https://finaproject-sadats-460pm.uw.r.appspot.com

Final Project: API Spec

CS 493: Cloud Application Development
Fall 2022
Oregon State University
Last Update: Oct 6. 11:15 am Pacific

https://finaproject-sadats-460pm.uw.r.appspot.com

Data Model

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

Users

Property	Data Type	Required	Notes
id	Integer	No	The id of the user. Datastore automatically generates it. Don't add it yourself as a property of the entity.
firstName	String	Yes	First Name of User.
lastName	String	Yes	Last Name of User.
subId	Integer	Yes	User Id for authentication.
Self	String	No	Dynamically generated self-link

Boats

Property	Data Type	Required	Notes
id	Integer	No	The id of the boat. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	Yes	Name of the boat.
type	String	Yes	Type of the boat. E.g., Sailboat, Catamaran, etc.
length	Integer	Yes	The length of the boat in feet.
owner	string	No	The subid from users, signifying the owner.
Loads	Array of [Key value pair]	No	Includes id and self-link to the load
Self	String	No	Dynamically generated self-link

Load

Property	Data Type	Required	Notes
id	Integer	No	The id of the slip. Datastore automatically generates it. Don't add it yourself as a property of the entity.
volume	Integer	Yes	The volume of the load.
carrier	Object{key :value}	No	The boats carrying the load includes, boats id, name, and self-link
item	String	Yes	Load being carried name
creation date	String	Yes	Date the load was created
Self	String	No	Dynamically generated self-link

Relationships

Boats and Loads

- All newly created loads should begin unassigned to any boat.
- Deleting a load should update the boat that was carrying it.

- It should be possible to assign a load to a boat.
- If a load is already assigned to one boat and then is assigned to another boat without first being removed, it should return a 403 status code.
- It should be possible to remove a load from a boat without deleting that load.
- Deleting a boat will unload any loads that were loaded on to it.

Users

- A valid google user who logs in the first time will be added to the datastore with their sub from the jwt token serving as their identifier.
- Unique identifier is the sub value from the jwt token.
- For requests that are protected a valid jwt token needs to be provided and if any resources exist for that user, then they will have access to those resources.
- A user who adds a boat will be the owner of the boat entity. The boat entity will have a property with the unique subid of the user and only that user will be able to update and delete the boat.

Get Users

Allows you to get a list of all the users.

Not Protected

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK

```
[
    {
        "name": "Sayeed",
        "subid": "100612745932206674450",
        "lastname": "Sadat",
        "id": 5153611168350208,
        "self": "http://localhost:8080/users/5153611168350208"
    },
    {
        "subid": "116789266201689881306",
        "name": "Sayeed Naser",
        "lastname": "Sadat",
        "id": 5673082664517632,
        "self": "http://localhost:8080/users/5673082664517632"
    }
]
```

Create a Boat

Allows you to create a new boat.

Protected

POST /boats

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	Yes
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Length of the boat in feet.	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	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code must be returned. You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid. You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).
Failure	415 invalid content type	Content Type must be Json
Failure	406 Invalid content type	Request content type in format not support besides json

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

```
Status: 201 Created

{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "self": "https://appspot.com/boats/123"
}
```

Failure

```
Status: 400 Bad Request

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

```
Status: 415 invalid content type

{
"Error":  "Server only accepts application/json data."
}
```

```
Status: 406 invalid accepts type

{
"Error":  "Not acceptable, response is in application/json only."
}
```

Get a Boat

Allows you to get an existing boat.

Protected

GET /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Invalid content type	Request content type in format not support besides json

Response Examples

Success

```
Status: 200 OK

{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "self": "https://appspot.com/boats/123"
}
```

Failure

```
Status: 404 Not Found

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

```
Status: 406 invalid accepts type

{
"Error":  "Not acceptable, response is in application/json only."
}
```

List all Boats

List all the boats, five at a time with pagination.

Protected

GET /boats

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Invalid content type	Request content type in format not support besides json

Response Examples

Success

Status: 200 OK

```
[
{
            "loads": [],
            "type": "Sailing Boat",
            "name": "Calypso",
            "length": 13,
            "id": 5464488283209728
        },
        {
            "type": "Yacht",
            "name": "401K",
            "loads": [],
            "length": 18,
            "id": 5635008819625984
        },
        {
            "loads": [
                {
                    "id": 5712837116690432
                },
                {
                    "id": 5090685099180032
                },
                {
                    "id": 5638358357245952
                },
                {
                    "id": 5092424393162752
                }
            ],
            "name": "Lucifer",
            "length": 15,
            "type": "Yacht",
            "id": 5655374346584064
        }
    ],
    "next":
"https://appspot.com/boats?cursor=Ci8SKWoUenV3fmh3NC1zYWRhdHMtNDE1cG1yEQsSBEJvYXQYgIC
AmIfxhQoMGAAgAA=="
}
```

Failure

Status: 406 invalid accepts type

```
{
"Error":  "Not acceptable, response is in application/json only."
}
```

Patch a Boat

Allows you to update a boat without providing all values.

