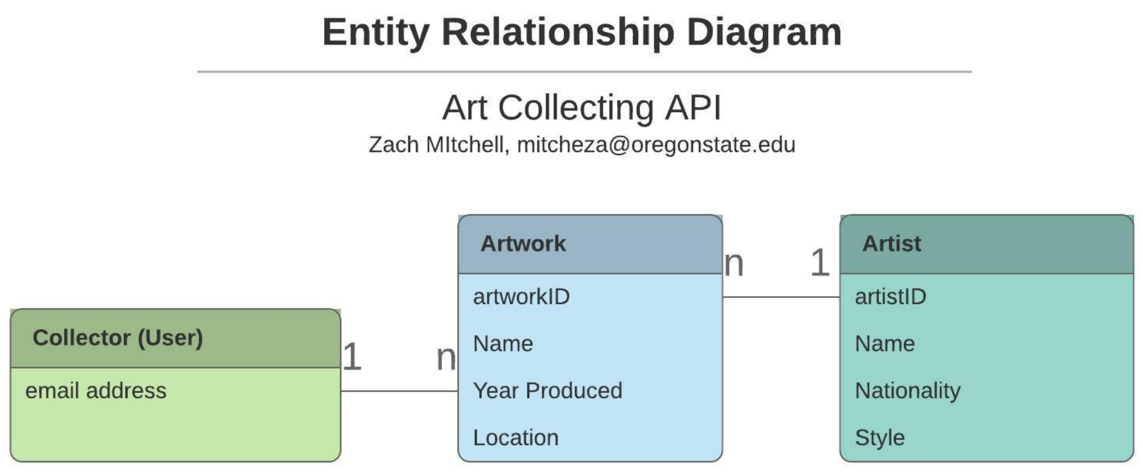
Zach Mitchell, mitcheza@oregonstate.edu
CS493 - Fall 2018, 12/2/2018
Final Project - Art Collecting API

## Project Description

Base URL: https://mitcheza-api-2.appspot.com
Video of Testing: https://www.youtube.com/watch?v=Gi9NsUkpjS8

My database contains 3 entities: **art collectors (users)**, **works of art**, and **artists**. User information is not saved in my database, but is instead outsourced to a 3rd party application, Auth0. Artworks and artists are saved in the database. An entity-relationship diagram of the database, with all corresponding entity properties looks like so:



## Example JSON Representation of Entities

Artwork:
```
{
        "owner": "museo@seattle.com",
        "name": "La Vie",
        "year": 1903,
        "location": "Cleveland",
        "artist": {
           "id": "5129017531301888",
           "self": "https://mitcheza-api-2.appspot.com/artists/5129017531301888",
           "name": "Pablo Picasso"
        },
        "id": "5633369601736704",
        "self": "https://mitcheza-api-2.appspot.com/works/5633369601736704"
}
```

Artist:

```
{
        "name": "Paul Gauguin",
        "nationality": "French",
        "style": "Post-Impressionism",
        "works": [
           {
              "id": "5635321228165120",
              "self": "https://mitcheza-api-2.appspot.com/works/5635321228165120",
              "name": "Woman with a Flower"
           },
           {
              "self": "https://mitcheza-api-2.appspot.com/works/5762273213677568",
              "name": "Where Do We Come From? What Are We? Where Are We Going?",
              "id": "5762273213677568"
           }
        ],
        "id": "5764984579555328",
        "self": "https://mitcheza-api-2.appspot.com/artists/5764984579555328"
 }
```

# API Documentation

Base URL: https://mitcheza-api-2.appspot.com

## 1. Collectors / Owners (Users)

To make *any* call to the database using the API, a user must first be registered and then login and use their Bearer token for all subsequent interactions.

### A. Create a New User:

Adds a new user to our Auth0 application for future authenication/authorization requests.

**Call**: *POST /newuser*

**Parameters:**
"username": - String, email address
"password": - String, password to login to authenticate for API usage

**Response:**
Status: 201 Created

"<username> account created!"

### B. User Login: Authenticates user and gives them a Bearer token which they can then use to authorize future calls to the API.

**Call**: *POST /login*

**Parameters:**
"username": - String, email address
"password": - String, password to login to authenticate for API usage

**Response:**
Status: 200 OK

```
{
   "access_token": "eyJ0...DsQ",
   "id_token": "eyJ0e….eEig",
   "scope": "openid profile email address phone read:current_user update:current_user_metadata
         delete:current_user_metadata create:current_user_metadata
         create:current_user_device_credentials delete:current_user_device_credentials
         update:current_user_identities",
   "expires_in": 86400,
   "token_type": "Bearer"
}
```

## 2. Works of Art

Works of art can be **created** and **viewed** (viewed in a collection or individually) by **any user** with a bearer token. However, only the **owner** of a work can **edit**, **delete**, **assign/remove artists** to a given work of art.

All calls must explicitly accept application/json and have a bearer token.

### A. Add a work of art: Adds a new piece of art to the database.

**Call**: *POST /works*

**Parameters:**
"name": - String, name of the artwork
"year": - Number, year the work was finished
"location" - String, location of the work (city, country, etc)

**Response:**
Status: 201 Created

```
{"id" : "5680680545550336" }
```

**B. View collection of art works:** View the entire collection of artworks and the total count of artworks in the collection.

