Emily Addiego
6/6/21
CS 493 Spring 2021

Final Project

1. Overview

A. **Description:** This API is based around users, animals and zoos. An animal can be "sponsored" by a user, wherein it is protected by the user's jwt. Unsponsored animals can be accessed without authentication. Zoos and animals have a one-to-many relationship, where many animals can live at one zoo. Users and animals have a one-to-many relationship, where a user can sponsor many animals. The entities and their attributes are as follows:

user	animal	zoo
{_id: <ObjectID> (required), name: string (required), email: string (required), auth0_id: string (required) }	{_id: <ObjectID> (required), name: string (required), type: string (required), age: int (required), habitat: string (required), user: string (corresponds to user's auth0_id) or null, zoo: zoo <ObjectID> or null }	{_id: <ObjectID> (required), name: string (required), city: string (required), state: string (required), animals: [animal <ObjectID>] (can be empty)}

B. **URL where the app is hosted:** https://quixotic-strand-309814.wl.r.appspot.com

C. Instructions for use:

a. Getting user info: I was not able to implement creating users. There are two sample users whose information can be used to log in and get jwts. On the main page, enter the email and password of the user. You will be redirected to a page that displays the user's auth0 sub and a jwt that can be used in Postman tests.

b. Sample users:

User 1

email: jill@email.com

password: jill_1234

User 2

email: sally@email.com

password: sally_1234

c. Testing with user info: *for user tests 2 through 7, part 1* log in with jill's email and password and copy/paste the user id and jwt into the current value for environment variables for user_id1 and jwt_1 (for some reason the test will not work if this value is pasted into initial value). These tests will illustrate CRUD operations with a valid JWT. *For user tests 7, part 2 through 10*, log in with sally's email and password and copy/paste the user id and jwt into the current value for the environment variables for user_id2 and jwt_2. These tests will illustrate that a user can't perform operations on another user's animal.

2. API Documentation:

User-Sponsored Animals

Get all Users

Returns all users

```
GET /user
```

Request

Path Parameters

None

Request Body

None

Request Body Format

n/a

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns response body

Response Examples

- Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "user1",
   "email": "user@email.com",
    "auth0_id": "46w7gtgh56"
}
```

Create a User-Sponsored animal

Creates a new animal with authentication.

POST /user/animal

Request

Path Parameters

None

Request Body

Required

Request Header

Authorization: bearer token

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the animal	Yes
type	String	The type of animal	Yes
age	Integer	Age of animal in years	Yes
habitat	String	Animal's habitat (ie jungle, arctic, ocean, etc)	Yes

Request Body Example

```
{
  "name": "Tiny",
  "type": "Rhinoceros",
  "age": 4,
  "habitat": "savannah"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Returns response body
Failure	400 Bad Request	Request is missing required parameters

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The user will be set to the req.user.sub and the zoo will be set to null
- The `self` attribute will contain the live link to the REST resource corresponding to this boat.
- Success

```
Status: 201 Created

{
  "id": "abc123",
  "name": "Tiny",
  "type": "Rhinoceros",
  "age": 4,
  "user": "fh36w5757"
  "zoo": null,
  "self": "https://<your-app>/animal/abc123"
}
```

Failure

```
Status: 400 Bad Request

{
"Error":  "Missing one or more of the required attributes"
}
```

Get all animals sponsored by a user

Gets all animals associated with a user, with authentication.

```
GET /user/animal
```

Request

Path Parameters

None

Request Body

None

Request Header

Authorization: bearer token

Request Body Format

N/a

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns response body
Failure	406 not acceptable	Client wants unacceptable response body format

Response Examples

- Results are paginated, with a limit of 5 items per page
- If more results exist, a link to the next page will be included
- Success

```
Status: 200 OK
```

```
{
  "id": "abc123",
  "name": "Tiny",
  "type": "Rhinoceros",
  "age": 4,
  "user": "fh36w5757"
  "zoo": null,
  "self": "https://<your-app>/animal/abc123"
}

{
  "id": "abc124",
  "name": "Orangey",
  "type": "Tiger",
  "age": 2,
  "user": "fh36w5757"
  "zoo": null,
  "self": "https://<your-app>/animal/abc124"
}
```

