

Final Project: API Spec

CS 493: Cloud Application Development

Spring 2023

Oregon State University

Last Update: Jun 11.

Change log.....	1
Data Model	2
Create a Museum.....	4
Get a Museum.....	7
List all Museums.....	9
Edit a Museum (PATCH).....	11
Edit a Museum (PUT)	14
Delete a Museum.....	17
Create an Artwork.....	18
Get an Artwork.....	21
List all Artworks.....	23
Edit an Artwork (PATCH).....	26
Edit an Artwork (PUT)	29
Delete an Artwork.....	32
Assign an Artwork to a Museum.....	34
Remove an Artwork from a Museum	36
Create a User.....	48
List all Users	40
Missing Zip File.....	41

Website

Home Page	https://russekat-project.wl.r.appspot.com
Login	https://russekat-project.wl.r.appspot.com/login

Change log

Version	Change	Date
1.0	Initial version.	June 11, 2023

Data Model

The app stores three kinds of entities in Datastore, Museums, Artworks and Users.

Museums

Property	Data Type	Notes	Required?
id	Integer	The id of the Museum.	No, generated
name	String	Name of the Museum. Must be less than 24 characters and must be unique.	Yes
location	String	Location of the Museum. Must be less than 24 characters. E.g., Portland, OR	Yes
admission	Integer	The cost of admission to enter the museum in USD. Must be greater than or equal to \$0.	Yes
artworks	Array of Objects	Array of artworks that the museum contains. If there are no artworks, then the array is empty. Museums 1:M Artworks	No, automatically [] if no artworks

Artworks (Protected)

Property	Data Type	Notes	Required?
Id	Integer	The id of the Artwork.	No, automatically generated
title	String	Title of the Artwork. Must be less than 50 characters and must be unique.	Yes
artist	String	Name of who created the Artwork. Must be less than 50 characters. E.g., Vincent Van Gogh	Yes
date	Integer	Year the artwork was created. Must be greater than 0. E.g., 1832	Yes
museum	Integer	If the artwork is at a museum, then id of that Museum. Otherwise null.	No, automatically null if no museum
owner	Integer	If the artwork is owned by a user, then the id of that user. Otherwise null.	No, but must access /artworks using a bearer token from login

Users

Property	Data Type	Notes	Required?
Id	Integer	The id of the Artwork.	No, automatically generated
sub	String	Sub of user's account. Must be unique.	Yes
nickname	String	Nickname of the user.	Yes
email	String	Email of the user. This will be used to login to the website.	Yes

artworks	Array of objects	Array of artworks that a user owns. If there are no artworks, then the array is empty. Users 1:M Artworks	No, automatically [] if no artworks
----------	------------------	---	-------------------------------------

Museums and Artworks

- Museums have a 1:M relationship with Artworks

Users

- Users have a 1:M relationship with Artworks
- Sub is the unique identifier of the Users entity
- Auth0 and express-jwt is used to map the jwt to the identifier of a user. It specifically checks the sub matches

Create a Museum

Allows you to create a new Museum.

POST /museums

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the Museum. Must be less than 24 characters and must be unique.	Yes
location	Location of the Museum. Must be less than 24 characters. E.g., Portland, OR.	Yes
admission	The cost of admission to enter the museum in USD. Must be greater than or equal to \$0.	Yes

Request Body Example

```
{
  "name": "Louvre Museum",
  "location": "Paris, France",
  "admission": 17
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, has extraneous attributes or doesn't meet the attribute requirements (name and location less than 24 characters and admission greater than or equal to 0), the Museum must not be updated, and 400 status code must be returned.
Failure	415 Unsupported Media Type	Server only accepts application/json data
Failure	406 Not Acceptable	Server only returns application/json data
Failure	403 Forbidden	Museum name must be unique

Success

Status: 201 Created

```
{
  "id": 123,
  "name": "Louvre Museum",
  "location": "Paris, France",
  "admission": 17,
  "artworks": [],
  "self": "http://url.com/museums/123"
}
```

Failure

Status: 400 Bad Request

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

or

```
{
  "Error": "Name and Location must be a string and should be less than 24
characters. Admission must be an integer and should be greater than or equal to 0."
}
```

or

