfolio-skinneem.uc.r.appspot.com/media/123"
}
```

Failure

```
Status: 400 Bad Request
{
  "400 Bad Request": "The request object contained an extraneous attribute."
}
```

Status: 403 Forbidden

```
{  
  "403 Forbidden": "The name is already taken by an existing media."  
}
```

Status: 404 Not Found

```
{  
  "404 Not Found": "No media with this media_id exists"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "415 Unsupported Media Type": "The server only accepts JSON format."  
}
```

Add Media to Collection

Allows the assigning of a media to a collection.

```
PUT /collections/:collection_id/media/:media_id
```

Request

Path Parameters

Name	Description
collection_id	The collection's ID
media_id	The media's ID

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	403 Forbidden	If the collection attribute is not null, 403 Error Code is returned.
	404 Not Found	If the collection_id and/or media_id does not exist, 404 Error Code is returned.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found
{
  "Error": "The specified collection and/or media does not exist"
}
```

```
Status: 403 Forbidden
{
  "Error": "The media is already attached to a collection"
}
```

Delete Media from Collection

Allows the removal of a media from a collection.

```
DELETE /collections/:collection_id/media/:media_id
```

Request

Path Parameters

Name	Description
collection_id	The collection's ID
media_id	The media's ID

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	If the collection_id and/or media_id does not exist, 404 Error Code is returned.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found
{
  "Error": "No collection with this collection_id is attached with the media containing this media_id"
}
```

View all Media for a Given Collection

Allows the retrieval of all existing media for a given collection.

GET /collections/:collection_id/media

Request

Path Parameters

Name	Description
collection_id	The collection's ID

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	If no collection with this collection_id exists, 404 Error Code is returned.

Response Examples

- Viewing all media for a given collection does not contain pagination.

Success

```
Status: 200 OK
{
  media: [
    {
      "id": 123,
      "name": "Fellowship of the Ring",
      "type": "Novel",
      "release_date": "07-29-1954",
      "collection": {
        "id": 111,
        "name": "Fantasy Novels",
        "self": "https://portfolio-skinneem.uc.r.appspot.com/collections/111"
      },
      "self": "https://portfolio-skinneem.uc.r.appspot.com/media/123"
    },
  ],
}
```



```
{
  "id": 456,
  "name": "The Two Towers",
  "type": "Novel",
  "release_date": "11-11-1954",
  "collection": {
    "id": 111,
    "name": "Fantasy Novels",
    "self": "https://portfolio-skinneem.uc.r.appspot.com/collections/111"
  },
  "self": "https://portfolio-skinneem.uc.r.appspot.com/media/456"
},
{
  "id": 789,
  "name": "Return of the King",
  "type": "Novel",
  "release_date": "10-20-1955",
  "collection": {
    "id": 111,
    "name": "Fantasy Novels",
    "self": "https://portfolio-skinneem.uc.r.appspot.com/collections/111"
  },
  "self": "https://portfolio-skinneem.uc.r.appspot.com/media/789"
},
],
"self": "https://portfolio-skinneem.uc.r.appspot.com/collections/111"
}
```

Failure

```
Status: 404 Not Found
{
  "Error": "No collection with this collection_id exists"
}
```