

API Spec

Steven Au

<https://final-350500.uc.r.appspot.com>

Change log	2
Access	2
Login and Registration	2
Relationship Between Entities (Non-User)	2
User Entity Modeling (With User)	2
Data Model	3
Authorization Errors	5
Create a Boat	6
Get a Boat	9
List all Boats	11
Edit a Boat - Patch	14
Edit a Boat - Put	17
Delete a Boat	20
Load a Boat	22
Remove a Load from a Boat	25
Create a Load	27
Get a Load	30
List all Loads	32
Edit a Load - Patch	35
Edit a Load - Put	38
Delete a load	41
List all Users	43
Login	45
Register	47
User Information	49
Decode	51

Change log

Version	Change	Date
1.0	Initial version.	April 8, 2022
1.1	Update load and boat details	April 21, 2022
2.0	Incorporate users model and user authentication details.	May 31, 2022

Access

Main route: <https://final-350500.uc.r.appspot.com>

Account creation: <https://final-350500.uc.r.appspot.com>

Login and Registration

On the main route/account creation screen, simply enter the appropriate email and password (passwords at 8 character length minimum) and click either the appropriate login or register buttons. The JWT, datastore id, and the Auth0 user id (referred to as the “owner” in the non-user entities) will be shown.

Relationship Between Entities (Non-User)

Each entity, boat and load, relate to one another by the process of loading a load onto a ship. This can be performed by following the endpoint described on the “Load a Boat” and “Remove a Load From a Boat” pages. Note that once loaded, the boat’s load property will be populated with the load ids. Similarly, the load carrier will be populated with the boat id.

User Entity Modeling (With User)

Each owner id is generated through the Authentication and Authorization JWT token. The id, known as the owner, is used to identify the ownership of the boat and load entities. To specify who is attempting to modify an entity, one must also send the JWT token in the Authorization header. Else, the request will be rejected if the owner value in the JWT and entity does not match.

Each entity created will always have the additional attribute called owner. This property maps to the identifying owner supplied through the JWT token. As such, the relationship between the user and non-user entities is based on the owner attribute identified when the entity is first created.

- See the Data Model section for the data model.

Data Model

The app stores these kinds of entities in Datastore, Users, Boats, and Loads.

Users

Property	Data Type	Validation	Notes
id	Integer	Not by user	The id of the user stored. Datastore automatically generates it.
user_id	String	Not by user	This is the registered user's Auth0 user ID. This is used as the "owner" to identify authentication and authorization ownership of Boats and Loads.

- The JWT is not stored in the database

Boats

Property	Data Type	Validation	Notes
id	Integer	Not by user	The id of the boat. Datastore automatically generates it.
name	String	Must be greater than 1 character length.	Name of the boat.
type	String	Must be greater than or equal to 1 character length.	Type of the boat. E.g., Sailboat, Catamaran, etc.
length	Integer	Must be greater than or equal to 1.	The length of the boat in feet.
loads	List of ints	Not by user	The list of load ids as JSON objects current loaded on the boat, else an empty list.
owner	String	Not by user	This is the registered user's Auth0 user id – the boat's owner.

Loads

Property	Data Type	Validation	Notes
id	Integer	Not by user	The id of the load. Datastore automatically generates it.
volume	Integer	Must be greater than or equal to 1.	The volume of the load.
creation_date	String	Must be greater than 1 character length.	The date the load was created.
item	String	Must be greater than or equal to 1 character length.	The name of the item currently inside the load. E.g. LEGO Bricks
carrier	JSON Object	Not by user	If there is a boat assigned, then id of that boat as a JSON object. Otherwise null. Only one carrier can be assigned to at any given time.
owner	String	Not by user	This is the registered user's Auth0

			user id – the load’s owner.
--	--	--	-----------------------------

Notes

- All “self” properties are generated on the fly and are not stored in the Datastore.

Authorization Errors

List of authorization errors that may occur per JWT verification. Please follow the description to determine the solution.

Status: 401 Unauthorized.