```
{
  "Error": "The request object contains extraneous attributes"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "Server only accepts application/json data"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server only returns application/json data"
}
```

Status: 403 Forbidden

```
{  
  "Error": "This name is already taken!"  
}
```

Get a Museum

Allows you to get an existing Museum

GET /museums/:museum_id

Request

Path Parameters

Name	Description
<code>museum_id</code>	ID of the Museum

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No Museum with this Museum_id exists
Failure	406 Not Acceptable	Server only returns application/json data
Failure	500 Internal Server Error	Content type got messed up

Response Examples

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Louvre Museum",
  "location": "Paris, France",
  "admission": 17,
  "artworks": [],
  "self": "http://url.com/museums/123"
}
```

Failure

Status: 404 Not Found

```
{
  "Error": "No museum with this museum_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable"
}
```

Status: 500 Internal Server Error

```
{  
"Error": "Content type got messed up!"  
}
```


List all Museums

List all the Museums.

GET /museums

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Results return in a pagination of 5.
Failure	406 Not Acceptable	Server only returns application/json data
Failure	500 Internal Server Error	Content type got messed up

Response Examples

Success

Status: 200 OK

```
{
  "museums": [
    {
      "id": 123,
      "name": "Louvre Museum",
      "location": "Paris, France",
      "admission": 17,
      "artworks": [],
      "self": "http://url.com/museums/123"
    },
    {
      "id": 456,
      "name": "The Getty",
      "location": "Los Angeles, California",
      "admission": 0,
      "artworks": [],
      "self": "http://url.com/museums/456"
    },
    {
      "id": 789,
      "name": "Reina Sofia",
      "location": "Madrid, Spain",
      "admission": 12,

```

```
"artworks": [],  
"self": "http://url.com/museums/789"  
},  
{  
  "id": 987,  
  "name": "National Art Center",  
  "location": "Tokyo, Japan",  
  "admission": 0,  
  "artworks": [],  
  "self": "http://url.com/museums/987"  
},  
{  
  "id": 654,  
  "name": "Vatican Museums",  
  "location": "Vatican City, Italy",  
  "admission": 17,  
  "artworks": [],  
  "self": "http://url.com/museums/654"  
}  
],  
"total": 8,  
"next": "https://url.com/museums?cursor=1234"  
}
```

Failure

Status: 406 Not Acceptable

```
{  
  "Error": "Not Acceptable"  
}
```

Status: 500 Internal Server Error

```
{  
  "Error": "Content type got messed up!"  
}
```

Edit a Museum

Allows you to edit a Museum.

PUT /museums/:museum_id

Request

Path Parameters

Name	Description
<code>museum_id</code>	ID of the Museum

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
<code>name</code>	Name of the Museum. Must be less than 24 characters and must be unique	Yes
<code>location</code>	Location of the Museum. Must be less than 24 characters. E.g., Portland, OR	Yes
<code>admission</code>	The cost of admission to enter the museum in USD. Must be greater than or equal to \$0.	Yes

Request Body Example

```
{
  "name": "Louvre Museum",
  "location": "Paris, France",
  "admission": 20
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 3 attributes, is trying to update the id , has extraneous attributes or doesn't meet the attribute requirements (name and location less than 24 characters and admission greater than or equal to 0), the museum must not be edited and 400 status code must be returned.
Failure	404 Not Found	No Museum with this <code>Museum_id</code> exists
Failure	415 Unsupported Media Type	Server only accepts application/json data
Failure	406 Not Acceptable	Server only returns application/json data

Failure	403 Forbidden	Museum name must be unique
---------	---------------	----------------------------

Response Examples

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Louvre Museum",
  "location": "Paris, France",
  "admission": 20,
  "self": "http://url.com/museums/123"
}
```

Failure

Status: 400 Bad Request

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

or

```
{
  "Error": "Name and Location must be a string and should be less than 24
characters. Admission must be an integer and should be greater than or equal to 0."
}
```

or

```
{
  "Error": "The request object contains extraneous attributes"
}
```

or

```
{
  "Error": "Cannot update the value of id"
}
```

Status: 404 Not Found

```
{
  "Error": "No Museum with this Museum_id exists"
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server only accepts application/json data"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Server only returns application/json data"  
}
```

Status: 403 Forbidden

```
{  
  "Error": "This name is already taken!"  
}
```

Edit a Museum

Allows you to edit any attribute(s) of a museum.

