

Assignment: Final Project  
Student: Jeongmuk Oh (934314624)  
Date: 11th Dec, 2023

### The URL where my application is deployed on GCP:

<https://cs493-assignment3-403322.uw.r.appspot.com>

### The URL for account creation/login:

Login: <https://cs493-assignment3-403322.uw.r.appspot.com>

Signup: <https://cs493-assignment3-403322.uw.r.appspot.com/signup?>

### Data Model Section

#### 1. Boat

- Properties for None User

```
{  
  "id": 5178988653707264,  
  "length": 52,  
  "name": "Jeongmuk Oh",  
  "protected": false,  
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5178988653707264",  
  "type": "Computer Science"  
}
```

#### Required Attributes:

- 1) length: (int), number greater than 0
- 2) name: (string), Only uppercase and lowercase letters and spaces are allowed
- 3) type: (string), Only uppercase and lowercase letters and spaces are allowed

#### Automatically created Attributes:

- 1) 'protected': (bool), if the data is going to be protected. If not protected, False.
- 2) 'self': (string), self-link where the data is stored at.
- 3) 'loads': (list). loads assigned to this boat

- Properties for User

```
{  
  "id": 5699103757434880,  
  "length": 52,  
  "name": "Jeongmuk Oh",  
  "protected": true,  
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5699103757434880",  
  "sub": "auth0|6575140c1b69345b9278926c",  
  "type": "cs"  
}
```

```
}
```

Required Attributes:

- 1) length: (int), number greater than 0
- 2) name: (string), Only uppercase and lowercase letters and spaces are allowed
- 3) type: (string), Only uppercase and lowercase letters and spaces are allowed

Automatically created Attributes:

- 4) 'protected': (bool), if the data is going to be protected. If protected, True.
- 5) 'self': (str), self-link where the data is stored at.
- 6) 'sub': (str), ID that identifies the users
- 7) 'loads': (list). loads assigned to this boat

## 2. Load

- Properties for None User

```
{  
  "carrier": null,  
  "creation_date": "10/18/2021",  
  "id": 6493235895599104,  
  "item": "LEGO Blocks",  
  "protected": false,  
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/6493235895599104",  
  "volume": 5  
}
```

Required Attributes:

- 1) volume: (int), number greater than 0
- 2) item: (string), Only uppercase and lowercase letters and spaces are allowed
- 3) creation\_date: (string), the only type above like "MM/DD/YYYY"

Automatically created Attributes:

- 4) 'protected': (bool), if the data is going to be protected. If not protected, False.
- 5) 'self': (str), self-link where the data is stored at.
- 8) 'carrier': (list), a container being able to make a list for loads

- Properties for User

```
{  
  "carrier": null,  
  "creation_date": "10/18/2021",  
  "id": 6329990094258176,  
  "item": "LEGO Blocks",  
  "protected": true,  
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/6329990094258176",  
  "sub": "auth0|6575140c1b69345b9278926c",  
}
```

```
"volume": 5
}
```

Required Attributes:

- 1) volume: (int), number greater than 0
- 2) item: (string), Only uppercase and lowercase letters and spaces are allowed
- 3) creation\_date: (string), the only type above like "MM/DD/YYYY"

Automatically created Attributes:

- 4) 'protected': (bool), if the data is going to be protected. If not protected, False.
- 5) 'self': (str), self-link where the data is stored at.
- 6) 'carrier': (list), a container being able to make a list for loads
- 7) 'sub': (str), ID that identifies the users

### 3. User

: When a user creates an account, it is automatically registered in the User entity database.

- Properties

```
{
  "GCP_id": 5672404563001344,
  "email": "ojm1818@gmail.com",
  "sub": "auth0|6575140c1b69345b9278926c"
}
```

Automatically created Attributes:

- 1) 'email': (str), user email
- 2) 'sub': (str), ID that identifies the users
- 3) GCP\_id: (int) data id

## The relationship between the non-user entities

1. created 'Boat' and 'Load' as non-user entities. 1. 'Load' can only be assigned to one 'Boat', and 'Boat' can contain multiple 'Load's.
2. If a specific Load has already been assigned to a Boat and you try to delete that Load, the Load will be deleted from the 'loads' attribute list of the Boat containing the Load.
3. If the 'carrier' property of a specific Boat contains Loads and you attempt to delete that Boat, the 'carrier' property of the Load is cleared and the Boat is deleted.

