

Final Project: API Spec

CS 493: Cloud Application Development

Casey Cheek

Fall 2021

Oregon State University

Last Update: Dec 3, 2021. 11:27 pm Pacific

Change log.....	2
Application URL.....	2
Create an Auth0 User Account/User Login.....	2
Data Model	3
Get a User	5
List all Users	6
Create a Boat	7
Get a Boat	9
Update a Boat - Partial.....	11
Update a Boat - Full	13
List all Boats	15
List all Loads for a Boat	17
Delete a Boat.....	19
Create a Load	20
Get a load.....	22
Update a Load - Partial.....	23
Update a Load - Full	25
List all Loads	27
Delete a Load	29
Assign a Load to a Boat	30
Remove a Load from a Boat.....	32
Error 401 Response Examples.....	34
Error 405 Response Example	34

Change log

Version	Change	Date
1.0	Initial version.	Dec 3, 2021

Application URL

<https://final-project-cheekc.uw.r.appspot.com>

Note: This URL leads to the Welcome page where users may login/create an account with Auth0. This is also the base URL where the application is deployed on GCP.

Create an Auth0 User Account/User Login

Allows you to create a new Auth0 user account and/or login with your username and password. **You must login through this webapp first to register your user id with this API, and thus access the protected resource, “Boats”.**

Instructions

1. Go to the application URL: <https://final-project-cheekc.uw.r.appspot.com>
2. Click the button that says “Click here to get a token”
3. Follow the instructions on screen to login or create an Auth0 account
4. Once you’re logged in, you’ll be redirected to the user info page which will display your user ID and your JWT.
5. Your user ID is now registered in this app. You can create, read, update, and delete boats that are associated with your user ID by including your JWT in the header of your requests.

Data Model

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

Users

Users can request protected resources by including their JWT as a bearer token in the Authorization header. Each JWT contains a unique Auth0 ID associated with user. After verifying/decoding the JWT in the request, this application can determine if that ID matches a boat's owner.

Property	Data Type	Notes
id	String	The user's auth0 unique ID.

Boats

Boats are considered a protected resource. When a user creates a boat, their user id (contained within their JWT) is set as the boat's "owner" attribute. Only the boat's owner may view, modify, or delete their boat. They are also the only one allowed to assign or remove loads from their boat. When loads are assigned to a boat, each load's id and self link are recorded in the boat's "loads" attribute.

Property	Data Type	Notes
id	Integer	The id of the boat. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	Name of the boat.
type	String	Type of the boat. E.g., Sailboat, Catamaran, etc.
length	Integer	The length of the boat in feet.
loads	List of Objects	None if the boat has no loads assigned to it. Otherwise, each object in the list represents a load. New boats are created without any loads assigned to them.
loads[index]:id	Integer	The id of the load.
loads[index]:self	String	The self link to the load.
owner	String	The user's auth0 unique ID.

Loads

Loads are an unprotected resource. A load may be created, viewed, updated, or deleted by anyone. When a load is assigned to a boat, the "carrier" attribute of that load will contain the id, name, and self link of that boat.

Property	Data Type	Notes
id	Integer	The id of the load. Datastore automatically generates it. Don't add it yourself as a property of the entity.
volume	Integer	The volume of the load.
carrier	Object	Null if the load is not assigned to a boat. Otherwise, the object represents the load's assigned boat. New loads are created without an assigned boat.
carrier:id	Integer	The id of the boat.
carrier:name	String	The name of the boat.
carrier:self	String	The self link to the boat.

content	String	Description of the load's contents.
creation_date	String	Date the load was created in mm/dd/yyyy format.

Get a User

Allows you to get an existing user.

GET /users/:user_id

Request

Path Parameters

Name	Description
user_id	ID of the user

Request Authorization Header

None required

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No user with this user_id exists
Failure	406 Not Acceptable	Client doesn't accept JSON

Response Examples

Success

```
Status: 200 OK

{
  "id": 123abc,
  "self": "https://final-project-cheekc.uw.r.appspot.com/users/123abc"
}
```

Failure

```
Status: 404 Not Found

{
  "Error": "Not Found. No user with this user_id exists."
}
```

```
Status: 406 Not Acceptable

{
  "Error": "Not Acceptable. The accepted MIME type in the request is unsupported."
}
```

List all Users

List all the users.