PATCH /museums/:museum_id

Request

Path Parameters

Name	Description
Museum_id	ID of the Museum

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the Museum. Must be less than 24 characters and must be unique	No
location	Location of the Museum. Must be less than 24 characters. E.g., Portland, OR	No
admission	The cost of admission to enter the museum in USD. Must be greater than or equal to \$0.	No

Request Body Example

```
{
  "admission": 18
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is trying to update the id , has extraneous attributes or doesn't meet the attribute requirements (name and location less than 24 characters and admission greater than or equal to 0), the Museum must not be edited, and 400 status code must be returned.
Failure	404 Not Found	No Museum with this Museum_id exists
Failure	415 Unsupported Media Type	Server only accepts application/json data
Failure	406 Not Acceptable	Server only returns application/json data
Failure	403 Forbidden	Museum name must be unique

Response Examples

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Louvre Museum",
  "location": "Paris, France",
  "admission": 18,
  "self": "http://url.com/museums/123"
}
```

Failure

Status: 400 Bad Request

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

or

```
{
  "Error": "Name and Location must be a string and should be less than 24
characters. Admission must be an integer and should be greater than or equal to 0."
}
```

or

```
{
  "Error": "The request object contains extraneous attributes"
}
```

or

```
{
  "Error": "Cannot update the value of id"
}
```

Status: 404 Not Found

```
{
  "Error": "No Museum with this Museum_id exists"
}
```

Status: 415 Unsupported Media Type

Server only accepts application/json data

Status: 406 Not Acceptable

Server only returns application/json data

Status: 403 Forbidden

```
{  
  "Error": "This name is already taken!"  
}
```


Delete a Museum

Allows you to delete a Museum. Note that if there are artworks in the museum, deleting the museum removes the artwork from the museum.

DELETE /museums/:museum_id

Request

Path Parameters

Name	Description
<code>museum_id</code>	ID of the Museum

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No Museum with this museum_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
  "Error": "No museum with this museum_id exists"
}
```

Create an Artwork

Protected

Allows you to create a new Artwork.

POST /artworks

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Authorization

Bearer Token Required

Request JSON Attributes

Name	Description	Required?
title	Title of the Artwork. Must be less than 50 characters and must be unique.	Yes
artist	Name of who created the Artwork. Must be less than 50 characters. E.g., Vincent Van Gogh	Yes
date	Year the artwork was created. Must be greater than 0. E.g., 1832	Yes

Request Body Example

```
{
  "title": "Rouen Cathedral ",
  "artist": "Claude",
  "date": 1894
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, has extraneous attributes or doesn't meet the attribute requirements (title and artist less than 50 characters and date greater than 0), and 400 status code must be returned.
Failure	415 Unsupported Media Type	Server only accepts application/json data

Failure	406 Not Acceptable	Server only returns application/json data
Failure	403 Forbidden	Artwork name must be unique
Failure	401 Unauthorized	Invalid bearer token

Success

Status: 201 Created

```
{
  "id": 123,
  "title": "Rouen Cathedral",
  "artist": "Claude",
  "date": 1894,
  "museum": null,
  "owner": "auth0|abcd",
  "self": "http://url.com/artworks/123"
}
```

Failure

Status: 400 Bad Request

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

or

```
{
  "Error": "Title and Artist must be a string and should be less than 50 characters.
  Date must be an integer and should be greater than 0."
}
```

or

```
{
  "Error": "The request object contains extraneous attributes."
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "Server only accepts application/json data"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server only returns application/json data"
}
```

Status: 403 Forbidden

```
{  
"Error": "This title is already taken!"  
}
```

Status: 401 Unauthorized

Unauthorized

Get an Artwork

Protected

Allows you to get an existing Artwork.