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

```
{
  "code": "invalid_header",
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
```

```
{
  "code": "token_expired",
  "description": "token is expired"
}
```

```
{
  "code": "invalid_claims",
  "description": "incorrect claims, please check the audience and issuer"
}
```

```
{
  "code": "no_rsa_key",
  "description": "No RSA key in JWKS"
}
```

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes

Request JSON Attributes

Name	Description	Type	Validation	Required?
name	The name of the boat.	String	Must be greater than 1 character length.	Yes
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	String	Must be greater than or equal to 1 character length.	Yes
length	Length of the boat in feet.	Integer	Must be greater than or equal to 1.	Yes

Request Body Example

```
{  
  "name": "Sea Witch",  
  "type": "Catamaran",  
  "length": 28  
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	A response with the data created will be returned.
Failure	400 Bad Request	If the request is missing any of the 3 required attributes or required data validation, the boat is not created.

Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .
Failure	415 Unsupported Media Type	The content sent is not in the requested JSON format.

Response Examples

Success

Status: 201 Created

```
{
  "id": 5657040122806272,
  "length": 28,
  "loads": [],
  "name": "Sea Witch",
  "owner": "auth0|627986546a0dc500692142f5",
  "self": "https://final-350500.uc.r.appspot.com/boats/5657040122806272",
  "type": "Catamaran"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Status: 405 Method Not Allowed

```
{
  "Error": "Method not allowed"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Client must accept valid JSON"
}
```

```
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Content must be sent as valid JSON"  
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.
- Datastore will automatically generate an ID and store it with the entity being created.
- The “self” property will be automatically created but is not actually stored in the datastore.

Get a Boat

Allows you to get an existing boat

GET /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Loads will be an empty list if the boat is currently not loaded. Else, the ID of the load. See "Notes" regarding self.
Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	403 Forbidden	Attempted to access the entity that does not belong to the currently verified JWT user.
Failure	404 Not Found	No boat with this boat_id exists
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .

Response Examples

Success

Status: 200 OK

{

```
{
  "id": 5657040122806272,
  "length": 28,
  "loads": [],
  "name": "Sea Witch",
  "owner": "auth0|627986546a0dc500692142f5",
  "self": "https://final-350500.uc.r.appspot.com/boats/5657040122806272",
  "type": "Catamaran"
}
```

Failure

Status: 401 Unauthorized

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Status: 403 Forbidden

```
{
  "Error": "This boat does not belong to you"
}
```

Status: 404 Not Found

```
{
  "Error": "No boat with this boat_id exists"
}
```

Status: 405 Method Not Allowed

```
{
  "Error": "Method not allowed"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Client must accept valid JSON"
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.
- Datastore will automatically generate an ID and store it with the entity being created.
- The “self” property will be automatically created but is not actually stored in the datastore.

List all Boats

List all the boats - only the user's boats if authenticated correctly, else all boats. Both scenarios will have a pagination of 5 per page.

GET /boats

Request

Path Parameters

None

Request Body

None

Protected

No – if an authentication was verified correctly, then the response will be the user's boats. Else, all the boats on record.

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	No
Accept	application/json or */*	Client response accept values	Yes

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	All boats if a JWT fails to be verified or sent. Else, all boats owned by the authenticated user. The next attribute will only appear if there are more than 5 boats on the page – URL links to the next page. The total entries in the database will be displayed in the "total" attribute. Loads will be an empty list if the boat is currently not loaded. Else, the ID of the load. See "Notes" regarding self.
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .

Response Examples

Success

Status: 200 OK

```
{
  "boats": [
    {
      "id": 5657040122806272,
      "length": 28,
      "loads": [],
      "name": "Sea Witch",
      "owner": "auth0|627986546a0dc500692142f5",
      "self": "https://final-350500.uc.r.appspot.com/boats/5657040122806272",
      "type": "Catamaran"
    },
    {
      "id": 5675369768157184,
      "length": 500,
      "loads": [],
      "name": "Shippy",
      "owner": "auth0|627986546a0dc500692142f5",
      "self": "https://final-350500.uc.r.appspot.com/boats/5675369768157184",
      "type": "Barge"
    },
    {
      "id": 5680760220549120,
      "length": 30,
      "loads": [],
      "name": "The London",
      "owner": "auth0|627986546a0dc500692142f5",
      "self": "https://final-350500.uc.r.appspot.com/boats/5680760220549120",
      "type": "Catamaran"
    },
    {
      "id": 5684987139457024,
      "length": 45,
      "loads": [],
      "name": "Mermaid",
      "owner": "auth0|627986546a0dc500692142f5",
      "self": "https://final-350500.uc.r.appspot.com/boats/5684987139457024",
      "type": "Speedboat"
    },
    {
      "id": 5687441444831232,
      "length": 1000,
      "loads": [],
      "name": "USS Enterprise",

```

```
"owner": "auth0|627986546a0dc500692142f5",  
"self": "https://final-350500.uc.r.appspot.com/boats/5687441444831232",  
"type": "Aircraft Carrier"  
}  
],  
"next": "https://final-350500.uc.r.appspot.com/boats?limit=5&offset=5",  
"total": 6  
}
```

