Final Project CS 493

Application URL: <http://skinnern-cs493-final.appspot.com/>

I do not have an account creation url, more info in the Non Implemented Features section

Login URL: <http://skinnern-cs493-final.appspot.com/login>

**Non-User Entities in Datastore**

My application has the Truck and Shipment entities. The relationship between them being that you can put a shipment onto a truck, with the intent for that to be the shipment that will be sent out.

A truck will be composed of the following:

|  |
| --- |
| {  "name": "new truck",  "type": "new",  "length": 100  } |

A shipment will be composed of the following:

|  |
| --- |
| {  "weight": 500,  "content": "spoons",  "delivery\_date": "future"  } |

\*\*\*\*\*NOTE\*\*\*\*\*

I have 7 errors in my postman tests, the errors are all related to a feature that I have not fully implemented, they are related to the “[Endpoints are provided to create and remove relationship between non-user entities.](#_6ld02maq8dl3)” section, documented in my not-fully implemented section below.

**Modeling the user entity**

* **What is the unique identifier for a user in your data model.**
  + The unique identifier for my user data model is the email used for login, when a user initially logs in they will get their email put into the datastore along with a unique identifier provided by google.
* **If a request needs to supply a user identifier, what needs to be specified by the person making the request.**
  + After logging in, the user will be supplied with a JWT token for their user
  + The person making the request needs to supply the JWT authorization token along with their request so that Auth0 will allow them to make modifications to the datastore
* **How your application maps a supplied JWT to the identifier of a user.**
  + My application uses Auth0 to check the JWT to the identifier of a user, and checks the jwt across Auth0 to make sure the user is valid
* **What is the relationship between the user entity and non-user entity.**
  + In my application, the Trucks need an owner to create them, and the user who created them will be assigned as the owner.

**API specification**

[**CREATE A TRUCK**](#_5rbo79g53kun) **4**

[**GET A TRUCK**](#_qn3m3c3knuei) **6**

[**LIST OF ALL TRUCKS**](#_39qmg05b8cdn) **8**

[**EDIT A TRUCK**](#_7vbodrm3sqsr) **10**

[**DELETE A TRUCK**](#_rfptmsddv2dr) **13**

[**CREATE A SHIPMENT**](#_5pq0isolsy3g) **15**

[**GET A SHIPMENT**](#_u5c21zdfwxud) **17**

[**LIST OF ALL SHIPMENTS**](#_gbshf7i1x1b) **19**

[**EDIT A SHIPMENT**](#_a15osxf1v7gt) **21**

[**DELETE A SHIPMENT**](#_mgopkvdwsg4c) **23**

[**TRUCK ADDS A SHIPMENT**](#_s4caw9j5hyg8) **25**

[**TRUCK REMOVES A SHIPMENT**](#_qxrauc746302) **27**

[**VIEW TRUCK SHIPMENT**](#_s3ic9jaxg18k) **29**

[**Non-Implemented Features**](#_po0ss4jh6ryi) **31**

[Allow users to create a login](#_o8ihj3mc3qez) 31

[Endpoints are provided to create and remove relationship between non-user entities.](#_6ld02maq8dl3) 31

# CREATE A TRUCK

Allows you to create a new truck

|  |
| --- |
| post /trucks |

Request

Request Header

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| Content-Type | application/json | Content format to send to server |
| Authorization | {{JWT}} | The user’s JWT |

Request Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| name | string | Name of the truck | yes |
| type | string | Type of truck | yes |
| length | integer | Length of truck | yes |

Request body example

|  |
| --- |
| {  "name": "new truck",  "type": "new",  "length": 28  } |

Response

**Response Body Format**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **201** | **Boat was made** |
| **failure** | **400 bad request** | **Boat is missing attributes** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500 unacceptable type** | **Request was not in application/json format** |

**Response Examples**

• Datastore will automatically generate an ID and store it with the entity being created.

|  |
| --- |
| **{**  **"Trucks":[{**  **"Type":"new",**  **"Length":100000,**  **"owner":"**[**skinnern@oregonstate.edu**](mailto:skinnern@oregonstate.edu)**",**  **"name":"new truck",**  **"Id":"5677303661068288",**  **"self":"http://skinnern-cs493-final.appspot.com/trucks/5677303661068288"}**  **]}** |

# GET A TRUCK

Allows you to get an existing truck

|  |
| --- |
| Get /trucks/[truckid] |

Request

Request Header

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| Authorization | {{JWT}} | The user’s JWT |

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| Boat id | string | Id of the truck | yes |

Request Body

none

Response

**Response Body Format**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **200 OK** | **Boat was found** |
| **failure** | **404 not found** | **Boat with truck\_id was not found** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• the datastore should display attributes of the selected truck, unless the selected truck does not exist

|  |
| --- |
| **{**  **"Trucks":[{**  **"Type":"new",**  **"Length":100000,**  **"owner":"**[**skinnern@oregonstate.edu**](mailto:skinnern@oregonstate.edu)**",**  **"name":"new truck",**  **"Id":"5677303661068288",**  **"self":"http://skinnern-cs493-final.appspot.com/trucks/5677303661068288"}**  **]}** |