GET /artworks/:artwork_id

Request

Path Parameters

Name	Description
artwork_id	ID of the Artwork

Request Body

None

Response

Response Body Format

JSON

Authorization

Bearer Token Required

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No Artwork with this Artwork_id exists
Failure	406 Not Acceptable	Server only returns application/json data
Failure	500 Internal Server Error	Content type got messed up
Failure	401 Unauthorized	Invalid Bearer Token
Failure	403 Forbidden	Bearer Token doesn't belong to owner of artwork

Response Examples

Success

Status: 200 OK
<pre>{ "id": 123, "title": "Rouen Cathedral", "artist": "Claude", "date": 1894 "museum": null, "owner": "auth0 abcd", "self": "http://url.com/artworks/123" }</pre>

Failure

Status: 404 Not Found
<pre>{ "Error": "No artwork with this artwork_id exists" }</pre>

```
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Not Acceptable"  
}
```

Status: 500 Internal Server Error

```
{  
  "Error": "Content type got messed up!"  
}
```

Status: 401 Unauthorized

Unauthorized

Status: 403 Forbidden

```
{  
  "Error": "Forbidden"  
}
```

List all Artworks

Protected

List all the Artworks.

GET /artworks

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Authorization

Bearer Token Required

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Results return in a pagination of 5. Returns only the artworks associated with the owner from the bearer token
Failure	406 Not Acceptable	Server only returns application/json data
Failure	500 Internal Server Error	Content type got messed up
Failure	401 Unauthorized	Invalid bearer token

Response Examples

Success

Status: 200 OK

```
{
  "artworks": [
    {
      "id": 123,
      "title": "Rouen Cathedral",
      "artist": "Claude",
      "date": 1894,
      "museum": null,
      "owner": "auth0|abcd",
      "self": "http://url.com/artworks/123"
    },
    {
      "id": 456,
      "title": "Saturn Devouring His Son",
      "artist": "Francisco Goya",

```

```
"date": 1819,
"museum": null,
"owner": "auth0|abcd",
"self": "http://url.com/artworks/456"
},
{
  "id": 789,
  "title": "The Garden of Earthly Delights",
  "artist": "Hieronymus Bosch",
  "date": 1503,
  "museum": null,
  "owner": "auth0|abcd",
  "self": "http://url.com/artworks/789"
},
{
  "id": 987,
  "title": "Nighthawks",
  "artist": "Edward Hopper",
  "date": 1942,
  "museum": null,
  "owner": "auth0|abcd",
  "self": "http://url.com/artworks/987"
},
{
  "id": 654,
  "title": "Judith Beheading Holofernes",
  "artist": "Caravaggio",
  "date": 1598,
  "museum": null,
  "owner": "auth0|abcd",
  "self": "http://url.com/artworks/654"
}
],
"total": 6,
"next": "https://url.com/artworks?cursor=1234"
}
```

Failure

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable"
}
```

Status: 500 Internal Server Error

```
{
  "Error": "Content type got messed up!"
}
```


}

Status: 401 Unauthorized

Unauthorized

Edit an Artwork

Protected

Allows you to edit an Artwork.

PUT /artworks/:artwork_id

Request

Path Parameters

Name	Description
artwork_id	ID of the artwork

Request Body

Required

Request Body Format

JSON

Authorization

Bearer Token Required

Request JSON Attributes

Name	Description	Required?
title	Title of the Artwork. Must be less than 50 characters and must be unique.	Yes
artist	Name of who created the Artwork. Must be less than 50 characters. E.g., Vincent Van Gogh	Yes
date	Year the artwork was created. Must be greater than 0. E.g., 1832	Yes

Request Body Example

```
{
  "title": "Rouen Cathedral, Full Sunlight",
  "artist": "Claude",
  "date": 1894
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 3 attributes, is trying to update the id , has extraneous attributes or doesn't meet the attribute requirements (title and artist are less than 50 characters and admission is greater than or equal to 0), the artwork must not be edited and 400 status code must be returned.

Failure	404 Not Found	No artwork with this artwork_id exists
Failure	415 Unsupported Media Type	Server only accepts application/json data
Failure	406 Not Acceptable	Server only returns application/json data
Failure	403 Forbidden	artwork name must be unique or artwork does not belong to the user
Failure	401 Unauthorized	Invalid Bearer Token

Response Examples

Success

Status: 200 OK

```
{
  "id": 123,
  "title": "Rouen Cathedral, Full Sunlight",
  "artist": "Claude",
  "date": 1894,
  "museum": null,
  "owner": "auth0|abcd",
  "self": "http://url.com/artworks/123"
}
```