Failure

Status: 406 Not acceptable

```
{
"Error":  "Not acceptable"
}
```

Get one User-Sponsored animal

Gets one animal by id, with authentication.

```
GET/user/animal/:id
```

Request

Path Parameters

:id corresponds to animal’s id

Request Body

None

Request Header

Authorization: bearer token

Request Body Format

N/a

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns response body
Failure	406 Not acceptable	Client wants unacceptable response body format
Failure	403 Forbidden	This user does not have access to this animal
Failure	404 Not found	This animal does not exist

Response Examples

- Success

```
Status: 201 OK

{
  "id": "abc123",
  "name": "Tiny",
  "type": "Rhinoceros",
  "age": 4,
  "user": "fh36w5757"
  "zoo": null,
  "self": "https://<your-app>/animal/abc123"
}
```

Failure

```
Status: 406 Not acceptable

{
"Error":  "Not acceptable"
}
```

Failure

```
Status: 403 Forbidden

{
"Error":  "You do not have access to this animal"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No animal with this id in the database"
}
```

Edit one User-Sponsored animal

Edits an animal with authentication.

PATCH /user/animal/:id

Request

Path Parameters

:id corresponds to animal’s id in database

Request Body

Required

Request Header

Authorization: bearer token

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the animal	No
type	String	The type of animal	No
age	Integer	Age of animal in years	No
habitat	String	Animal’s habitat (ie jungle, arctic, ocean, etc)	No

Request Body Example

```
{
  "name": "Tiny",
  "type": "Rhinoceros",
  "age": 5,
  "habitat": "savannah"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Returns response body
Failure	404 Not found	Animal not found
Failure	403 forbidden	User doesn’t have access to animal
Failure	405 method not allowed	Client is trying to add a zoo through the wrong path

Response Examples

- Success

Status: 201 Created

```
{
  "id": "abc123",
  "name": "Tiny",
  "type": "Rhinoceros",
  "age": 5,
  "user": "fh36w5757"
  "zoo": null,
  "self": "https://<your-app>/animal/abc123"
}
```

Failure

Status: 404 Not found

```
{
"Error":  "No animal with this id in the database"
}
```

Failure

Status: 403 Forbidden

```
{
"Error":  "You do not have access to this animal"
}
```

Failure

Status: 405 Wrong Method

```
{
"Error":  "Wrong method. You can't add animal to zoo this way"
}
```

Delete one User-Sponsored animal

Deletes one animal by id, with authentication.

```
DELETE/user/animal/:id
```

Request

Path Parameters

:id corresponds to animal’s id

Request Body

None

Request Header

Authorization: bearer token

Request Body Format

N/a

Response

Response Body Format

Status code

Response Statuses

Outcome	Status Code	Notes
Success	204 deleted	Returns response body
Failure	403 Forbidden	This user does not have access to this animal
Failure	404 Not found	This animal does not exist

Response Examples

- If an animal is in a zoo, it will also be deleted from that zoo’s array

Success

```
Status: 204 Deleted
```

Failure

```
Status: 403 Forbidden

{
"Error":  "You do not have access to this animal"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No animal with this id in the database"
}
```

Animals

Create an animal

Creates a new animal

```
POST /animal
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the animal	Yes
type	String	The type of animal	Yes
age	Integer	Age of animal in years	Yes
habitat	String	Animal’s habitat (ie jungle, arctic, ocean, etc)	Yes

Request Body Example

```
{
  "name": "Bob",
  "type": "Penguin",
  "age": 2,
  "habitat": "arctic"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Returns response body
Failure	400 Bad Request	Request is missing required parameters

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The user and the zoo will be set to null
- The `self` attribute will contain the live link to the REST resource corresponding to this boat.
- Success

```
Status: 201 Created

{
  "id": "abc123",
  "name": "Bob",
  "type": "Penguin",
  "age": 2,
  "user": null,
  "zoo": null,
  "self": "https://<your-app>/animal/abc123"
}
```

Failure

```
Status: 400 Bad Request

{
"Error":  "Missing one or more of the required attributes"
}
```

Get all animals

Gets all animals

```
GET /animal
```

Request

Path Parameters

None

Request Body

None

Request Body Format

N/a

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns response body
Failure	406 not acceptable	Client wants unacceptable response body format

Response Examples

- Results are paginated, with a limit of 5 items per page
- If more results exist, a link to the next page will be included
- Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Tiny",
  "type": "Rhinoceros",
  "age": 4,
  "user": null,
  "zoo": null,
  "self": "https://<your-app>/animal/abc123"
}

{
  "id": "abc124",
  "name": "Orangey",
  "type": "Tiger",
  "age": 2,
  "user": null,
  "zoo": null,
  "self": "https://<your-app>/animal/abc124"
}
```