**Call**: *GET /works*

**Parameters:**

**Response:**
Status: 200 Ok

```
{
   “works”: [
      {
         "owner": "museo@seattle.com",
         "name": "La Vie",
         "year": 1903,
         "location": "Cleveland",
         "artist": {
            "self": "https://mitcheza-api-2.appspot.com/artists/5129017531301888",
            "name": "Pablo Picasso",
            "id": "5129017531301888"
         },
         "id": "5633369601736704",
         "self": "https://mitcheza-api-2.appspot.com/works/5633369601736704"
      },
         … ,
      {
         "owner": "museo@seattle.com",
         "name": "Where Do We Come From? What Are We? Where Are We Going?",
         "year": 1897,
         "location": "Boston",
         "artist": {
            "self": "https://mitcheza-api-2.appspot.com/artists/5764984579555328",
            "name": "Paul Gauguin",
            "id": "5764984579555328"
         },
         "id": "5762273213677568",
         "self": "https://mitcheza-api-2.appspot.com/works/5762273213677568"
      }
   ],
   "total": 4
}
```

## C. View a single artwork: View a single artwork.

**Call**: *GET /works/:workID*

**Parameters:**

**Response:**
Status: 200 Ok

```
{
   "location": "Cleveland",
   "artist": {
      "id": "5129017531301888",
      "self": "https://mitcheza-api-2.appspot.com/artists/5129017531301888",
      "name": "Pablo Picasso"
   },
   "owner": "museo@seattle.com",
   "name": "La Vie",
   "year": 1903,
   "id": "5633369601736704",
   "self": "https://mitcheza-api-2.appspot.com/works/5633369601736704"
}
```

## D. Edit a single artwork (Owner only): Adjust the name, location, year of production for a work of art.

**Call**: *PUT /works/:workID*

**Parameters:**
“name”: - String, name of the artwork
“year”: - Number, year the work was finished
“location” - String, location of the work (city, country, etc)

**Response:**
Status: 200 Ok

## E. Delete a single artwork (Owner only): Remove work of art from database.

**Call**: *DELETE /works/:workID*

**Parameters:**

**Response:**
Status: 204 No Content

## 3. Artists

All CRUD calls on artists can be performed by anyone with a bearer token, irrespective of who created the artist.

### A. Add an artist: Adds a new artist to the database.

**Call**: *POST /artists*

**Parameters:**
"name": - String, name of the artist
"nationality": - String, country they were born
"style" - String, major art movement they are thought to inhabit

**Response:**
Status: 201 Created

```
{"id" : "5691967484723200" }
```

### B. View collection of artists: View the entire collection of artists and the total count of artists in the collection.

**Call**: *GET /artists*

**Parameters:**

**Response:**
Status: 200 Ok

```
{
   "artists": [
      {
         "style": "Pop-Art",
         "works": [],
         "name": "Andy Warhol",
         "nationality": "American",
         "id": "5127195055882240",
         "self": "https://mitcheza-api-2.appspot.com/artists/5127195055882240"
      },
         …,
         ],
   "total" : 8,
   "next": "https://mitcheza-api-2.appspot.com/artists?cursor=Ci0SJ2o...BgAIAA="
}
```

## C. View a single artist: View an artist and a collection of their attributed works.

**Call**: *GET /artists/:artistID*

**Parameters:**

**Response:**
Status: 200 Ok

```
{
        "nationality": "French",
        "style": "Post-Impressionism",
        "works": [
           {
              "self": "https://mitcheza-api-2.appspot.com/works/5635321228165120",
              "name": "Woman with a Flower",
              "id": "5635321228165120"
           },
           {
              "self": "https://mitcheza-api-2.appspot.com/works/5762273213677568",
              "name": "Where Do We Come From? What Are We? Where Are We Going?",
              "id": "5762273213677568"
           }
        ],
        "name": "Paul Gauguin",
        "id": "5764984579555328",
        "self": "https://mitcheza-api-2.appspot.com/artists/5764984579555328"
}
```

## D. Edit a single artist: Adjust an artist's attributes.

**Call**: *PUT /artists/:artistID*

**Parameters:**
"name": - String, name of the artist
"nationality": - String, country they were born
"style" - String, major art movement they are thought to inhabit

**Response:**
Status: 200 Ok

### E. Delete a single artist:

Remove artist from database and clear any corresponding works of art of this artist's name.

**Call**: *DELETE /artists/:artistID*

**Parameters:**

**Response:**
Status: 204 No Content

## 4. Relationships between artwork and artists

To **assign** or **remove** an artist to a work of art, the user must also be the **owner of the work of art** on top of having a bearer token.

### A. Add work to an artist (Owner of work only):

Adds artist to the "artist" property of a work of art and then adds the work of art to the artist's "works" array.

**Call**: *PUT /artists/:artist_id/works/:work_id*

**Parameters:**

**Response:**
Status: 200 Ok

### B. Remove work from an artist (Owner of work only):

Removes artist from the "artist" property of a work of art and then removes the work of art from the artist's "works" array.

**Call**: *DELETE /artists/:artist_id/works/:work_id*

**Parameters:**

**Response:**
Status: 200 Ok