Failure

Status: 405 Method Not Allowed

```
{  
  "Error": "Method not allowed"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.
- Datastore will automatically generate an ID and store it with the entity being created.
- The “self” property will be automatically created but is not actually stored in the datastore.

Edit a Boat - Patch

Allows you to edit some of the attributes of a boat.

PATCH /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Body Format

JSON

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes

Request JSON Attributes

At least one of these attributes must be within the request.

Name	Description	Type	Validation	Required?
name	The name of the boat.	String	Must be greater than 1 character length.	No
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	String	Must be greater than or equal to 1 character length.	No
length	Length of the boat in feet.	Integer	Must be greater than or equal to 1.	No

Request Body Example

```
{  
  "name": "thee Sea Witch",  
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	A response with the data created will be returned.
Failure	400 Bad Request	If the request is missing any of the attributes or required

		data validation, the boat is not updated.
Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	403 Forbidden	Attempted to access the entity that does not belong to the currently verified JWT user.
Failure	404 Not Found	No boat with this boat_id exists
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .
Failure	415 Unsupported Media Type	The content sent is not in the requested JSON format.

Response Examples

Success

Status: 201 Created

```
{
  "id": 5657040122806272,
  "length": 28,
  "loads": [],
  "name": "thee Sea Witch",
  "owner": "auth0|627986546a0dc500692142f5",
  "self": "https://final-350500.uc.r.appspot.com/boats/5657040122806272",
  "type": "Catamaran"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Status: 403 Forbidden

```
{
  "Error": "This boat does not belong to you"
}
```

Status: 404 Not Found

```
{  
  "Error": "No boat with this boat_id exists"  
}
```

Status: 405 Method Not Allowed

```
{  
  "Error": "Method not allowed"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Content must be sent as valid JSON"  
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.
- Datastore will automatically generate an ID and store it with the entity being created.
- The “self” property will be automatically created but is not actually stored in the datastore.

Edit a Boat - Put

Allows you to edit all the attributes of a boat.

PUT /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Body Format

JSON

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes

Request JSON Attributes

Name	Description	Type	Validation	Required?
name	The name of the boat.	String	Must be greater than 1 character length.	Yes
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	String	Must be greater than or equal to 1 character length.	Yes
length	Length of the boat in feet.	Integer	Must be greater than or equal to 1.	Yes

Request Body Example

```
{
  "name": "Sea Witch4",
  "type": "Catamaran4",
  "length": 284
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	A response with the data created will be returned.
Failure	400 Bad Request	If the request is missing any of the attributes or required

		data validation, the boat is not updated.
Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	403 Forbidden	Attempted to access the entity that does not belong to the currently verified JWT user.
Failure	404 Not Found	No boat with this boat_id exists
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .
Failure	415 Unsupported Media Type	The content sent is not in the requested JSON format.

Response Examples

Success

Status: 201 Created

```
{
  "id": 5657040122806272,
  "length": 284,
  "loads": [],
  "name": "Sea Witch4",
  "owner": "auth0|627986546a0dc500692142f5",
  "self": "https://final-350500.uc.r.appspot.com/boats/5657040122806272",
  "type": "Catamaran4"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Status: 403 Forbidden

```
{
  "Error": "This boat does not belong to you"
}
```

Status: 404 Not Found

```
{  
  "Error": "No boat with this boat_id exists"  
}
```

Status: 405 Method Not Allowed

```
{  
  "Error": "Method not allowed"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Content must be sent as valid JSON"  
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.
- Datastore will automatically generate an ID and store it with the entity being created.
- The “self” property will be automatically created but is not actually stored in the datastore.

Delete a Boat

Allows you to delete a boat. Note that if the boat is currently loaded, deleting the boat unloads the loads.

DELETE /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes (Due to error statuses)

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	403 Forbidden	Attempted to access the entity that does not belong to the currently verified JWT user.
Failure	404 Not Found	No boat with this boat_id exists
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{  
  "code": "invalid_header",  
  "description": "Unable to parse authentication token."  
}
```

Status: 403 Forbidden

```
{  
  "Error": "This boat does not belong to you"  
}
```

Status: 404 Not Found

```
{  
  "Error": "No boat with this boat_id exists"  
}
```