GET /users

Request

Path Parameters

None

Request Authorization Header

None required

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	Client doesn't accept JSON

Response Examples

Success

Status: 200 OK

```
{
  [
    {
      "id": 123abc,
      "self": "https://final-project-cheekc.uw.r.appspot.com/users/123abc"
    },
    {
      "id": 456efg,
      "self": "https://final-project-cheekc.uw.r.appspot.com/users/456efg"
    }
  ]
}
```

Failure

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable. The accepted MIME type in the request is unsupported."
}
```

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Path Parameters

None

Request Authorization Header

Bearer JWT Required

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.
Failure	401 Unauthorized	Credentials are bad or missing
Failure	406 Not Acceptable	Client doesn't accept JSON

Response Examples

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

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. In other words, this is the URL to get this newly created boat. The `self` attribute is not stored in Datastore
- Any additional attributes included in the body will not be added.

Success

Status: 201 Created

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": null,
  "self": "https://final-project-cheekc.uw.r.appspot.com/boats/123"
}
```

Failure

Status: 400 Bad Request

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

Status 401: See last page for Status 401 Response Examples

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable. The accepted MIME type in the request is unsupported."
}
```


Get a Boat

Allows you to get an existing boat. Only the boat's owner may view their boat.

GET /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Authorization Header

Bearer JWT Required

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Credentials are bad or missing
Failure	403 Forbidden	User is not the owner of this boat
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	Client doesn't accept JSON

Response Examples

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": null,
  "self": "https://final-project-cheekc.uw.r.appspot.com/boats/123"
}
```

Failure

Status 401: See last page for Status 401 Response Examples

Status: 403 Forbidden

```
{
  "Error": "Forbidden. This user cannot access this boat."
}
```

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{  
  "Error": "Not Acceptable. The accepted MIME type in the request is unsupported."  
}
```

Update a Boat - Partial

Allows you to update one or more attributes of your boat. Only the name, type, or length may be changed from this endpoint. Any attribute excluded from the body will remain unchanged. Only the boat's owner may update their boat.

PATCH /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Authorization Header

Bearer JWT Required

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. E.g., Sailboat, Catamaran, etc.	No
length	Length of the boat in feet.	No

Request Body Example

```
{
  "length": 59
}
```

Response

Response Body Format

Success: No body, Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Credentials are bad or missing
Failure	403 Forbidden	User is not the owner of this boat
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status 401: See last page for Status 401 Response Examples
Status: 403 Forbidden { "Error": "Forbidden. This user cannot access this boat." }
Status: 404 Not Found { "Error": "Not found. No boat with this boat_id exists." }

Update a Boat - Full

Allows you to update one or more attributes of your boat. Only the name, type, or length may be changed from this endpoint. Only the boat's owner may update their boat.

PUT /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Authorization Header

Bearer JWT Required

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

```
{
  "length": 4,
  "name": "Carpe Diem",
  "type": "Rowboat"
}
```

Response

Response Body Format

Success: No body, Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat will not be created.
Failure	401 Unauthorized	Credentials are bad or missing
Failure	403 Forbidden	User is not the owner of this boat
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 400 Bad Request

<pre>{ "Error": "Bad Request. The request object is missing at least one of the required attributes." }</pre>

Status 401: See last page for Status 401 Response Examples
--

Status: 403 Forbidden

<pre>{ "Error": "Forbidden. This user cannot access this boat." }</pre>

Status: 404 Not Found

<pre>{ "Error": "Not found. No boat with this boat_id exists." }</pre>
--

List all Boats

List all the boats owned by the current user.

GET /boats

Request

Path Parameters

None

Request Authorization Header

Bearer JWT Required

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Response will only show 5 boats at a time. If more results exist, the 'next' link will point to the next set, otherwise there will not be a 'next' attribute.
Failure	401 Unauthorized	Credentials are bad or missing
Failure	406 Not Acceptable	Client doesn't accept JSON

Response Examples

Success

Status: 200 OK

```
{
  "Total number of boats for this user": 6,
  "boats": [
    {
      "id": 123,
      "name": "Sea Witch",
      "type": "Catamaran",
      "length": 28,
      "loads": null,
      "self": "https://final-project-cheekc.uw.r.appspot.com/boats/123"
    },
    {
      "id": 321,
      "name": "Titanic",
      "type": "Cruise Ship",
      "length": 200,
      "loads": null,
      "self": "https://final-project-cheekc.uw.r.appspot.com/boats/321"
    },
  ],
}
```

```

{
  "id": 999,
  "name": "Dolphin",
  "type": "Sloop",
  "length": 21,
  "loads": null
  "self": "https://final-project-cheekc.uw.r.appspot.com/boats/999"
},
{
  "id": 456,
  "name": "Adventure",
  "type": "Sailboat",
  "length": 50,
  "loads": [
    {
      "id": 8765,
      "self": "https://final-project-cheekc.uw.r.appspot.com/loads/8765"
    }
  ],
  "self": "https://final-project-cheekc.uw.r.appspot.com/boats/456"
},
{
  "id": 789,
  "name": "Hocus Pocus",
  "type": "Sailboat",
  "length": 100,
  "loads": [
    {
      "id": 7418,
      "self": "https://final-project-cheekc.uw.r.appspot.com/loads/7418"
    },
    {
      "id": 8989,
      "self": "https://final-project-cheekc.uw.r.appspot.com/loads/8989"
    }
  ],
  "self": "https://final-project-cheekc.uw.r.appspot.com/boats/789"
} ],
"next": "https://final-project-cheekc.uw.r.appspot.com/boats?limit=5&offset=5"
}

```

Failure

Status 401: See last page for Status 401 Response Examples

Status: 406 Not Acceptable