Failure

Status: 400 Bad Request

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

or

{
  "Error": "Title and Artist must be a string and should be less than 50 characters.
Date must be an integer and should be greater than 0."
}

or

{
  "Error": "The request object contains extraneous attributes"
}

or

{
  "Error": "Cannot update the value of id"
}
```

Status: 404 Not Found

```
{  
"Error": "No Artwork with this Artwork_id exists"  
}
```

Status: 415 Unsupported Media Type

```
{  
"Error": "Server only accepts application/json data"  
}
```

Status: 406 Not Acceptable

```
{  
"Error": "Server only returns application/json data"  
}
```

Status: 403 Forbidden

```
{  
"Error": "This name is already taken!"  
}
```

or

```
{  
"Error": "Forbidden"  
}
```

Status: 401 Unauthorized

Unauthorized

Edit an Artwork

Protected

Allows you to edit any attribute(s) of an artwork.

PATCH /artworks/:artwork_id

Request

Path Parameters

Name	Description
artwork_id	ID of the artwork

Request Body

Required

Request Body Format

JSON

Authorization

Bearer Token Required

Request JSON Attributes

Name	Description	Required?
title	Title of the Artwork. Must be less than 50 characters and must be unique.	No
artist	Name of who created the Artwork. Must be less than 50 characters. E.g., Vincent Van Gogh	No
date	Year the artwork was created. Must be greater than 0. E.g., 1832	No

Request Body Example

```
{
  "artist": "Claude Monet"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is trying to update the id , has extraneous attributes or doesn't meet the attribute requirements (title and artist are less than 50 characters and admission is greater than or equal to 0), the artwork must not be edited and 400 status code must be returned.
Failure	404 Not Found	No artwork with this artwork_id exists
Failure	415 Unsupported Media Type	Server only accepts application/json data

Failure	406 Not Acceptable	Server only returns application/json data
Failure	403 Forbidden	artwork name must be unique or this artwork doesn't belong to this user
Failure	401 Unauthorized	Invalid Bearer Token

Response Examples

Success

Status: 200 OK

```
{
  "id": 123,
  "title": "Rouen Cathedral, Full Sunlight",
  "artist": "Claude Monet",
  "date": 1894,
  "museum": null,
  "owner": "auth0|abcd",
  "self": "http://url.com/artworks/123"
}
```

Failure

Status: 400 Bad Request

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

or

```
{
  "Error": "Title and Artist must be a string and should be less than 50 characters.
Date must be an integer and should be greater than 0."
}
```

or

```
{
  "Error": "The request object contains extraneous attributes"
}
```

or

```
{
  "Error": "Cannot update the value of id"
}
```

Status: 404 Not Found

```
{
  "Error": "No artwork with this artwork_id exists"
}
```

```
}
```

Status: 415 Unsupported Media Type

```
{  
"Error": "Server only accepts application/json data"  
}
```

Status: 406 Not Acceptable

```
{  
"Error": "Server only returns application/json data"  
}
```

Status: 403 Forbidden

```
{  
"Error": "This name is already taken!"  
}
```

or

```
{  
"Error": "Forbidden"  
}
```

Status: 401 Unauthorized

Unauthorized

Delete an Artwork

Protected

Allows you to delete an Artwork. If the Artwork being deleted has a Museum, the Artwork is no longer at a museum.

DELETE /artworks/:artwork_id

Request

Path Parameters

Name	Description
artwork_id	ID of the Artwork

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Authorization

Bearer Token Required

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No Artwork with this artwork_id exists
Failure	403 Forbidden	Artwork doesn't belong to this user
Failure	401 Unauthorized	Invalid bearer token

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
  "Error": "No Artwork with this artwork_id exists"
}
```

Status: 403 Forbidden

```
{
  "Error": "Forbidden"
}
```


Status: 401 Unauthorized

Unauthorized

Assign an Artwork to a Museum

Protected

Assign an artwork to a museum.

PUT /museums/:museum_id/artworks/:artwork_id

Request

Path Parameters

Name	Description
artwork_id	ID of the Artwork
museum_id	ID of the Museum

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Authorization

Bearer Token Required

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a Museum exists with this Museum_id, a Artwork exists with this Artwork_id and the Artwork is not at a museum.
Failure	403 Forbidden	The Artwork is not empty because there is already a Museum holding this Artwork. Or this artwork doesn't belong to this user.
Failure	404 Not Found	The specified Museum and/or Artwork does not exist
Failure	401 Unauthorized	Invalid Bearer Token

Response Examples

Success

Status: 204 No Content

Failure

Status: 403 Forbidden