Status: 405 Method Not Allowed

```
{  
  "Error": "Method not allowed"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.

Load a Boat

Allows you to add a load to a boat. To change a boat's load, please delete the load then put the new load with this route.

PUT /boats/:boat_id/loads/:load_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat
load_id	ID of the load

Request Body

None

Request Body Format

None

Request JSON Attributes

None

Request Body Example

None

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes (Due to error statuses)

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	403 Forbidden	Two cases: <ul style="list-style-type: none">Attempted to access the entity that does not belong

		to the currently verified JWT user. <ul style="list-style-type: none"> A load can only be assigned to only one boat at any time
Failure	404 Not Found	Load or boat does not exist
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither <code>application/json</code> nor <code>*/*</code> .

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Status: 403 Forbidden

```
{
  "Error": "The load is already loaded on another boat"
}
```

Status: 403 Forbidden

```
{
  "Error": " This boat does not belong to you"
}
```

Status: 404 Not Found

```
{
  "Error": "The specified boat and/or load does not exist"
}
```

Status: 405 Method Not Allowed

```
{
  "Error": "Method not allowed"
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```

Notes

See Authorization Errors for details regarding each status code 401 Unauthorized error.

Remove a Load from a Boat

Allows you to remove a load to a boat. To change a load, please delete the load with this route. Then put a new load.

DELETE /boats/:boat_id/loads/:load_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat
load_id	ID of the load

Request Body

None

Request Body Format

None

Request JSON Attributes

None

Request Body Example

None

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes (Due to error statuses)

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	403 Forbidden	Attempted to access the entity that does not belong to the currently verified JWT user.
Failure	404 Not Found	Causes:

		<ul style="list-style-type: none"> • Load or boat does not exist. • There are no loads on this boat. • This load is not on this boat.
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither <code>application/json</code> nor <code>*/*</code> .

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Status: 403 Forbidden

```
{
  "Error": " This boat does not belong to you"
}
```

Status: 404 Not Found

```
{
  "Error": "No boat with this boat_id is loaded with the load with this load_id"
}
```

Status: 405 Method Not Allowed

```
{
  "Error": "Method not allowed"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Client must accept valid JSON"
}
```

Notes

See Authorization Errors for details regarding each status code 401 Unauthorized error.

Create a Load

Allows you to create a new Load

POST /loads

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes

Request JSON Attributes

Name	Description	Type	Validation	Required?
item	The item on the load	String	Must be greater than 1 character length.	Yes
creation_date	The date the load was created.	String	Must be greater than or equal to 1 character length.	Yes
volume	The volume of the load	Integer	Must be greater than or equal to 1.	Yes

Request Body Example

```
{  
  "volume": 5,  
  "item": "LEGO Blocks",  
  "creation_date": "10/18/2021"  
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	The response data will be returned
Failure	400 Bad Request	If the request is missing any of the 3 required attributes or required data validation, the boat is not created.

Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .
Failure	415 Unsupported Media Type	The content sent is not in the requested JSON format.

Response Examples

Success

Status: 201 Created

```
{
  "carrier": null,
  "creation_date": "10/18/2021",
  "id": 5693366486433792,
  "item": "LEGO Blocks",
  "owner": "auth0|627986546a0dc500692142f5",
  "self": "https://final-350500.uc.r.appspot.com/loads/5693366486433792",
  "volume": 5
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Status: 405 Method Not Allowed

```
{
  "Error": "Method not allowed"
}
```

Status: 406 Not Acceptable

```
{
```

```
"Error": "Client must accept valid JSON"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "Content must be sent as valid JSON"
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.
- Datastore will automatically generate an ID and store it with the entity being created.
- The “self” property will be automatically created but is not actually stored in the datastore.

Get a Load

Allows you to get an existing load.

GET /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

None

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	If the load is currently loaded on a boat, the carrier will be the JSON details of the boat. Else, carrier will be null. See the notes regarding "self."
Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	403 Forbidden	Attempted to access the entity that does not belong to the currently verified JWT user.
Failure	404 Not Found	No load with this load_id exists
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .

Response Examples

Success

Status: 200 OK

{

```
"carrier": null,  
"creation_date": "10/18/2021",  
"id": 5693366486433792,  
"item": "LEGO Blocks",  
"owner": "auth0|627986546a0dc500692142f5",  
"self": "https://final-350500.uc.r.appspot.com/loads/5693366486433792",  
"volume": 5  
}
```

Failure

Status: 401 Unauthorized

```
{  
  "code": "invalid_header",  
  "description": "Unable to parse authentication token."  
}
```

Status: 403 Forbidden

```
{  
  "Error": "This boat does not belong to you"  
}
```

Status: 404 Not Found

```
{  
  "Error": "No load with this load_id exists"  
}
```

Status: 405 Method Not Allowed

```
{  
  "Error": "Method not allowed"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.
- Datastore will automatically generate an ID and store it with the entity being created.
- The “self” property will be automatically created but is not actually stored in the datastore.