# LIST OF ALL TRUCKS

Allows you to get all existing trucks

|  |
| --- |
| Get /trucks |

Request

Request Parameters

None

Request Body

none

Response

**Response Body Format**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **200 OK** | **Listing all trucks** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• this should display a list of all trucks contained in the datastore

|  |
| --- |
| {"trucks":[{"type":"smiley","length":100000,"owner":"skinnern@oregonstate.edu","name":"happy truck","id":"5677303661068288","self":"http://skinnern-cs493-final.appspot.com/trucks/5677303661068288"},{"length":100,"owner":"skinnern@oregonstate.edu","name":"new truck","type":"small","id":"5748345607815168","self":"http://skinnern-cs493-final.appspot.com/trucks/5748345607815168"}]} |

# EDIT A TRUCK

**Allows you to edit a truck**

|  |
| --- |
| PUT /trucks/[truckid] |

Request

Request Header

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| Authorization | {{JWT}} | The user’s JWT |
| Content-Type | application/json | Content type of request |

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| Boat id | string | Id of the truck | yes |

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| name | string | Name of the truck | yes |
| type | string | Type of truck | yes |
| length | integer | Length of truck | yes |

Request body example

|  |
| --- |
| {  "name": "new truck",  "type": "new",  "length": 28  } |

Response

**Response Body Format**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **303** | **redirect** |
| **failure** | **403** | **you don't have permission to do this** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• the datastore ship should be edited with the proper attributes, and will return the current attributes of the ship

|  |
| --- |
| **{**  **"Trucks":[{**  **"Type":"new",**  **"Length":100000,**  **"owner":"**[**skinnern@oregonstate.edu**](mailto:skinnern@oregonstate.edu)**",**  **"name":"new truck",**  **"Id":"5677303661068288",**  **"self":"http://skinnern-cs493-final.appspot.com/trucks/5677303661068288"}**  **]}** |

# DELETE A TRUCK

**Allows you to delete a truck**

**If a truck is deleted, any shipment relationships that had existed will become nullified**

|  |
| --- |
| DELETE /trucks/[truckid] |

Request

Request Header

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| Authorization | {{JWT}} | The user’s JWT |

Request Parameters

None

Request Body

none

Request Body Format

JSON

Response

**Response Body Format**

**Success: no body**

**Failure: JSON**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **204 no content** | **Boat was deleted** |
| **failure** | **403** | **You don't have permission to do this** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• if delete is successful, no message will be sent, if failure, will send the following:

|  |
| --- |
| {  Error: "No truck with this truck\_id exists"  } |

# CREATE A SHIPMENT

Allows you to create a new truck

|  |
| --- |
| post /shipments |

Request

Request Header

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| Content-Type | application/json | The content type of the submission |

Request Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| content | string | contents | yes |
| delivery\_date | string | Date to deliver | yes |
| weight | integer | Weight of deliverable | yes |

Request body example

|  |
| --- |
| {  "weight": 500,  "content": "Megablocks",  "delivery\_date": "05/05/2019"  } |

Response

**Response Body Format**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **201** | **Load was made** |
| **failure** | **400 bad request** | **Load is missing attributes** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• Datastore will automatically generate an ID and store it with the entity being created.

|  |
| --- |
| {"delivery\_date":"never","content":"People","weight":800,"id":"5086100271923200","self":"http://skinnern-cs493-final.appspot.com/shipments/5086100271923200"} |

# GET A SHIPMENT

Allows you to get an existing shipment

|  |
| --- |
| Get /shipments/[shipmentid] |

Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| Load id | string | Id of the shipment | yes |

Request Body

none

Response

**Response Body Format**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **200 OK** | **Load was found** |
| **failure** | **404 not found** | **Load with shipment\_id was not found** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• the datastore should return an individual item as long as a proper object was specified

|  |
| --- |
| {"delivery\_date":"never","content":"People","weight":800,"id":"5086100271923200","self":"http://skinnern-cs493-final.appspot.com/shipments/5086100271923200"} |

# LIST OF ALL SHIPMENTS

Allows you to get all existing shipments

|  |
| --- |
| Get /shipments |

Request

Request Parameters

None

Request Body

none

Response

**Response Body Format**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **200 OK** | **Listing all shipments** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• this will cause the datastore to grab all shipments that it can find, these results will be pagnitated 3 at a time

|  |
| --- |
| {  "shipments": [{"delivery\_date":"never","content":"People","weight":800,"id":"5086100271923200","self":"http://skinnern-cs493-final.appspot.com/shipments/5086100271923200"}  ]  } |

# 

# EDIT A SHIPMENT

**Allows you to edit a truck**

|  |
| --- |
| PUT /trucks/[truckid] |

Request

Request Header

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| Authorization | {{JWT}} | The user’s JWT |
| Content-Type | application/json | Content type of request |

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| Boat id | string | Id of the truck | yes |

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| content | string | contents | yes |
| delivery\_date | string | Date to deliver | yes |
| weight | integer | Weight of deliverable | yes |