## How I am modeling the user entity in the application

- What is the unique identifier for a user?

User requests are sent with a JWT valid token included in the Authorization. All possible endpoints inspect the user request to see if the header contains a JWT. After completing token validation, the extracted sub is used as a unique ID.

- What needs to be specified by the person making the request?

Users must send their valid token with every request from any endpoint.

- How your application maps a supplied JWT to the identifier of a user.

1. If it is confirmed that a JWT is included, check whether the JWT is valid.
2. Checking is done in the path '/decode'.
3. If the JWT test is successfully completed, sub information is extracted from the token.
4. If the operation the user is trying to perform is a CRUD operation related to 'Boat' and 'Load', check whether it matches the non-user entity 'Boat' and 'Load' property 'sub' based on the extracted sub information.
5. If there is a match, the user can continue performing the task.

- What is the relationship between the user entity and non-user entity?

The user entity contains a unique attribute 'sub'. When a specific user performs a CRUD operation on a non-user entity, 'sub' is added as an attribute of the non-user entity to leave traces of the specific user.

If a user wants to continue working by accessing previously created data, the 'sub' data left in non-user properties is used.

## All EndPoints & Methods & Protected & Status Code & Sample req and res.

- Endpoint: '/boats'  
Supported Methods: ['POST', 'GET']

### 1) 'POST':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(201), Accept Error(406), Missed Attributes(400),  
invalid/missing JWT(401)
- Sample req & res:

### [req for 201]

```
{
  "name": "Jeongmuk Oh",
  "type": "cs",
  "length": 52
}
```

### [res for 201]

#### Protected: 302

```
{
  "id": 5951867817295872,
  "length": 52,
  "loads": [],
  "name": "Jeongmuk Oh",
  "protected": true,
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5951867817295872",
  "sub": "auth0|6575140c1b69345b9278926c",
  "type": "cs"
}
```

#### Unprotected 302

```
{
  "id": 5951867817295872,
```

```
"length": 52,  
"loads": [],  
"name": "Jeongmuk Oh",  
"protected": false,  
"self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5951867817295872",  
"type": "cs"  
}
```

### [req for 400] Missing Attributes

```
{  
  "name": "Jeongmuk Oh",  
  "length" : 52  
}
```

### [res for 400]

#### Protected & Unprotected

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

### [req for 401] Missing JWT / Invalid JWT

```
{  
  "name": "Jeongmuk Oh",  
  "type": "cs",  
  "length" : 52  
}
```

### [res for 401]

#### Protected

```
{  
  "Error": "Invalid JWT."  
}
```

### [req for 406] Accept Error:

```
{
  "name": "Jeongmuk Oh",
  "type": "cs",
  "length": 52
}
```

### [res for 406]

#### Protected & Unprotected:

```
{
  "Error": "Accept Content Type user request can't be supported by server"
}
```

#### 1) 'GET':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(200), Accept Error(406), invalid/missing JWT(401)
- Sample req & res:

### [res for 200] pages:

#### Protected:

```
{
  "boats": [
    {
      "id": 4825967910453248,
      "length": 52,
      "loads": [
        {
          "id": 6252131799531520,
          "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/6252131799531520"
        }
      ],
      "name": "Jeongmuk Oh",
      "protected": true,
      "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/4825967910453248",
      "sub": "auth0|6575140c1b69345b9278926c",
      "type": "cs"
    },
  ],
}
```

```
{
  "id": 5110792424783872,
  "length": 52,
  "loads": [],
  "name": "Odliis",
  "protected": true,
  "self": "http://localhost:8080/boats/5110792424783872",
  "sub": "auth0|6575140c1b69345b9278926c",
  "type": "kk-562"
},
{
  "id": 5121158059917312,
  "length": 52,
  "loads": [
    {
      "id": 6233342794006528,
      "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/6233342794006528"
    },
    {
      "id": 6239544055693312,
      "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/6239544055693312"
    }
  ],
  "name": "Jeongmuk Oh",
  "protected": true,
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5121158059917312",
  "sub": "auth0|6575140c1b69345b9278926c",
  "type": "cs"
},
{
  "id": 5132278334226432,
  "length": 52,
  "loads": [],
  "name": "Jeongmuk Oh",
  "protected": true,
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5132278334226432",
  "sub": "auth0|6575140c1b69345b9278926c",
  "type": "cs"
},
{
  "id": 5162520876679168,
  "length": 52,
```



```

    "loads": [],
    "name": "Odlis",
    "protected": true,
    "self": "http://localhost:8080/boats/5162520876679168",
    "sub": "auth0|6575140c1b69345b9278926c",
    "type": "kk-562"
  }
],
"next": "https://cs493-assignment3-403322.uw.r.appspot.com/boats?limit=5&offset=5"
}