List all Loads

List all the loads - only the user's boats if authenticated correctly, else all loads. Both scenarios will have a pagination of 5 per page.

GET /loads

Request

Path Parameters

None

Request Body

None

Protected

No – if an authentication was verified correctly, then the response will be the user's boats. Else, all the boats on record.

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	No
Accept	application/json or */*	Client response accept values	Yes

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	All loads if a JWT fails to be verified or sent. Else, all loads owned by the authenticated user. The next attribute will only appear if there are more than 5 loads on the page – URL links to the next page. The total entries in the database will be displayed in the "total" attribute. Carrier will be null if the load is currently not loaded on a boat. Else, the ID of the id and name of the boat. See "Notes" regarding self.
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .

Response Examples

Success

Status: 200 OK

```
{
  "loads": [
    {
      "carrier": null,
      "creation_date": "10/18/2021",
      "id": 5662596023713792,
      "item": "Laptops",
      "owner": "auth0|627986546a0dc500692142f5",
      "self": "https://final-350500.uc.r.appspot.com/loads/5662596023713792",
      "volume": 10
    },
    {
      "carrier": null,
      "creation_date": "10/18/2022",
      "id": 5676496995745792,
      "item": "Toys",
      "owner": "auth0|627986546a0dc500692142f5",
      "self": "https://final-350500.uc.r.appspot.com/loads/5676496995745792",
      "volume": 30
    },
    {
      "carrier": null,
      "creation_date": "10/18/2021",
      "id": 5693366486433792,
      "item": "LEGO Blocks",
      "owner": "auth0|627986546a0dc500692142f5",
      "self": "https://final-350500.uc.r.appspot.com/loads/5693366486433792",
      "volume": 5
    },
    {
      "carrier": null,
      "creation_date": "10/18/2021",
      "id": 5727818231054336,
      "item": "Tea",
      "owner": "auth0|627986546a0dc500692142f5",
      "self": "https://final-350500.uc.r.appspot.com/loads/5727818231054336",
      "volume": 10
    },
    {
      "carrier": null,
      "creation_date": "10/18/2021",
      "id": 5746593479262208,
      "item": "Books",
```

```
"owner": "auth0|627986546a0dc500692142f5",  
  "self": "https://final-350500.uc.r.appspot.com/loads/5746593479262208",  
  "volume": 10  
},  
"next": "https://final-350500.uc.r.appspot.com/loads?limit=5&offset=5",  
"total": 6  
}
```

Failure

Status: 405 Method Not Allowed

```
{  
  "Error": "Method not allowed"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.
- Datastore will automatically generate an ID and store it with the entity being created.
- The “self” property will be automatically created but is not actually stored in the datastore.

Edit a Load - Patch

Allows you to edit some of the attributes of a load.

PATCH /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request Body Format

JSON

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes

Request JSON Attributes

At least one of these attributes must be within the request.

Name	Description	Type	Validation	Required?
item	The item on the load	String	Must be greater than 1 character length.	No
creation_date	The date the load was created.	String	Must be greater than or equal to 1 character length.	No
volume	The volume of the load	Integer	Must be greater than or equal to 1.	No

Request Body Example

```
{  
  "item": "Mega Blocks",  
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	The response data will be returned
Failure	400 Bad Request	If the request is missing any of the 3 required attributes or required data validation, the boat is not created.

Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	403 Forbidden	Attempted to access the entity that does not belong to the currently verified JWT user.
Failure	404 Not Found	No load with this load_id exists
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .
Failure	415 Unsupported Media Type	The content sent is not in the requested JSON format.

Response Examples

Success

Status: 201 Created

```
{
  "carrier": null,
  "creation_date": "10/18/2021",
  "id": 5693366486433792,
  "item": "Mega Blocks",
  "owner": "auth0|627986546a0dc500692142f5",
  "self": "https://final-350500.uc.r.appspot.com/loads/5693366486433792",
  "volume": 5
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Status: 403 Forbidden

```
{
  "Error": "This load does not belong to you"
}
```

Status: 404 Not Found

```
{  
  "Error": "No load with this load_id exists"  
}
```

Status: 405 Method Not Allowed

```
{  
  "Error": "Method not allowed"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Content must be sent as valid JSON"  
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.
- Datastore will automatically generate an ID and store it with the entity being created.
- The “self” property will be automatically created but is not actually stored in the datastore.