```
{
  "Error": "The Artwork is not empty"
}
```

or

```
{  
  "Error": "Forbidden"  
}
```

Status: 404 Not Found

```
{  
  "Error": "The specified Museum and/or Artwork does not exist"  
}
```

Status: 401 Unauthorized

Unauthorized

Comment

- A museum can have multiple artworks, but an artwork can only be assigned to one museum.

Remove an Artwork from a Museum

Protected

Remove an Artwork from a Museum. Artwork is no longer a part of a museum collection.

DELETE /museums/:museum_id/artworks/:artwork_id

Request

Path Parameters

Name	Description
artwork_id	ID of the Artwork
museum_id	ID of the Museum

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Authorization

Bearer Token Required

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a Museum exists with this Museum_id, an Artwork exists with this Artwork_id and this Artwork is at this Museum.
Failure	404 Not Found	No Artwork with this Artwork_id is at the Museum with this Museum_id. This could be because no Museum with this Museum_id exists, or because no Artwork with Artwork_id exists, or even if both Museum_id and Artwork_id are valid, the Artwork with this Artwork_id is not at this Museum with this Museum_id
Failure	403 Forbidden	Artwork doesn't belong to this user
Failure	401 Unauthorized	Invalid bearer token

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

<pre>{ "Error": " No Museum with this Museum_id is at the Artwork with this Artwork_id" }</pre>

Status: 403 Forbidden

```
{  
  "Error": "Forbidden"  
}
```

Status: 401 Unauthorized

Unauthorized

Create a User

Allows you to create a new User. Done through the “create a new user” page.

POST /users

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
sub	Sub of users account	Yes
nickname	Nickname of user	Yes
email	Email of the user. This will be used to login to the website.	Yes

Request Body Example

```
{
  "sub": "auth0|abc",
  "nickname": "jane",
  "email": "jane@test.com"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing the number attribute, the Artwork must not be created, and 400 status code must be returned.

Success

Status: 201 Created

```
{
  "id": 123,
  "sub": "auth0|abc",
  "nickname": "jane",
  "email": "jane@test.com",
  "artworks": [],
  "self": "http://url.com/users/123"
}
```

Failure

Status: 400 Bad Request

```
{  
  "Error": "The request object is missing the required number"  
}
```

List all Users

List all the Artworks.

GET /users

Request

Path Parameters

None

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,
  "sub": "auth0|abc",
  "nickname": "jane",
  "email": "jane@test.com",
  "artworks": []
},
{
  "id": 456,
  "sub": "auth0|def",
  "nickname": "john",
  "email": "john@test.com",
  "artworks": []
}
]
```


Missing ZIP File

Missing zip file with the source code.

I intended to meet the requirement by downloading all of the source code from google cloud console.

However, when I reloaded my console, I could no longer access it as I reached a cloud shell limit. After researching online, I found out the only way to increase the limit is by submitting a request that will take up to 3-4 business days. There was also no other way that I found, such as google cloud SDK shell or VScode extension, where I can access my files as it was only stored on the google cloud console.

I found that the quota is 50 hours per week which I believe I reached while looking over my code while creating the API document and Postman tests since I had to compare the response codes.

Although I cannot provide the source code, here are the files that my project contained in the folder: final-project/nodejs-sample-docs/datastore

app.js

- Which included the code for logging in and logging out and was adapted from
Title: Auth0-express-webapp-sample
Author: Auth0-samples
Source: <https://github.com/auth0-samples/auth0-express-webapp-sample/tree/master/01-Login>

datastore.js

index.js

museums.js

- Routes for /museums

artworks.js

- Routes for /artworks

users.js

- Routes for /users and /callback

views/welcome.hbs

package.json

- I believe the dependencies are the same from the example code in module: authentication in node
 - @google-cloud/datastore
 - Express
 - Express-jwt
 - Google-gax
 - Json-to-html
 - Jwks-rsa
- Though instead of request I used http and I also installed express-handlebars to connect the views