TO-DO LIST API DOCUMENT

Amelia Walsh (for Cloud Application Development CS493)

URL: https://cs493-final-walshame.wl.r.appspot.com/
LOGIN: https://cs493-final-walshame.wl.r.appspot.com/

***__Signup works__ users for contingency only **

email: kate@bush.com
Password: *Runn1ngUpThatH1ll*

email: wednesday@adams.com
Password: *Thing1sCool*

Relationship between non-user entities: In this simple to do list API the two non-user data types are lists and items. An item can be added and removed from a list. Item entities have a 'list' property where a list's identification number can be stored. Items can only be on one list at a time. List entities have an 'items' property that contains an array of items assigned to that list.

User-entity modeling: This api uses auth0 and open-id-connect as modeled in the assignment 7 example. Auth0 handles the storage of users and the creation of user_ids. Lists and items are both owned by users. The auth0 user_id's are stored in the 'owner' property of both lists and items.

***** PUT/items, PUT/lists support 405 HTTP response***

DATA MODEL

List

Property	***Data Type***	***Notes***
id	integer	datastore -generated integer that represents boat entity
name (required)	string	name of to-do list
label (required)	string	type of list (personal, work, etc)
date_created	string	timestamp of list creation
owner	string	user-owner of the list sub property
items	array	array of item ids and urls on this list
self	string	url to get this item

Item

Property	***Data Type***	***Notes***
id	integer	datastore -generated integer that represents boat entity
name (required)	string	name of to-do item
done (required)	boolean	true completed, false incomplete
due_date (required)	string	date string formatted to ISO 8601
priority (required)	Integer	Range 0-5, 0 being lowest priority and 5 being highest
list	integer	id of list this item belongs to
owner	string	user-owner of the list sub property
self	string	url to get this item

REQUESTS

Create Item

POST/items

Request:

Path Parameters: none

Request Header: "Authorization" Bearer [JSON Web Token]

Request Body (MIME type application/json): Attributes -- (name, done, due_date, priority)

Request Body Example:

```
{
  "name": "clean bathroom",
  "done": false,
  "due_date": "2022-11-29T23:59:59-0800",
  "priority": 3
}
```

Response:

Status Codes

Outcome	Status Code	Notes
success	201 Created	
failure	400 Bad Request	missing/incorrect attributes, incorrect date format, priority outside range (0-5)
failure	401 Unauthorized	JSON Web Token invalid
failure	406 Not Acceptable	unsupported return type in accepts header

Request Response Examples:

Success (201 Created):

```
{
    "id": "5073695114526720",
    "self": "http://localhost:8080/items/5073695114526720"
}
```

Failure (400 Bad Request) – attribute error:

```
{
    "Error": "400 Bad Request: Request body contains invalid
     attributes."
}
```

Failure (400 Bad Request) – invalid date:

```
{
    "Error": "400 Bad Request: Request body has invalid
     'due_date' value."
}
```

Failure (400 Bad Request) – invalid priority:

```
{
    "Error": "400 Bad Request: Request body has invalid
     'priority' value."
}
```

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (406 Not Acceptable):

```
{
    "Error": "406 Not Acceptable: Return type in Accept header
     unsupported."
}
```

Get Items

GET/items

Request:

Path Parameters: none

Request Header: “Authorization” Bearer [JSON Web Token]

Request Body: none

Response:

Status Codes

Outcome	Status Code	Notes
success	200 OK	pagination items > 5
failure	401 Unauthorized	JSON Web Token invalid
failure	406 Not Acceptable	unsupported return type in accepts header

Request Response Examples:

Success (200 OK) – items < 5:

```
{
    "total_items": 2,
    "items": [
        {
            "priority": 5,
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "due_date": "2022-12-29T23:59:59-0800",
            "list": null,
            "name": "Wash dishes",
            "done": false,
            "id": "5277197375897600"
        },
        {
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "name": "Vaccuum floors",
            "priority": 0,
            "done": false,
            "due_date": "2022-11-29T23:59:59-0800",
            "list": null,
            "id": "5873666730491904"
        }
    ]
}
```