Edit a Load - Put

Allows you to edit all the attributes of a load.

POST /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request Body Format

JSON

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes

Request JSON Attributes

Name	Description	Type	Validation	Required?
item	The item on the load	String	Must be greater than 1 character length.	Yes
creation_date	The date the load was created.	String	Must be greater than or equal to 1 character length.	Yes
volume	The volume of the load	Integer	Must be greater than or equal to 1.	Yes

Request Body Example

```
{
  "volume": 50,
  "item": "LEGO Blocks0",
  "creation_date": "10/18/2020"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	The response data will be returned
Failure	400 Bad Request	If the request is missing any of the 3 required attributes or required data validation, the boat is not created.

Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	403 Forbidden	Attempted to access the entity that does not belong to the currently verified JWT user.
Failure	404 Not Found	No load with this load_id exists
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .
Failure	415 Unsupported Media Type	The content sent is not in the requested JSON format.

Response Examples

Success

Status: 201 Created

```
{
  "carrier": null,
  "creation_date": "10/18/2020",
  "id": 5693366486433792,
  "item": "LEGO Blocks0",
  "owner": "auth0|627986546a0dc500692142f5",
  "self": "https://final-350500.uc.r.appspot.com/loads/5693366486433792",
  "volume": 50
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Status: 403 Forbidden

```
{
  "Error": "This load does not belong to you"
}
```

Status: 404 Not Found

```
{  
  "Error": "No load with this load_id exists"  
}
```

Status: 405 Method Not Allowed

```
{  
  "Error": "Method not allowed"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Content must be sent as valid JSON"  
}
```

Notes

- See Authorization Errors for details regarding each status code 401 Unauthorized error.
- Datastore will automatically generate an ID and store it with the entity being created.
- The “self” property will be automatically created but is not actually stored in the datastore.

Delete a load

Allows you to delete a load. If the load being deleted has a boat, the boat has unloaded that load.

DELETE /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

None

Protected

Yes

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes
Accept	application/json or */*	Client response accept values	Yes (Due to error statuses)

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Failure to verify the JWT token provided in Authorization header or the JWT was not provided – message description will appear to indicate the type of error.
Failure	403 Forbidden	Attempted to access the entity that does not belong to the currently verified JWT user.
Failure	404 Not Found	No load with this load_id exists
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
  "code": "invalid_header",
  "description": "Unable to parse authentication token."
}
```

Status: 403 Forbidden

```
{
  "Error": "This load does not belong to you"
}
```

Status: 404 Not Found

```
{
  "Error": "No load with this load_id exists"
}
```

Status: 405 Method Not Allowed

```
{
  "Error": "Method not allowed"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Client must accept valid JSON"
}
```

Notes

See Authorization Errors for details regarding each status code 401 Unauthorized error.

List all Users

List all the users.

GET /users

Request

Path Parameters

None

Request Body

None

Protected

No

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	All users on record are shown their Datastore ID and Auth0 ID.
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .

Response Examples

Success

Status: 200 OK

```
{
  "users": [
    {
      "id": 5182123778506752,
      "user_id": "auth0|627986546a0dc500692142f5"
    },
    {
      "id": 5648810965467136,
      "user_id": "auth0|6279b0cf32d05600685f43aa"
    }
  ]
}
```

Failure

Status: 405 Method Not Allowed

```
{
  "Error": "Method not allowed"
}
```

```
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```

Notes

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

Login

Canonical login of a user.

POST /login

Request

Path Parameters

None

Request Body

JSON

Protected

No

Request JSON Attributes

Name	Description	Type	Validation	Required?
username	The email of the user	String	Standard email validation. Eg: 123@456.com	Yes
password	The password of the user	String	Must be at least 8 characters.	Yes

Request Body Example

```
{
  "username": "123@456.com",
  "password": "12345678"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Auth0 JWT information is rerouted back.
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .

Response Examples

Success

Status: 200 OK

```
{
  "access_token":
  "eyJhbGciOiJIc2EiLCJlbnMiOiJBMjU2R0NNIiwiaXNzIjoiaHR0cHM6Ly80OTMtYXVzLWWh3Ny51cy5hdXRoMjUybn0..vetO42W0mlaxD_Yp.IISxHp22fpw7hnoJE62lRXXqYrnWujc1lyYBguiLPqv-
  l_7Ozc4v9RfgAE6D8PQ2MnfwDvcT-2E_OqOsbHd2OXZJmK1FB4lBUHzlBMNsRwYU-
```


Register

Canonical register of a user.

POST /register

Request

Path Parameters

None

Request Body

JSON

Protected

No

Request JSON Attributes

Name	Description	Type	Validation	Required?
username	The email of the user	String	Standard email validation. Eg: 123@456.com	Yes
password	The password of the user	String	Must be at least 8 characters.	Yes

Request Body Example