Request body example

|  |
| --- |
| {  "weight": 500,  "content": "Megablocks",  "delivery\_date": "05/05/2019"  } |

Response

**Response Body Format**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **303** | **redirect** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• the datastore ship should be edited with the proper attributes, and will return the current attributes of the ship

|  |
| --- |
| {  "shipments": [{"delivery\_date":"never","content":"People","weight":800,"id":"5086100271923200","self":"http://skinnern-cs493-final.appspot.com/shipments/5086100271923200"}  ] |

# DELETE A SHIPMENT

**Allows you to delete a shipment**

**If a shipment is deleted, any truck relationships that had existed will become nullified**

|  |
| --- |
| DELETE /shipments/[shipmentid] |

Request

Request Parameters

None

Request Body

none

Request Body Format

JSON

Response

**Response Body Format**

**Success: no body**

**Failure: JSON**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **204 no content** | **Load was deleted** |
| **failure** | **403** | **You don't have permission to do this** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• if delete is successful, no message will be sent, if failure, will send the following:

|  |
| --- |
| {  Error: "No truck with this truck\_id exists"  } |

# TRUCK ADDS A SHIPMENT

Allows you to create a new truck

|  |
| --- |
| Put /shipments/[shipmentid]trucks/[truckid] |

Request

Request Header

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| Authorization | {{JWT}} | The user’s JWT |

Request Parameters

None

Request Body

Required

Request Body Format

JSON

Response

**Response Body Format**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **201** | **Load was added to truck** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

* I did not implement a response if it was successful, but I did give a failure state:

|  |
| --- |
| {  "Error": "Could not put Shipment on Truck. That Shipment is on another Truck."  } |

# TRUCK REMOVES A SHIPMENT

Allows you to remove a shipment from a truck

|  |
| --- |
| Delete /trucks/[truckid]/shipment/[shipmentid] |

Request

Request Header

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| Authorization | {{JWT}} | The user’s JWT |

Request Parameters

None

Request Body

none

Request Body Format

JSON

Response

**Response Body Format**

**Success: no body**

**Failure: JSON**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **204 no content** | **Load was deleted** |
| **failure** | **403** | **You don't have permission to do this** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• if delete is successful, no message will be sent, if an error occurs, it should look like one of the following:

|  |
| --- |
| {unacceptable type, please try application json}  {you dont have permission to do this} |

# VIEW TRUCK SHIPMENT

Allows you to get all existing shipments associated with a truck

|  |
| --- |
| Get /trucks/[truckid]/shipments |

Request

Request Header

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| Authorization | {{JWT}} | The user’s JWT |

Request Parameters

None

Request Body

none

Response

**Response Body Format**

**JSON**

**Response Statuses**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status code** | **notes** |
| **success** | **200 OK** | **Listing all shipments** |
| **failure** | **406 unacceptable type** | **Accept type was not application/json** |
| **failure** | **500** | **Invalid content type** |

**Response Examples**

• this will cause the datastore to grab all shipments that it can find

|  |
| --- |
| {"delivery\_date":"never","content":"People","weight":800,"id":"5086100271923200","self":"http://skinnern-cs493-final.appspot.com/shipments/5086100271923200"} |

# Non-Implemented Features

## Allow users to create a login

How you intended to meet the requirement.

* I intended to give the user an Auth0 login page which would add the user to my login, and put their email into my user table in my datastore. This login page would have allowed the user to sign in with a Google account, and would send back the jwt for postman to grab.

What steps you attempted in order to meet the requirement.

* I created a post user function, and would send the user information to Auth0 to create a new user account for the signup. When testing this out, I would receive forbidden errors, and the user would not be created. After some troubleshooting I managed to get a new error, but still had no idea how to get around it.

What you suspect was the most likely reason it didn't work.

* I think that some Auth0 configuration might be wrong. I had the same issue with Assignment 7, where you could create an account, but it was not a google account, and the redirect after creating an account would give a forbidden error. These combined make me think that I may have missed a few steps in setting up my Auth0 environment.

## Endpoints are provided to create and remove relationship between non-user entities.

I have a working solution to create relationships between the entities, but I do not have a working system to remove these relationships

How you intended to meet the requirement.

* I intended to set up a function to remove a Shipment from a Truck, and a Truck from a Shipment, like we did with Boats and Loads from the assignments, this would allow the user to freely add and remove a Shipment from a truck. As it stands, I have a working function to place a Shipment onto a truck, but no function to remove it.

What steps you attempted in order to meet the requirement.

* As stated, I have the function to place the Shipment onto a Truck. However, I do not have a working function to remove the truck from the shipment and remove the shipment from the truck. I wrote a function “remove\_shipmentFromTruck”, which contains what I made to delete the relationship, which checks if it needs to remove all shipments, or just a few. After it checks and removes the truck, it will then check and remove the shipment.

What you suspect was the most likely reason it didn't work.

* Mostly just that I ran out of time, and was unable to get a working delete for the relationship. I tried copying and pasting the version that I made for a previous assignment, and modifying it to work for this final, but that had some different issues. I think if I had been able to tweak my current function a bit more to get it to delete the relationship, or at least null out the variables.