Success (200 OK) – items > 5:

```
{
    "total_items": 6,
    "items": [
        {
            "done": false,
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "priority": 3,
            "due_date": "2022-12-13T23:59:59-0800",
            "name": "Clean mirrors",
            "list": null,
            "id": "4524433356095488"
        },
        {
            "done": false,
            "name": "Wash dishes",
            "list": null,
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "due_date": "2022-12-29T23:59:59-0800",
            "priority": 5,
            "id": "5277197375897600"
        },
        {
            "due_date": "2022-12-10T23:59:59-0800",
            "list": null,
            "done": false,
            "name": "Clean toilet",
            "priority": 2,
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "id": "5290624651624448"
        },
        {
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "done": false,
            "priority": 2,
            "list": null,
            "name": "Clean Shower",
            "due_date": "2022-12-01T23:59:59-0800",
            "id": "5509714557272064"
        },
        {
            "priority": 0,
            "name": "Vaccuum floors",
            "list": null,
            "due_date": "2022-11-29T23:59:59-0800",
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "done": false,
            "id": "5873666730491904"
        }
    ],
    "next": "http://localhost:8080/items?
cursor=CjESK2oWbX5jczQ5My1maW5hbC13YWxzaGFtZXIRCxIEaXRlbRiAgIDYmcK3CgwYACAA"
}
```

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (406 Not Acceptable):

```
{
    "Error": "406 Not Acceptable: Return type in Accept header
     unsupported."
}
```

Get an Item

GET/items/item_id

Request:

Path Parameters: item_id (integer assigned by datastore)

Request Header: “Authorization” Bearer [JSON Web Token]

Request Body: none

Response:

Status Codes

Outcome	Status Code	Notes
success	200 OK	
failure	401 Unauthorized	JSON Web Token invalid
failure	403 Forbidden	item with this id isn’t owned by user
failure	404 Not Found	invalid item_id
failure	406 Not Acceptable	unsupported return type in accepts header

Request Response Examples:

Success (200 OK):

```
{
    "name": "Vaccuum floors",
    "priority": 0,
    "owner": "auth0|63824f2bc0f8aabff64a8728",
    "list": null,
    "done": false,
    "due_date": "2022-11-29T23:59:59-0800",
    "id": "5873666730491904",
    "self": "http://localhost:8080/items/5873666730491904"
}
```

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (403 Forbidden):

```
{
    "Error": "403 Forbidden: User is not authorized to access
     this item."
}
```

Failure (404 Not Found):

```
{
    "Error": "404 Not Found: Request id parameter invalid."
}
```

Failure (406 Not Acceptable):

```
{
    "Error": "406 Not Acceptable: Return type in Accept header
     unsupported."
}
```

Update Item (all non-list attributes)

PUT/items/item_id

Request:

Path Parameters: item_id (integer assigned by datastore)

Request Header: “Authorization” Bearer [JSON Web Token]

*Request Body (MIME type application/json): Attributes -- (name, done, due_date, priority) ** all required*

Request Body Example:

```
{
  "name": "Mop floors",
  "done": false,
  "due_date": "2022-11-29T23:59:59-0800",
  "priority": 3
}
```

Response:

Status Codes

Outcome	Status Code	Notes
success	204 No Content	
failure	400 Bad Request	missing/incorrect attributes, incorrect date format, priority outside range (0-5)
failure	401 Unauthorized	JSON Web Token invalid
failure	403 Forbidden	item with this id isn’t owned by user
failure	404 Not Found	invalid item_id
failure	406 Not Acceptable	unsupported return type in accepts header

Request Response Examples:

Success (204 No Content):

Failure (400 Bad Request) – attribute error:

```
{
    "Error": "400 Bad Request: Request body contains invalid
     attributes."
}
```

Failure (400 Bad Request) – invalid date:

```
{
    "Error": "400 Bad Request: Request body has invalid
     'due_date' value."
}
```

Failure (400 Bad Request) – invalid priority:

```
{
    "Error": "400 Bad Request: Request body has invalid
     'priority' value."
}
```

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (403 Forbidden):

```
{
    "Error": "403 Forbidden: User is not authorized to access
     this item."
}
```

Failure (404 Not Found):

```
{
    "Error": "404 Not Found: Request id parameter invalid."
}
```