Protected

Patch /boats/:boat_id

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	No
type	The type of the boat.	No
length	Length of the boat in feet.	No

Request Body Example

```
{
  "name": "Odyssey III",
  "type": "Yatch",
  "length": 99
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200	
Failure	403 unauthorized user	Unauthorized access denied, boat doesn’t belong to user.
Failure	404 invalid id	No boat with this id exists.
Failure	415 invalid content type	Content Type must be Json
Failure	406 Invalid content type	Request content type in format not support besides json

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

```
Status: 200 updated

{
  "id": 123,
  "name": "Odyssey",
  "type": "Yatch",
  "length": 99,
  "self": "https://appspot.com/boats/123"
}
```

Failure

Status: 403 Unauthorized

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

Status: 404 Not Found

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

Status: 406 invalid accepts type

```
{
"Error":  "Not acceptable, response is in application/json only."
}
```

Status: 415 invalid content type

```
{
"Error":  "Server only accepts application/json data."
}
```

Put a Boat

Allows you to update a boat, requires all attributes

Put /boats/:boat_id

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

https://finaproject-sadats-460pm.uw.r.appspot.com

Request JSON Attributes

Name	Description	Required?
name	The name of the boat. Must be Unique Name : must be mix of upper- and lower-case characters With/without spaces between 5-20 characters.	Yes
type	The type of the boat. E.g., Sailboat, Catamaran, etc. Type : must be mix of upper- and lower-case characters With/without spaces between 5-20 characters.	Yes
length	Length of the boat in feet. Length : must be integer between 10-1000	Yes

Request Body Example

```
{
  "name": "Odyssey III",
  "type": "Yatch",
  "length": 99
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 updated	
Failure	400 invalid attributes	Invalid attributes Name : must be mix of upper- and lower-case characters With/without spaces between 5-20 characters. Type : must be mix of upper- and lower-case characters With/without spaces between 5-20 characters. Length : must be integer between 10-1000
Failure	403 not unique name	Boat name must be unique
Failure	404 incorrect id	Boat with this id doesn’t exist

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

Status: 200 updated

```
{
  "id": 123,
  "name": "Odyssey",
  "type": "Yatch",
  "length": 99,
  "self": "https://appspot.com/boats/123"
}
```

Failure

Status: 404 Bad Request

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

Status: 400 invalid attributes

```
{
"Error":  "The request is invalid due to invalid attributes."
}
```

Status: 403 not unique name

```
{
"Error":  "The boat name is not unique."
}
```

Delete a Boat

Allows you to delete a boat. Note that if the boat is currently in loaded, then it unloads the loads and updates the load carrier and then deletes the boat.

```
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

Response Statuses

Outcome	**Status Code**	**Notes**
Success	204 No Content	
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

https://finaproject-sadats-460pm.uw.r.appspot.com

Add a Load

Allows you to create a new Load.

POST /loads

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Yes
item	Load being carried name	Yes
Creation_date	Date the load was created	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	
Failure	400 Bad Request	If the request is missing the number attribute, the slip must not be created, and 400 status code must be returned. You don't need to validate the values of this attribute and can assume that if the number attribute is specified, then its value is valid. You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute other than number).

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. This value needs to be sent in the response body as shown in the example.

Success

```
Status: 201 Created
```

```
{
   "id": 5631671361601536,
    "volume": 5,
    "carrier": null,
    "item": "LEGO Blocks",
    "creation_date": "10/18/2021",
    "self": " https://appspot.com/loads/5631671361601536"
}
```

Failure

```
Status: 400 Bad Request

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

https://finaproject-sadats-460pm.uw.r.appspot.com

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

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No load with this load_id exists

Response Examples

Success

```
Status: 200 OK

{
   "id": 5631671361601536,
    "volume": 5,
    "carrier": null,
    "item": "LEGO Blocks",
    "creation_date": "10/18/2021",
    "self": " https://appspot.com/loads/5631671361601536"
}
```

