

API Documentation

Base URL:

<https://piccionj-node-app.uc.r.appspot.com/>

This is the app url as well as the endpoint which can take you to the login and create a userpage. The postman test suite provides a way to do this so you don't have to.

DATA MODEL

The application appears to be a web application that manages portfolios and pieces of artwork. It uses Express.js as the web framework and integrates with the Google Cloud Datastore for data storage. The application allows users to create portfolios and add pieces to their portfolios.

Entities:

1. User:

- Attributes:
 - id: unique identifier for the user
 - email: email address of the user

2. Portfolio:

- Attributes:
 - id: unique identifier for the portfolio
 - name: name of the portfolio
 - type: type or category of the portfolio
 - public: indicates whether the portfolio is public or private
 - owner: foreign key reference to the user who owns the portfolio

3. Piece:

- Attributes:
 - id: unique identifier for the piece
 - name: name of the piece
 - description: description or details about the piece
 - price: price of the piece
 - portfolio_id: foreign key reference to the portfolio the piece belongs to
 - owner: foreign key reference to the user who owns the piece
 - image_url: URL of the image representing the piece
 - self: URL of the piece resource

Relationships:

- Each portfolio has a one-to-many relationship with pieces. A portfolio can have multiple pieces, but a piece can only belong to one portfolio.
- Each piece belongs to one portfolio and one user. A user can own multiple pieces and a piece can only be owned by one user.
- a user can have multiple portfolios but a portfolio can only have one user

The data model represents the entities and their relationships in the application. The User entity represents the users of the application, while the Portfolio entity represents the portfolios created by users. The Piece entity represents the individual pieces of artwork that users can

add to their portfolios. The relationships between entities are defined using foreign key references.

Authentication

The API endpoints require authentication using JWT (JSON Web Tokens). The JWT token should be included in the request cookies using the key `jwt`. Meaning it is stored into the cookie header for design purposes.

API Endpoints

Home Page

- URL: `/`
- Method: `GET`
- Description: Renders the home page.
- Response Body:
 - `title`: The title of the page.

User Page

- URL: `/userpage`
- Method: `GET`
- Description: Renders the user page.
- Authorization: Requires a valid JWT token.
- Response Body:
 - `title`: The title of the page.
 - `portfolios`: Array of user's portfolios.
 - `pieces`: Array of user's pieces.

Get Pieces

- URL: `/piece`
- Method: `GET`
- Description: Retrieves a list of pieces.
- Response Body:
 - `title`: The title of the page.
 - `pieces`: Array of pieces.
 - `next`: URL for the next page of results (optional).

Get Piece by ID

- URL: `/piece/:id`
- Method: `GET`
- Description: Retrieves a specific piece by ID.
- Authorization: Requires a valid JWT token.
- Response Body:
 - `title`: The title of the page.
 - `piece`: The requested piece.

Create Piece

- URL: `/piece`
- Method: `POST`
- Description: Creates a new piece.
- Authorization: Requires a valid JWT token.
- Request Body:
 - `name`: The name of the piece.
 - `description`: The description of the piece.
 - `price`: The price of the piece.
 - `portfolio_id`: The ID of the portfolio the piece belongs to.
 - `owner`: The owner of the piece.
 - `image_url`: The URL of the piece's image.
- Response Body:
 - `title`: The title of the page.
 - `pieces`: Array of pieces.

Delete Piece

- URL: `/piece/:id`
- Method: `DELETE`
- Description: Deletes a specific piece by ID.
- Authorization: Requires a valid JWT token.
- Response Body:
 - `title`: The title of the page.

Delete Piece from Portfolio

- URL: `/piece/:id/portfolio`
- Method: `DELETE`
- Description: Removes a piece from its associated portfolio.
- Authorization: Requires a valid JWT token.
- Response Body:
 - `title`: The title of the page.
 - `piece`: The updated piece.

Update Piece

- URL: `/piece/:id`

- Method: `PUT`
- Description: Updates a specific piece by ID.
- Authorization: Requires a valid JWT token.
- Request Body:
 - `name`: The updated name of the piece.
 - `description`: The updated description of the piece.
 - `price`: The updated price of the piece.
 - `portfolio_id`: The updated ID of the portfolio the piece belongs to.
 - `image_url`: The updated URL of the piece's image.
- Response Body:
 - `title`: The title of the page.

Partially Update Piece

- URL: `/piece/:id`
- Method: `PATCH`
- Description: Partially updates a specific piece by ID.
- Authorization: Requires a valid JWT token.
- Request Body (optional):
 - `name`: The updated name of the piece.
 - `description`: The updated description of the piece.
 - `price`: The updated price of the piece.
 - `portfolio_id`: The updated ID of the portfolio the piece belongs to.
 - `image_url`: The updated URL of the piece's image.
- Response Body:
 - `title`: The title of the page.

Get Portfolios

- URL: `/portfolio`
- Method: `GET`
- Description: Retrieves a list of portfolios.
- Authorization: Requires a valid JWT token.
- Response Body:
 - `title`: The title of the page.
 - `portfolios`: Array of portfolios.
 - `next`: URL for the next page of results (optional).

Get Portfolio by ID

- URL: `/portfolio/:id`
- Method: `GET`
- Description: Retrieves a specific portfolio by ID.
- Authorization: Requires a valid JWT token.
- Response Body:
 - `title`: The title of the page.
 - `portfolio`: The requested portfolio.
 - `pieces`: Array of pieces within the portfolio.

Create Portfolio

- URL: `/portfolio`
- Method: `POST`
- Description: Creates a new portfolio.
- Authorization: Requires a valid JWT token.
- Request Body:
 - `name`: The name of the portfolio.
 - `description`: The description of the portfolio.
 - `owner`: The owner of the portfolio.
- Response Body:
 - `title`: The title of the page.
 - `portfolios`: Array of portfolios.

Delete Portfolio

- URL: `/portfolio/:id`
- Method: `DELETE`
- Description: Deletes a specific portfolio by ID.
- Authorization: Requires a valid JWT token.
- Response Body:
 - `title`: The title of the page.

Update Portfolio

- URL: `/portfolio/:id`
- Method: `PUT`
- Description: Updates a specific portfolio by ID.
- Authorization: Requires a valid JWT token.
- Request Body:
 - `name`: The updated name of the portfolio.

- `description`: The updated description of the portfolio.
- Response Body:
 - `title`: The title of the page.

Partially Update Portfolio

- URL: `/portfolio/:id`
- Method: `PATCH`
- Description: Partially updates a specific portfolio by ID.
- Authorization: Requires a valid JWT token.
- Request Body (optional):
 - `name`: The updated name of the portfolio.
 - `description`: The updated description of the portfolio.
- Response Body:
 - `title`: The title of the page.

.