Failure (406 Not Acceptable):

```
{
    "Error": "406 Not Acceptable: Return type in Accept header
     unsupported."
}
```

Update Item (some non-list attributes)

PATCH/items/item_id

Request:

Path Parameters: item_id (integer assigned by datastore)

Request Header: "Authorization" Bearer [JSON Web Token]

Request Body (MIME type application/json): Attributes -- (name, done, due_date, priority)

Request Body Example:

```
{
  "done": true,
  "priority": 5
}
```

Response:

Status Codes

Outcome	Status Code	Notes
success	204 No Content	
failure	400 Bad Request	incorrect attributes, incorrect date format, priority outside range (0-5)
failure	401 Unauthorized	JSON Web Token invalid
failure	403 Forbidden	item with this id isn't owned by user
failure	404 Not Found	invalid item_id
failure	406 Not Acceptable	unsupported return type in accepts header

Request Response Examples:

Success (204 No Content):

Failure (400 Bad Request) – attribute error:

```
{
    "Error": "400 Bad Request: Request body contains invalid
     attributes."
}
```

Failure (400 Bad Request) – invalid date:

```
{
    "Error": "400 Bad Request: Request body has invalid
     'due_date' value."
}
```

Failure (400 Bad Request) – invalid priority:

```
{
    "Error": "400 Bad Request: Request body has invalid
     'priority' value."
}
```

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (403 Forbidden):

```
{
    "Error": "403 Forbidden: User is not authorized to access
     this item."
}
```

Failure (404 Not Found):

```
{
    "Error": "404 Not Found: Request id parameter invalid."
}
```

Failure (406 Not Acceptable):

```
{
    "Error": "406 Not Acceptable: Return type in Accept header
     unsupported."
}
```

Delete an Item

DELETE/items/item_id

Request:

Path Parameters: item_id (assigned by datastore)

Request Header: "Authorization" Bearer [JSON Web Token]

Request Body: none

Response:

Status Codes

Outcome	Status Code	Notes
success	204 No Content	
failure	401 Unauthorized	JSON Web Token invalid
failure	403 Forbidden	item with this id isn't owned by user
failure	404 Not Found	invalid item_id

Request Response Examples:

Success (204 No Content):

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (403 Forbidden):

```
{
    "Error": "403 Forbidden: User is not authorized to access this item."
}
```

Failure (404 Not Found):

```
{
    "Error": "404 Not Found: Request id parameter invalid."
}
```

Assign Item to List

PUT/lists/list_id/item/item_id

Request:

Path Parameters: item_id, list_id (integers assigned by datastore)

Request Header: "Authorization" Bearer [JSON Web Token]

Request Body: none

Response:

Status Codes

Outcome	Status Code	Notes
success	204 No Content	
failure	401 Unauthorized	JSON Web Token invalid
failure	403 Forbidden	item/list with this id isn't owned by user, item already assigned to list
failure	404 Not Found	invalid item_id/list_id

Request Response Examples:

Success (204 No Content):

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (403 Forbidden) – incorrect owner:

```
{
    "Error": "403 Forbidden: User is not authorized to access
     this item."
}
```

Failure (403 Forbidden) – item already assigned:

```
{
    "Error": "403 Forbidden: Item is already assigned to list"
}
```

Failure (404 Not Found) – list_id or item_id invalid:

```
{
    "Error": "404 Not Found: Request id parameter invalid."
}
```

Remove Item from List

DELETE/lists/list_id/item/item_id

Request:

Path Parameters: item_id, list_id (integers assigned by datastore)
Request Header: “Authorization” Bearer [JSON Web Token]
Request Body: none

Response:

Status Codes

Outcome	Status Code	Notes
success	204 No Content	
failure	401 Unauthorized	JSON Web Token invalid
failure	403 Forbidden	item/list with this id isn’t owned by user
failure	404 Not Found	invalid item_id/list_id

Request Response Examples:

Success (204 No Content):

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (403 Forbidden):

```
{
    "Error": "403 Forbidden: User is not authorized to access
     this item."
}
```

Failure (404 Not Found) – list_id or item_id invalid:

```
{
    "Error": "404 Not Found: Request id parameter invalid."
}
```

Create List

POST/lists

Request:

Path Parameters: none

Request Header: "Authorization" Bearer [JSON Web Token]

Request Body (MIME type application/json): Attributes -- (name, label)

Request Body Example:

```
{
  "name": "Weekly Cleaning To-Do List",
  "label": "personal"
}
```

Response:

Status Codes

Outcome	Status Code	Notes
success	201 Created	
failure	400 Bad Request	missing/incorrect attributes
failure	401 Unauthorized	JSON Web Token invalid
failure	406 Not Acceptable	unsupported return type in accepts header

Request Response Examples:

Success (201 Created):

```
{
    "id": "5756247156457472",
    "self": "http://localhost:8080/lists/5756247156457472"
}
```

Failure (400 Bad Request):

```
{
    "Error": "400 Bad Request: Request body contains invalid
     attributes."
}
```

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (406 Not Acceptable):

```
{
    "Error": "406 Not Acceptable: Return type in Accept header
     unsupported."
}
```

Get Lists

GET/lists

Request:

Path Parameters: none

Request Header: “Authorization” Bearer [JSON Web Token]

Request Body: none

Response:

Status Codes

Outcome	Status Code	Notes
success	200 OK	pagination lists > 5
failure	401 Unauthorized	JSON Web Token invalid
failure	406 Not Acceptable	unsupported return type in accepts header

Request Response Examples:

Success (200 OK) – lists < 5:

```
{
    "total_items": 1,
    "lists": [
        {
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "label": "personal",
            "name": "Healthcare To-Do List",
            "items": [],
            "date_created": "2022-12-04T03:33:55.832Z",
            "id": "4995722399186944"
        }
    ]
}
```

Success (200 OK) – lists > 5:

```
{
    "total_items": 6,
    "lists": [
        {
            "label": "personal",
            "name": "Healthcare To-Do List",
            "date_created": "2022-12-04T03:33:55.832Z",
            "items": [],
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "id": "4995722399186944"
        },
        {
            "label": "personal",
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "items": [],
            "date_created": "2022-12-04T03:33:49.642Z",
            "name": "Weekly Cleaning To-Do List",
            "id": "5640638045356032"
        },
        {
            "items": [],
            "date_created": "2022-12-04T03:34:09.097Z",
            "name": "Movies to Watch",
            "label": "personal",
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "id": "5650333262938112"
        },
        {
            "label": "personal",
            "date_created": "2022-12-04T03:34:05.896Z",
            "items": [],
            "name": "Queer Camp Planning To-Do List",
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "id": "5896984644812800"
        },
        {
            "label": "work",
            "date_created": "2022-12-04T03:34:02.499Z",
            "owner": "auth0|63824f2bc0f8aabff64a8728",
            "name": "Classroom Documentation To-Do List",
            "items": [],
            "id": "6121622306029568"
        }
    ],
    "next": "http://localhost:8080/lists?
cursor=CjESK2oWbX5jczQ5My1maW5hbC13YWxzaGFtZXIRCxIEbGlzdBiAgICY1PLvCgw
YACAA"
}
```

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (406 Not Acceptable):

```
{
    "Error": "406 Not Acceptable: Return type in Accept header
     unsupported."
}
```

Get a List

GET/list/list_id

Request:

Path Parameters: list_id (integer assigned by datastore)

Request Header: “Authorization” Bearer [JSON Web Token]

Request Body: none

Response:

Status Codes

Outcome	Status Code	Notes
success	200 OK	
failure	401 Unauthorized	JSON Web Token invalid
failure	403 Forbidden	list with this id isn’t owned by user
failure	404 Not Found	invalid list_id
failure	406 Not Acceptable	unsupported return type in accepts header

Request Response Examples:

Success (200 OK):

```
{
    "label": "personal",
    "owner": "auth0|63824f2bc0f8aabff64a8728",
    "name": "Weekly Cleaning To-Do List",
    "items": [],
    "date_created": "2022-12-04T03:45:43.352Z",
    "id": "5558672352608256",
    "self": "http://localhost:8080/lists/5558672352608256"
}
```

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (403 Forbidden):

```
{
    "Error": "403 Forbidden: User is not authorized to access
     this item."
}
```

Failure (404 Not Found):

```
{
    "Error": "404 Not Found: Request id parameter invalid."
}
```