Failure

```
Status: 404 Not Found

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

List all Loads

List all the loads. Returns 3 at a time with a cursor

GET /loads

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

```
Status: 200 OK

[
{
            "creation_date": "10/19/2021",
            "item": "Stuffed Animals",
            "carrier": {
                "name": "Lucifer",
                "id": 5655374346584064,
                "self": "https://appspot.com/boats/5655374346584064"
            },
            "volume": 10,
            "id": 5712837116690432,
            "self": "https://appspot.com/loads/5712837116690432"
        },
        {
            "carrier": {
                "id": 5655374346584064,
                "name": "Lucifer",
                "self": "https://appspot.com/boats/5655374346584064"
            },
            "creation_date": "10/19/2021",
            "item": "Food",
            "volume": 3,
            "id": 5090685099180032,
            "self": "https://appspot.com/loads/5090685099180032"
        }
],
```

https://finaproject-sadats-460pm.uw.r.appspot.com

```
"next":
"https://appspot.com/loads?cursor=Ci8SKWoUenV3fmh3NC1zYWRhdHMtNDE1cG1yEQsSBExvYWQYgICA2OmBggoMGAAgAA=="
```

Patch a Load

Allows you to update a boat without providing all values

Patch /loads/:load_id

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat. Must be Unique Name : must be mix of upper- and lower-case characters With/without spaces between 5-20 characters.	No
type	The type of the boat. E.g., Sailboat, Catamaran, etc. Type : must be mix of upper- and lower-case characters With/without spaces between 5-20 characters.	No
length	Length of the boat in feet. Length : must be integer between 10-1000	No

Request Body Example

```
{
  "name": "Odyssey III",
  "type": "Yatch",
  "length": 99
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303updated	Location header contains the object location
Failure	400 missing attributes	All attributes are required
Failure	400 invalid attributes	Invalid attributes Name : must be mix of upper- and lower-case characters With/without spaces between 5-20 characters. Type : must be mix of upper- and lower-case characters With/without spaces between 5-20 characters. Length : must be integer between 10-1000
Failure	403 not unique name	Boat name must be unique
Failure	404 incorrect id	Boat with this id doesn’t exist

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

```
Status: 303 updated

{
  "id": 123,
  "name": "Odyssey",
  "type": "Yatch",
  "length": 99,
  "self": "https://appspot.com/boats/123"
}
```

Failure

```
Status: 400 Bad Request

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

```
Status: 404 Bad Request

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

```
Status: 400 invalid attributes

{
"Error":  "The request is invalid due to invalid attributes."
}
```

```
Status: 403 not unique name

{
"Error":  "The boat name is not unique."
}
```

Put a Load

Allows you to update a boat, requires all attributes

```
Put /loads/:load_id
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	Yes
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Length of the boat in feet.	Yes

Request Body Example

```
{
  "name": "Odyssey III",
  "type": "Yatch",
  "length": 99
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 updated	
Failure	404 incorrect id	Boat with this id doesn't exist

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

```
Status: 200 updated

{
  "id": 123,
  "name": "Odyssey",
  "type": "Yatch",
  "length": 99,
  "self": "https://appspot.com/boats/123"
}
```

Failure

```
Status: 404 Bad Request
```

{
"Error": "No boat with this id exists"
}

Status: 400 invalid attributes
 	
{
"Error": "The request is invalid due to invalid attributes."
}

Status: 403 not unique name
 	
{
"Error": "The boat name is not unique."
}

Delete a Load

Allows you to delete a load, if the load is assigned to a boat, it is first removed from the boat.

DELETE /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

None

Response

No body

https://finaproject-sadats-460pm.uw.r.appspot.com

Response Body Format

Success: No body
Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No load with this load_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

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

Put Load on Boat

Adding a load to a boat.

```
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

Note: Set Content-Length to 0 in your request when calling out to this endpoint.

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists that has not been assigned to another ship. Updates the load with the boat information and adds the load to the ship loads.
Failure	403 Forbidden	The load is already load onto another ship.
Failure	404 Not Found	The specified boat and/or load does not exist

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 403 Forbidden

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

```
Status: 404 Not Found

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

Comment

- A load can be assigned to only one ship and a ship can carry multiple loads.

Remove Load from Boat

Removes load from the boat and updates the load to remove the boat from carrier.

```
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

Response

No body

Response Body Format

Success: No body
Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and the load is on the boat.
Failure	404 Not Found	No boat with this boat_id and no load with this load_id exists and or are assigned to each other.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found
{
"Error":  " No boat with this boat_id is loaded with the load with this load_id"
}
```