```

{
  "Error": "Not Acceptable. The accepted MIME type in the request is unsupported."
}

```


List all Loads for a Boat

Allows you to view the loads assigned to a boat. Only the owner of a boat may view their boat's loads at this endpoint.

GET /boats/:boat_id/loads

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Authorization Header

Bearer JWT Required

Request Body

None

Request Body Format

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Credentials are bad or missing
Failure	403 Forbidden	User is not the owner of this boat
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	Client doesn't accept JSON

Response Examples

Success

Status: 200 OK

```
{[
  {
    "id": 1122,
    "self": "https://final-project-cheekc.uw.r.appspot.com/loads/1122"
  },
  {
    "id": 3344,
    "self": "https://final-project-cheekc.uw.r.appspot.com/loads/3344"
  },
]}
```

Failure

Status 401: See last page for Status 401 Response Examples

Status: 403 Forbidden { "Error": "Forbidden. This user cannot access this boat." }
Status: 404 Not Found { "Error": "No boat with this boat_id exists" }
Status: 406 Not Acceptable { "Error": "Not Acceptable. The accepted MIME type in the request is unsupported." }

Delete a Boat

Allows you to delete a boat. Deleting a boat also unloads any loads that were assigned to it.

DELETE /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Authorization Header

Bearer JWT Required

Request Body

None

Response

No body

Response Body Format

Success: No body, Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	If loads were assigned to the boat, this action will remove the boat as the carrier of each load.
Failure	401 Unauthorized	Credentials are bad or missing
Failure	403 Forbidden	User is not the owner of this boat
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status 401: See last page for Status 401 Response Examples

Status: 403 Forbidden

```
{
  "Error": "Forbidden. This user cannot access this boat."
}
```

Status: 404 Not Found

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

Create a Load

Allows you to create a new load.

POST /loads

Request

Path Parameters

None

Request Authorization Header

None required

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load, an integer	Yes
content	The content of the load, a string	Yes
creation_date	The date the load was created, a string in mm/dd/yyyy format	Yes

Request Body Example

```
{
  "volume": 5,
  "content": "LEGO Blocks",
  "creation_date": "10/18/2020"
}
```

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 load must not be created, and 400 status code must be returned.
Failure	406 Not Acceptable	Client doesn't accept JSON

Response Examples

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

- The `self` attribute will contain the live link to the REST resource corresponding to this load. In other words, this is the URL to get this newly created load. The `self` attribute is not stored in Datastore.
- Any additional attributes included in the body will not be added.

Success

Status: 201 Created

```
{
  "id": 5678,
  "volume": 5,
  "carrier": null,
  "content": "LEGO Blocks",
  "creation_date": "10/18/2020"
  "self": "https://final-project-cheekc.uw.r.appspot.com/loads/5678"
}
```

Failure

Status: 400 Bad Request

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

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable. The accepted MIME type in the request is unsupported."
}
```

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 Authorization Header

None required

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
Failure	406 Not Acceptable	Client doesn't accept JSON

Response Examples

Success

Status: 200 OK

```
{
  "id": 5678,
  "volume": 5,
  "carrier": null,
  "content": "LEGO Blocks",
  "creation_date": "10/18/2020"
  "self": "https://final-project-cheekc.uw.r.appspot.com/loads/5678"
}
```

Failure

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable. The accepted MIME type in the request is unsupported."
}
```

Update a Load - Partial

Allows you to update one or more attributes of a load. Only the volume, content, and creation_date may be changed from this endpoint. Any attribute excluded from the body will remain unchanged.

PATCH /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Authorization Header

None Required

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load, an integer	No
content	The content of the load, a string	No
creation_date	The date the load was created, a string in mm/dd/yyyy format	No

Request Body Example

```
{
  "volume": 50,
  "content": "peaches"
}
```

Response

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": "Not found. No load with this load_id exists."  
}
```


Update a Load - Full

Allows you to update all attributes of a load. Only the volume, content, and creation_date may be changed from this endpoint.

PUT /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Authorization Header

None Required

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load, an integer	Yes
content	The content of the load, a string	Yes
creation_date	The date the load was created, a string in mm/dd/yyyy format	Yes

Request Body Example

```
{
  "volume": 50,
  "content": "peaches",
  "creation_date": "05/12/2005"
}
```

Response

Response Body Format

Success: No body, Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the load will not be created.
Failure	404 Not Found	No load with this load_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 400 Bad Request

<pre>{ "Error": "Bad Request. The request object is missing at least one of the required attributes." }</pre>

Status: 404 Not Found

<pre>{ "Error": "Not found. No load with this load_id exists." }</pre>
--

List all Loads

List all the loads.