Failure (406 Not Acceptable):

```
{
    "Error": "406 Not Acceptable: Return type in Accept header
     unsupported."
}
```

Update List (all non-item attributes)

PUT/lists/list_id

Request:

Path Parameters: list_id (integer assigned by datastore)

Request Header: “Authorization” Bearer [JSON Web Token]

*Request Body (MIME type application/json): Attributes -- (name, label) ** all required (date_created cannot be changed)*

Request Body Example:

```
{
  "name": "Weekly Tidying To-Do List",
  "label": "personal"
}
```

Response:

Status Codes

Outcome	Status Code	Notes
success	204 No Content	
failure	400 Bad Request	missing/incorrect attributes
failure	401 Unauthorized	JSON Web Token invalid
failure	403 Forbidden	list with this id isn’t owned by user
failure	404 Not Found	invalid list_id
failure	406 Not Acceptable	unsupported return type in accepts header

Request Response Examples:

Success (204 No Content):

Failure (400 Bad Request):

```
{
    "Error": "400 Bad Request: Request body contains invalid
     attributes."
}
```

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (403 Forbidden):

```
{
    "Error": "403 Forbidden: User is not authorized to access
     this item."
}
```

Failure (404 Not Found):

```
{
    "Error": "404 Not Found: Request id parameter invalid."
}
```

Failure (406 Not Acceptable):

```
{
    "Error": "406 Not Acceptable: Return type in Accept header
     unsupported."
}
```

Update List (some non-item attributes)

PATCH/lists/list_id

Request:

Path Parameters: list_id (integer assigned by datastore)

Request Header: "Authorization" Bearer [JSON Web Token]

Request Body (MIME type application/json): Attributes -- (name, label)

Request Body Example:

```
{
  "label": "work"
}
```

Response:

Status Codes

Outcome	Status Code	Notes
success	204 No Content	
failure	400 Bad Request	incorrect attributes
failure	401 Unauthorized	JSON Web Token invalid
failure	403 Forbidden	list with this id isn't owned by user
failure	404 Not Found	invalid list_id
failure	406 Not Acceptable	unsupported return type in accepts header

Request Response Examples:

Success (204 No Content):

Failure (400 Bad Request):

```
{
    "Error": "400 Bad Request: Request body contains invalid
     attributes."
}
```

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (403 Forbidden):

```
{
    "Error": "403 Forbidden: User is not authorized to access
     this item."
}
```

Failure (404 Not Found):

```
{
    "Error": "404 Not Found: Request id parameter invalid."
}
```

Failure (406 Not Acceptable):

```
{
    "Error": "406 Not Acceptable: Return type in Accept header
     unsupported."
}
```

Delete a List

DELETE/lists/list_id

Request:

Path Parameters: list_id (assigned by datastore)

Request Header: “Authorization” Bearer [JSON Web Token]

Request Body: none

Response:

Status Codes

Outcome	Status Code	Notes
success	204 No Content	
failure	401 Unauthorized	JSON Web Token invalid
failure	403 Forbidden	list with this id isn’t owned by user
failure	404 Not Found	invalid list_id

Request Response Examples:

Success (204 No Content):

Failure (401 Unauthorized):

```
{
    "Error": "401 Unauthorized : Invalid JSON web token."
}
```

Failure (403 Forbidden):

```
{
    "Error": "403 Forbidden: User is not authorized to access
     this item."
}
```

Failure (404 Not Found):

```
{
    "Error": "404 Not Found: Request id parameter invalid."
}
```

Get Users

GET/users

Request:

Path Parameters: none

Request Body: none

Response:

Status Codes

Outcome	Status Code	Notes
success	200 OK	‘html’

Request Response Examples:

Success (200 OK):

```
<html>

<head>
    <link rel="stylesheet" type="text/css" href="/css/style.css" />
</head>

<body>
    <h1>Users</h1>
    <h3>CS493 Homework 7</h3>
    <ul>
        <li>wednesday@adams.com, auth0|6384217fa8b2c2ec60b33387</li>
        <li>kate@bush.com, auth0|63824f2bc0f8aabff64a8728</li>
    </ul>
</body>

</html>
```