Failure

```
Status: 406 Not acceptable

{
"Error":  "Not acceptable"
}
```

Get one animal

Gets one animal by id

```
GET/animal/:id
```

Request

Path Parameters

:id corresponds to animal’s id

Request Body

None

Request Body Format

N/a

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns response body
Failure	406 Not acceptable	Client wants unacceptable response body format
Failure	403 Forbidden	This user does not have access to this animal
Failure	404 Not found	This animal does not exist

Response Examples

- Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Tiny",
  "type": "Rhinoceros",
  "age": 4,
  "user": null,
  "zoo": null,
  "self": "https://<your-app>/animal/abc123"
}
```

Failure

```
Status: 406 Not acceptable

{
"Error":  "Not acceptable"
}
```

Failure

```
Status: 403 Forbidden

{
"Error":  "You must be logged in to access this animal"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No animal with this id in the database"
}
```

Edit one animal

Edits one animal by id

```
PATCH /animal/:id
```

Request

Path Parameters

:id corresponds to animal's id in database

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	**Type**	**Description**	**Required?**
name	String	The name of the animal	No
type	String	The type of animal	No
age	Integer	Age of animal in years	No

habitat	String	Animal’s habitat (ie jungle, arctic, ocean, etc)	No

Request Body Example

```
{
  "name": "Tiny",
  "type": "Rhinoceros",
  "age": 5,
  "habitat": "savannah"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	201 Created	Returns response body
Failure	404 Not found	Animal not found
Failure	403 forbidden	User doesn’t have access to animal
Failure	405 method not allowed	Client is trying to add a zoo through the wrong path

Response Examples

- Success

```
Status: 201 Created

{
  "id": "abc123",
  "name": "Tiny",
  "type": "Rhinoceros",
  "age": 5,
  "user": "fh36w5757"
  "zoo": null,
  "self": "https://<your-app>/animal/abc123"
}
```

Failure

```
Status: 404 Not found

{
"Error":  "No animal with this id in the database"
}
```

Failure

```
Status: 403 Forbidden

{
"Error":  "You must be logged in to access this animal"
}
```

Failure

```
Status: 405 Wrong Method

{
"Error":  "Wrong method. You can't add animal to zoo this way"
}
```

Delete one animal

Deletes one animal by id

```
DELETE/animal/:id
```

Request

Path Parameters

:id corresponds to animal’s id

Request Body

None

Request Body Format

N/a

Response

Response Body Format

Status code

Response Statuses

Outcome	Status Code	Notes
Success	204 deleted	Returns response body
Failure	403 Forbidden	This user does not have access to this animal
Failure	404 Not found	This animal does not exist

Response Examples

- If an animal is in a zoo, it will also be deleted from that zoo’s array

Success

```
Status: 204 Deleted
```

Failure

```
Status: 403 Forbidden

{
"Error":  "You must be logged in to access this animal"
}
```

Failure

```
Status: 404 Not Found
```

```
{
"Error":  "No animal with this id in the database"
}
```

Zoos

Create a zoo

Creates a new zoo

```
POST /zoo
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the zoo	Yes
city	String	The city the zoo is in	Yes
state	String	The state the zoo is in	Yes

Request Body Example

```
{
  "name": "Oregon Zoo",
  "city": "Portland",
  "state": "OR"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Returns response body
Failure	400 Bad Request	Request is missing required parameters

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The animals attribute is set to an empty array
- The `self` attribute will contain the live link to the REST resource corresponding to this boat.
- Success

```
Status: 201 Created

{
  "id": "abc123",
  "name": "Oregon Zoo",
  "city": "Portland",
  "state": "OR",
  "animals": [],
  "self": "https://<your-app>/zoo/abc123"
}
```

Failure

```
Status: 400 Bad Request

{
"Error":  "Missing one or more of the required attributes"
}
```

Get all zoos

Gets all animals

```
GET /zoo
```

Request

Path Parameters

None

Request Body

None

Request Body Format

N/a

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns response body

Failure	406 not acceptable	Client wants unacceptable response body format

Response Examples

- Results are paginated, with a limit of 5 items per page
- If more results exist, a link to the next page will be included

Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Oregon Zoo",
  "city": "Portland",
  "state": "OR",
  "animals": [],
  "self": "https://<your-app>/zoo/abc123"
}

{
  "id": "abc124",
  "name": "Atlanta Zoo",
  "city": "Atlanta",
  "state": "GA",
  "animals": [“dfhjh657”, “fgfhsaf”],
  "self": "https://<your-app>/zoo/abc124"
}
```

Failure

```
Status: 406 Not acceptable

{
"Error":  "Not acceptable"
}
```

Get one zoo

Gets one zoo by id

```
GET/zoo/:id
```

Request

Path Parameters

:id corresponds to zoo’s id

Request Body

None

Request Body Format

N/a

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns response body
Failure	406 Not acceptable	Client wants unacceptable response body format
Failure	404 Not found	This zoo does not exist

Response Examples

- Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Oregon Zoo",
  "city": "Portland",
  "state": "OR",
  "animals": [],
  "self": "https://<your-app>/zoo/abc123"

}
```

Failure

```
Status: 406 Not acceptable

{
"Error":  "Not acceptable"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No zoo with this id in the database"
}
```

Edit one zoo

Edits one zoo by id

```
PATCH /zoo/:id
```

Request

Path Parameters

:id corresponds to zoo's id in database

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the zoo	No
city	String	The city the zoo is in	No
state	String	State the zoo is in	No

Request Body Example

```
{
  "name": "Portland Zoo"}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Returns response body
Failure	404 Not found	Zoo not found
Failure	405 method not allowed	Client is trying to add an animal to a zoo through the wrong path

Response Examples

Success

```
Status: 201 Created

{
  "id": "abc123",
  "name": "Portland Zoo",
  "city": "Portland",
  "state": "OR",
  "animals": [],
  "self": "https://<your-app>/zoo/abc123"

}
```

Failure

```
Status: 404 Not found
```

```
{
"Error":  "No zoo with this id in the database"
}
```

Failure

```
Status: 406 Not acceptable

{
"Error":  "Not acceptable"
}
```

Failure

```
Status: 405 Wrong Method

{
"Error":  "Wrong method. You can't add animal to zoo this way"
}
```

Delete one zoo

Deletes a zoo by id

```
DELETE/zoo/:id
```

Request

Path Parameters

:id corresponds to zoo’s id

Request Body

None

Request Body Format

N/a

Response

Response Body Format

Status code

Response Statuses

Outcome	**Status Code**	**Notes**
Success	204 deleted	Returns status code
Failure	404 Not found	This animal does not exist

Response Examples

- If a zoo contains animals, those animals’ zoo will be set to null

Success

```
Status: 204 Deleted
```

Failure

```
Status: 404 Not Found

{
"Error":  "No zoo with this id in the database"
}
```

Zoo-Animal Relationship

Add animal to zoo

Adds an animal to a zoo’s array

```
PUT /zoo/:zoo_id/:animal_id
```

Request

Path Parameters

:zoo_id corresponds to zoo’s id in database, :animal_id corresponds to animal’s id in database

Request Body

None

Response

Response Body Format

Status code

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Returns status code
Failure	404 Not found	Zoo/animal not found
Failure	403 Forbidden	Animal is already in another zoo

Response Examples

Success

```
Status: 201 Created
```

Failure

```
Status: 404 Not found

{
```

```
"Error":  "No zoo and/or animal with this id in the database"
}
```

Failure

```
Status: 403 Forbidden

{
"Error":  "This animal is already in another zoo"
}
```

Remove animal from zoo

Removes an animal from zoo’s array

```
DELETE/zoo/:id
```

Request

Path Parameters

:zoo_id corresponds to zoo’s id, :animal_id corresponds to animal’s id

Request Body

None

Request Body Format

N/a

Response

Response Body Format

Status code

Response Statuses

Outcome	**Status Code**	**Notes**
Success	204 deleted	Returns status code
Failure	404 Not found	This animal/zoo does not exist
Failure	403 Forbidden	Animal in another zoo

Response Examples

Success

```
Status: 204 Deleted
```

Failure

```
Status: 404 Not Found

{
"Error":  "No zoo with this id in the database"
}
```

Failure

```
Status: 403 Forbidden

{
"Error":  "This animal is already in another zoo"
}
```