GET /loads

Request

Path Parameters

None

Request Authorization Header

None required

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Response will only show 5 loads at a time. If more results exist, the 'next' link will point to the next set, otherwise there will not be a 'next' attribute.
Failure	406 Not Acceptable	Client doesn't accept JSON

Response Examples

Success

Status: 200 OK

```
{
  "Total number of loads": 6,
  "loads": [
    {
      "id": 5678,
      "volume": 5,
      "carrier": null,
      "content": "LEGO Blocks",
      "creation_date": "10/18/2020",
      "self": "https://final-project-cheekc.uw.r.appspot.com/loads/5678"},
    {
      "id": 8765,
      "volume": 10,
      "carrier": {
        "id": 456,
        "name": "Adventure",
        "self": "https://final-project-cheekc.uw.r.appspot.com/boats/456"
      },
      "content": "Mammoth Ivory",
      "creation_date": "10/20/2020"
    }
  ]
}
```

```
"self": "https://final-project-cheekc.uw.r.appspot.com/loads/8765"},
{
  "id": 7418,
  "volume": 15,
  "carrier": {
    "id": 789,
    "name": "Hocus Pocus",
    "self": "https://final-project-cheekc.uw.r.appspot.com/boats/789"
  },
  "content": "Twinkies",
  "creation_date": "12/04/2020",
  "self": "https://final-project-cheekc.uw.r.appspot.com/loads/7418"
} ],
"next": "https://final-project-cheekc.uw.r.appspot.com/loads?limit=5&offset=5"
}
```

Failure

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable. The accepted MIME type in the request is unsupported."
}
```

Delete a Load

Allows you to delete a load. Deleting a load also removes its record from the boat carrying it (if any).

DELETE /load/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Authorization Header

None required

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	If the load was assigned to a boat, this action will remove the load from that boat's loads.
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"
}
```

Assign a Load to a Boat

Assigns a load to a carrier boat. A load cannot be assigned to multiple boats.

PUT /boats/:boat_id/loads/:load_id

Request

Path Parameters

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

Request Authorization Header

Bearer JWT Required

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 currently unassigned to a carrier boat.
Failure	401 Unauthorized	Credentials are bad or missing
Failure	403 Forbidden	The load is already assigned to a boat, or the user does not own this boat.
Failure	404 Not Found	The specified boat and/or load does not exist

Response Examples

Success

Status: 204 No Content

Failure

Status 401: See last page for Status 401 Response Examples

Status: 403 Forbidden

```
{
  "Error": "This load already has a carrier boat"
}
```

Status: 403 Forbidden

```
{
  "Error": "Forbidden. This user cannot access this boat."
}
```

Status: 404 Not Found

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

Remove a Load from a Boat

Removes a load from a carrier boat without deleting the load.

DELETE /boats/:boat_id/loads/:load_id

Request

Path Parameters

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

Request Authorization Header

Bearer JWT Required

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 assigned to this boat.
Failure	401 Unauthorized	Credentials are bad or missing
Failure	403 Forbidden	The boat is not carrying this load, or the user does not own this boat.
Failure	404 Not Found	The specified boat and/or load does not exist.

Response Examples

Success

Status: 204 No Content

Failure

Status 401: See last page for Status 401 Response Examples

Status: 403 Forbidden

```
{
  "Error": "This boat is not carrying this load"
}
```

Status: 403 Forbidden

```
{
  "Error": "Forbidden. This user cannot access this boat."
}
```


Status: 404 Not Found

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

Error 401 Response Examples

<p>Status: 401 Unauthorized</p> <pre>{ "code": "no_auth_header", "description": "Authorization header is missing" }</pre>
<p>Status: 401 Unauthorized</p> <pre>{ "code": "invalid_header", "description": "Invalid header. Use an RS256 signed JWT Access Token" }</pre>
<p>Status: 401 Unauthorized</p> <pre>{ "code": "token_expired", "description": "token is expired" }</pre>
<p>Status: 401 Unauthorized</p> <pre>{ "code": "invalid_claims", "description": "incorrect claims, please check the audience and issuer" }</pre>
<p>Status: 401 Unauthorized</p> <pre>{ "code": "invalid_header", "description": "Unable to parse authentication token" }</pre>
<p>Status: 401 Unauthorized</p> <pre>{ "code": "no_rsa_key", "description": "No RSA key in JWKS" }</pre>

Error 405 Response Example

<p>Status: 405 Method Not Allowed</p> <pre>{ "Error": "Method Not Allowed. Check the Allowed header to see which methods are allowed for this endpoint." }</pre>
--