```
{
  "username": "123@456.com",
  "password": "12345678"
}
```

Response

Response Body Format

Success: JSON

Failure: HTML

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Auth0 JWT information is rerouted back.
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.
Failure	406 Not Acceptable	Accept header is neither application/json nor */* .

Response Examples

Success

Status: 200 OK

```
{
  "access_token":
```

```
"eyJhbGciOiJIaXZlcjIjbmMiOiJBMjU2R0NNIiwiaXNzIjoiaHR0cHM6Ly80OTMtYXVzLWWh3Ny51cy5hdXRoM  
C5jb20vIn0..vetO42W0mlaxD_Yp.lISxHp22fpw7hnoJE62IRXXqYrnWujc1lyYBguiLPqv-  
l_7OzC4v9RfgAE6D8PQ2MnfwDvcT-2E_OqOsbHd2OXZJmK1FB4IBUHziBMNsrWYU-  
Zv6pbntkGUZfWcX0Zulxav9DkCOj4-4dlwRk1Z7fYYfP-  
NUh8Rhh8N7zGrQohgdqYN5KwDaqonnXMp61FH5f_vm9FUC02ykQjVc0sf4tR4Ojd0UmY9GZziZSvZsQyni  
q1m409OyU3hMQA7OJ3zUsb8qcaM4ATaxl3fAEfkYSvjwU-p9oY7t-FP-  
dEvuLL73UzF13yE8n0W60gXgtQ4NrF3QGfQA2VEnK6JpB5FN-  
gmrjOoXIYUfkVZclm4Xkw.VyyBwKV5vrsQMquGK2qoWg",  
  "id_token":  
  "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjhaGjPQ3pkTXhBZlZzdi05RVZaQyJ9.eyJuaWNrbmFtZSI6  
6InRlc3QiLCJuYW1lIjoiaGVzdEB0ZXN0LmNvbSIsInBpY3R1cmUiOiJodHRwczovL3MuZ3JhdmFOYXluY29tL2  
F2YXRhcnI9InJqYyQyMTdiMzRiMmU4ZDNiZDkxNWZjNjVjNDQ1Mj9zPTQ4MCZyPXBnJmQ9aHR0cHMIM0  
EIMkYIMkZjZG4uYXV0aDAuY29tJTJGYXZhdGFycyUyRnRlLnBuZyIsInVwZGF0ZWRfYXQiOiIyMDIyLTA1LTIOV  
DIyOjEwOjE5LjI5NVoiLCJlbWFpbCI6InRlc3RAdGVzdC5jb20iLCJlbWFpbF92ZXJpZmllZCI6ZmFsc2UsImIzcyI6  
Imh0dHBzOi8vNDkzLWF1cy1odzcudXMubWV0aDAuY29tLyIsInN1YiI6ImF1dGgwfdYyNzk4NjU0NmEwZGM  
1MDA2OTIxNDJmNSIsImF1ZCI6InNTMmNzcHcwYnNudGh0fFk2RkdrcExb3pZWJRVjBGliwiaWF0IjojNj  
UzNDMmWmJjE5LCJleHAiOiE2NTM0NjYyMTI9.CsWNN4ly44rbT4B8DSkp3FtCneiscWzw2CwoiovmQGdrsY9  
2BCA_skkX6fSwHe1zYYb8p4xOoXi4htOQl6vIHQsUos-jFKiS7yAaAbrEpz-l9Za54UG48gJT-  
mthf2nsFxxW_GDstkplUKVZG9GlcRhA59HcvpVKiDhYq74-  
nOUyYdc59B0akNlFSDI32ltsrjcY1fumGHLtOo7zhdN_hAbQNrXP4PO8GGGyF2H0skcae2nG-6rMiFGsT9z-  
LRbzaruakhqS_JhJb-  
UjP4EVftsL60EQyHx9Aen4RYzmgFHRMFi8b1NeCdHRQj20ZfZO9ogvTORGZYxQNBALFHphg",  
  "scope": "openid profile email address phone",  
  "expires_in": 86400,  
  "token_type": "Bearer"  
}
```

Failure

Status: 405 Method Not Allowed

Method Not Allowed

The method is not allowed for the requested URL.

Status: 406 Not Acceptable