```

## Unprotected:

```

{
  "boats": [
    {
      "id": 4814221644660736,
      "length": 52,
      "loads": [],
      "name": "Jeongmuk Oh",
      "protected": false,
      "self": "http://localhost:8080/boats/4814221644660736",
      "type": "Computer Science"
    },
    {
      "id": 5071950955151360,
      "length": 52,
      "loads": [],
      "name": "Jeongmuk Oh",
      "protected": false,
      "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5071950955151360",
      "type": "Computer Science"
    },
    {
      "id": 5095696621371392,
      "length": 52,
      "loads": [],
      "name": "Odlis",
      "protected": false,
      "self": "http://localhost:8080/boats/5095696621371392",
      "type": "kk-562"
    },
  ],
}

```

```

{
  "id": 5107442887163904,
  "length": 52,
  "loads": [],
  "name": "Jeongmuk Oh",
  "protected": false,
  "self": "http://localhost:8080/boats/5107442887163904",
  "type": "Computer Science"
},
{
  "id": 5151970692169728,
  "length": 52,
  "loads": [],
  "name": "Odlis",
  "protected": false,
  "self": "http://localhost:8080/boats/5151970692169728",
  "type": "kk-562"
}
],
"next": "https://cs493-assignment3-403322.uw.r.appspot.com/boats?limit=5&offset=5"
}

```

## [res for 406] Accept Error Protected & Unprotected

```

{
  "Error": "Accept Content Type user request can't be supported by server"
}

```

## [res for 401] Invalid/Missing JWT Protected

```

{
  "Error": "Invalid JWT."
}

```

All Accept errors and Invalid Missing JWT errors are the same. so I will omit them.

- Endpoint: '/boats/<b\_id>'  
Supported Methods: ['GET', 'PUT', 'PATCH', 'DELETE']

1) 'GET':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(200), Accept Error(406), Invalid Boat ID(404),  
invalid/missing JWT(401)
- Sample req & res:

**[res for 200] Success**

**Protected**

```
{
  "id": 5951867817295872,
  "length": 52,
  "name": "Jeongmuk Oh",
  "protected": true,
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5951867817295872",
  "sub": "auth0|6575140c1b69345b9278926c",
  "type": "cs"
}
```

**Unprotected**

```
{
  "id": 5178988653707264,
  "length": 52,
  "name": "Jeongmuk Oh",
  "protected": false,
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5178988653707264",
  "type": "Computer Science"
}
```

### [res for 404] Invalid Boat ID

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

#### 1) 'PUT':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(303), Accept Error(406), Invalid Boat ID(404),  
invalid/missing JWT(401)
- Sample req & res:

### [req for 303] Success

```
{  
  "name": "Jeongmuk",  
  "type": "Computer",  
  "length": 15  
}
```

### [res for 303]

#### Protected & UnProtected

```
{  
  "id": 5178988653707264,  
  "length": 15,  
  "name": "Jeongmuk",  
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5178988653707264",  
  "type": "Computer"  
}
```

#### 1) 'PUT':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(303), Accept Error(406), Invalid Boat ID(404),  
invalid/missing JWT(401)
- Sample req & res:

#### [req for 303] Success

```
{  
  "name": "Jeongmuk",  
  "length": 15  
}
```

#### [res for 303]

##### Protected & UnProtected

```
{  
  "length": 15,  
  "name": "Jeongmuk"  
}
```

#### 1) 'DELETE':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(204), Accept Error(406), Invalid Boat ID(404),  
invalid/missing JWT(401)
- Sample req & res:

#### [res for 204] Success

##### Protected & Unprotected

```
return(' ', 204)
```

- Endpoint: '/loads'  
Supported Methods: ['POST', 'GET']

1) 'POST':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(201), Accept Error(406), invalid/missing JWT(401),  
Missing Attributes(400)
- Sample req & res:

**[req for 201] Success**

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

**[res for 201]**

**Protected**

```
{  
  "carrier": null,  
  "creation_date": "10/18/2021",  
  "id": 5388917863874560,  
  "item": "LEGO Blocks",  
  "protected": false,  
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/5388917863874560",  
  "volume": 5  
}
```

**Unprotected**

```
{  
  "carrier": null,  
  "creation_date": "10/18/2021",  
  "id": 6329990094258176,  
  "item": "LEGO Blocks",  
  "protected": true,  
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/6329990094258176",  
}
```

```
"sub": "auth0|6575140c1b69345b9278926c",  
"volume": 5  
}
```

1) 'GET':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(200), Accept Error(406), invalid/missing JWT(401)
- Sample req & res:

**[res for 200] Success  
Protected**

```
{  
  "boats": [  
    {  
      "carrier": null,  
      "creation_date": "10/18/2021",  
      "id": 4832169172140032,  
      "item": "LEGO Blocks",  
      "protected": true,  
      "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/4832169172140032",  
      "sub": "auth0|6575140c1b69345b9278926c",  
      "volume": 5  
    },  
    {  
      "carrier": null,  
      "creation_date": "10/18/2021",  
      "id": 4839683083206656,  
      "item": "LEGO Blocks",  
      "protected": true,  
      "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/4839683083206656",  
      "sub": "auth0|6575140c1b69345b9278926c",  
      "volume": 5  
    },  
    {  
      "carrier": null,  
      "creation_date": "10/18/2021",
```

```

    "id": 4899245085687808,
    "item": "LEGO Blocks",
    "protected": true,
    "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/4899245085687808",
    "sub": "auth0|6575140c1b69345b9278926c",
    "volume": 5
  },
  {
    "carrier": null,
    "creation_date": "10/18/2021",
    "id": 5077016667750400,
    "item": "LEGO Blocks",
    "protected": true,
    "self": "http://localhost:8080/loads/5077016667750400",
    "sub": "auth0|6575140c1b69345b9278926c",
    "volume": 5
  },
  {
    "carrier": {
      "id": 5465048373788672,
      "name": "Jeongmuk Oh",
      "self": "https://cs493-assignment3-403322.uw.r.appspot.com/boats/5465048373788672"
    },
    "creation_date": "10/18/2021",
    "id": 5093331570786304,
    "item": "LEGO Blocks",
    "protected": true,
    "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/5093331570786304",
    "sub": "auth0|6575140c1b69345b9278926c",
    "volume": 5
  }
],
"next": "https://cs493-assignment3-403322.uw.r.appspot.com/loads?limit=5&offset=5"
}

```

## Unprotected

```

{
  "loads": [
    {
      "carrier": null,
      "creation_date": "10/10/2018",

```



```
    "id": 4902098420367360,
    "item": "LEGO",
    "protected": false,
    "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/4902098420367360",
    "volume": 120
  },
  {
    "carrier": null,
    "creation_date": "10/18/2021",
    "id": 5070729708044288,
    "item": "LEGO Blocks",
    "protected": false,
    "self": "http://localhost:8080/loads/5070729708044288",
    "volume": 5
  },
  {
    "carrier": null,
    "creation_date": "10/18/2021",
    "id": 5111941596971008,
    "item": "LEGO Blocks",
    "protected": false,
    "self": "http://localhost:8080/loads/5111941596971008",
    "volume": 5
  },
  {
    "carrier": null,
    "creation_date": "10/18/2021",
    "id": 5123625652846592,
    "item": "LEGO Blocks",
    "protected": false,
    "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/5123625652846592",
    "volume": 5
  },
  {
    "carrier": null,
    "creation_date": "10/10/2019",
    "id": 5163314036342784,
    "item": "LEGO",
    "protected": false,
    "self": "http://localhost:8080/loads/5163314036342784",
    "volume": 20
  }
}
```

```

    ],
    "next": "https://cs493-assignment3-403322.uw.r.appspot.com/loads?limit=5&offset=5"
  }
}

```

- Endpoint: '/loads/<id>'  
Supported Methods: ['GET', 'PUT', 'PATCH', 'DELETE']

#### 1) 'GET':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(200), Accept Error(406), Invalid Load ID(404),  
invalid/missing JWT(401)
- Sample req & res:

### [res for 200] Success Protected

```

{
  "carrier": null,
  "creation_date": "10/18/2021",
  "id": 6278645437956096,
  "item": "LEGO Blocks",
  "protected": true,
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/6278645437956096",
  "sub": "auth0|6575140c1b69345b9278926c",
  "volume": 5
}

```

### Unprotected

```

{
  "carrier": null,
  "creation_date": "10/18/2021",
  "id": 5388917863874560,
  "item": "LEGO Blocks",
  "protected": false,
  "self": "https://cs493-assignment3-403322.uw.r.appspot.com/loads/5388917863874560",
  "volume": 5
}

```

```
}
```

#### 1) 'PUT':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(303), Accept Error(406), Invalid Load ID(404),  
invalid/missing JWT(401)
- Sample req & res:

#### [res for 303] Success

```
{  
  "volume": 20,  
  "item": "LEGO",  
  "creation_date": "10/10/2019"  
}
```

#### [res for 303] Protected & Unprotected

```
{  
  "creation_date": "10/10/2019",  
  "item": "LEGO",  
  "volume": 20  
}
```

#### 1) 'PATCH':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(303), Accept Error(406), Invalid Load ID(404),  
invalid/missing JWT(401),
- Sample req & res:

#### [res for 303] Success

```
{  
  "volume": 120,  
  "creation_date": "10/10/2018"  
}
```

**[req for 303]**

**Protected & Unprotected**

```
{  
  "creation_date": "10/10/2018",  
  "volume": 120  
}
```

1) 'DELETE':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(204), Accept Error(406), Invalid Boat ID(404),  
invalid/missing JWT(401)
- Sample req & res:

**[res for 204] Success**

**Protected & Unprotected**

```
return('',204)
```

- Endpoint: '/boats/<b\_id>/loads/<l\_id>'  
Supported Methods: ['PUT', 'DELETE']

1) 'PUT':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(204), Accept Error(406), Invalid Boat & Load ID(404),  
invalid/missing JWT(401), already assigned(403)
- Sample req & res:

**[res for 204] Success**

**Protected & Unprotected**

```
return ('',204)
```

**[res for 403] already assigned**

**Protected & Unprotected**

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

**[res for 404] Invalid Boat & Load ID**

**Protected & Unprotected**

```
{
  "Error": "The specified boat and/or load does not exist or The IDs of either boat or load don't valid"
}
```

1) 'DELETE':

- Protected:  
Protected if the user request includes a token. Otherwise not protected.
- Status Code:  
Success(204), Accept Error(406), Invalid Boat & Load ID(404),  
invalid/missing JWT(401), already unassigned(403)
- Sample req & res:

**[res for 204] Success**  
**Protected & Unprotected**

```
return ( '',204)
```

**[res for 404] already unassigned**  
**Protected & Unprotected**

```
{  
  "Error": "The load is already unassigned on the boat"  
}
```

**[res for 404] Invalid Boat & Load ID**  
**Protected & Unprotected**

```
{  
  "Error": "The specified boat and/or load does not exist or The IDs of either boat or load don't valid"  
}
```

- Endpoint: '/myaccount'  
Supported Methods: ['GET']

1) 'GET':

- Unprotected:  
This is a path where the user email and unique ID can be confirmed through the user ID. Not protected.
- Status Code:  
Success(200), Failed(404)
- Sample req & res:

**[res for 200] Success**

```
{  
  "GCP_id": 5672404563001344,  
  "email": "oim1818@gmail.com",  
  "sub": "auth0|6575140c1b69345b9278926c"  
}
```

- Endpoint: '/users'  
Supported Methods: ['GET']

1) 'GET':

- Unprotected:  
This is a path where the user email and unique ID can be confirmed through the user ID. Not protected.
- Status Code:  
Success(200)
- Sample req & res:

**[res for 200] Success**

```
[  
  {  
    "GCP_id": 5198218900013056,  
    "email": "oim1818@gmail.com",  
    "sub": "auth0|6575140c1b69345b9278926c"  
  },  
  {  
    "GCP_id": 5198218900013056,  
    "email": "oim1818@gmail.com",  
    "sub": "auth0|6575140c1b69345b9278926c"  
  }  
]
```

```
"GCP_id": 5639966621171712,
```

```
"email": "ohjeo@oregonstate.edu",
```

```
"sub": "auth0|65774445ef593280157d73ab"
```

```
}
```

```
[
```