```
{  
  "Error": "Client must accept valid JSON"  
}
```


User Information

Shows the user information – note that this is an ephemeral endpoint and can only be shown once per login or registration. It is best to access this page through a web browser. This endpoint uses a temporary session cookie with the JWT and is deleted the moment this page is accessed.

GET /user-info

Request

Path Parameters

None

Request Body

None

Protected

No

Response

Response Body Format

Success: JSON

Failure: HTML

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Datastore ID, Auth0 JWT, and Auth0 user id (used as the “owner” in non-user entities) is returned back.
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.

Response Examples

Success

Status: 200 OK

```
{
  "data_store_id": 5648810965467136,
  "jwt_id_token":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjhhXGJpQ3pkTXhBZlZzdj05RVZaQyJ9.eyJuaWNrbmFtZSI6IjEiLCJuYVlwYWRjaWoiOiMUBlbWFpbC5jb20iLCJwaWN0dXllIjoiaHR0cHM6Ly9zMldyYXZhdGFyLmNvbS9hdmFOYXlvYjM1ZGM4ODMyZjAwNTNmYjY3NG0yYjBhNGQ5MTZjZTQ_cz00ODAmcj1wZyZkPWh0dHBzJTBJTGJGTJGY2RuLmF1dGgwLmNvbSUyRmF2YXRhcmlMIMkYxLnBuZylslnVwZGF0ZWRFYXQiOiilyMDIyLTU0VDlyOjIwOjExLjE5MVoiLCJlbWFpbC5jb20iLCJFAZW1haWwuyY29tliwiZW1haWxfdmVyaWZpZWQiOmZhbnHNILCJpc3MiOiJodHRwcovLzQ5My1hdXMtaHc3LnVzLmF1dGgwLmNvbS8iLCJzdWliOiJhdXR0MHw2Mjc5YjBjZjMyZDA1NjAwNjg1ZjQzYWFiLCJhdWQiOiZlUzljc3B3MGJzbncRocThZnkZHh3JBMBW96VVhSUUVywrRiIsImhdCl6MTY1MzQzMDgxMSwiZXhwIjoxNjUzNDY2ODEfXQ.O4I54bCB7EZKbNNJVsd38e2D07MdTDuF_GLu_mxwg0dGFEWUN6e0MVgX9yWRe1bul4lf1CXLrm8vjZeVKPvTgfE3P42BjjtSm-msALNmfraKf2HXD1KaHYXIyNIJH5eyK6Jq3OgGHE-I1pA47xRBbk1b9ey39SweOlajJIRpLAwhRxhafC6okxQv6VVKrsmR-
```

```
FGppBlxwWnkcAu4KcMkbcKdiezs_i4j_cvbez3XOAicDZ-TZ8fF5Y-VeJht3l4s8iLL7Bs8eKEOnStL6cjm60-  
wPPwJlpb51A2Fngaroh3hw5g_Cz_p5yCqAKTi1YvocKc1Ik4DIUiuWrdhDw",  
  "user_id": "auth0|6279b0cf32d05600685f43aa"  
}
```

Failure

Status: 405 Method Not Allowed

Method Not Allowed

The method is not allowed for the requested URL.

Decode

Shows the decoded user Auth0 JWT information.

GET /decode

Request

Path Parameters

None

Request Body

None

Protected

No

Request Header

Header	Value	Description	Required?
Authorization	Bearer JWT	Auth0's "id_token" as JWT	Yes

Response

Response Body Format

Success: JSON

Failure: HTML

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Auth0 JWT information is rerouted back.
Failure	405 Method Not Allowed	Invalid REST method used on this endpoint.

Response Examples

Success

Status: 200 OK

```
{
  "aud": "sS2cspw0bsnthq8Y6FGkrA1ozYXRQV0F",
  "email": "test@test.com",
  "email_verified": false,
  "exp": 1653466219,
  "iat": 1653430219,
  "iss": "https://493-aus-hw7.us.auth0.com/",
  "name": "test@test.com",
  "nickname": "test",
  "picture":
    "https://s.gravatar.com/avatar/b642b4217b34b1e8d3bd915fc65c4452?s=480&r=pg&d=https%3A%2F%
    2Fcdn.auth0.com%2Favatars%2Fte.png",
  "sub": "auth0|627986546a0dc500692142f5",
  "updated_at": "2022-05-24T22:10:19.295Z"
```

}

Failure

Status: 405 Method Not Allowed

Method Not Allowed

The method is not allowed for the requested URL.