

San Ligtas API's Documentation

<https://github.com/lcySkype/SanLigtas-API>

- [Requirements](#)
- [Installation](#)
- [Configuration](#)
- [Getting Started](#)
- [API Endpoints](#)

Requirements

Postgresql Database
Python 3.x

Installation

UNIX/Linux: Enter on the terminal `make install`

Windows: Run on the command line `pip install -r requirements.txt`

Configuration

Configuration for the local database is located in `postgres_local_base` variable in `app\main\config.py` file. To configure to fit local database, simply change the URI by following this format: `postgresql://USERNAME:PASSWORD@localhost:5432/DATABASE_NAME`

To setup the database, run the command `py manage.py db migrate` followed by `py manage.py db migrate`. Please note to repeat this step for every changes made to the database models.

Getting Started

To run the API, simply call `py manage.py run`. To run tests, call `py manage.py test`.

API Endpoints

- [/user/](#)
- [/auth/](#)

- </distcenter/>

User Related Operations: `/user/`

List all users

GET <http://localhost:5000/user/>

Response:

200 - Success

Note: A list of all registered users will be returned.

Register a user

POST <http://localhost:5000/user/>

Parameters:

```
{
  "email": "string",
  "username": "string",
  "password": "string",
  "public_id": "string"
}
```

Response:

201 - User successfully created.

Note: Upon successful registration, an Authorization token will be generated.

Get a specific user

GET http://localhost:5000/user/{public_id}

Response:

200 - Success

```
{
  "email": "email_string",
  "username": "username_string",
  "password": null,
  "public_id": "string"
}
```

404 - User not found

Note: password will always be null as there is no "password" in the user model, but rather a hashed password under "pass_hash"

Update a specific user

Token Required

PUT http://localhost:5000/user/{public_id}

Parameters:

```
{
  "email": "string",
  "username": "string",

```

```
    "password": "string",  
    "public_id": "string"  
  }
```

Response:

201 - User Successfully Updated.

404 - User not found

Note: Needs a check if current user is admin or if user being updated is current user. If current user is admin, admin can go ahead and update. If current user is not user being updated, then api should return an error. Only a user(and an admin) can update their own data.

Delete a specific user

Token Required

DELETE http://localhost:5000/user/{public_id}

Response:

204 - User Successfully Deleted.

404 - User not found

Note: Needs a check if user being deleted is current user. If true, then user must be logged out before safely deleting.

Authentication Related Operations: **/auth/**

Login

POST <http://localhost:5000/auth/login>

Parameters:

```
{  
  "email": "string",  
  "password": "string"  
}
```

Response:

200 - Success

Note: Authorization token will be returned.

Logout

Token Required

POST <http://localhost:5000/auth/logout>

Response:

200 - Success

Note: Token Required since user needs to login before they logout.

Distribution Center Related Operations: `/distcenter/`

List all centers

Token Required

GET <http://localhost:5000/distcenter/>

Response:

200 - Success

Note: A list of all registered distribution centers will be returned

Register a center

Admin Token Required

POST <http://localhost:5000/distcenter/>

Parameters:

```
{
    "name": "string",
    "address": "string",
    "capacity": "integer"
}
```

Response:

201 - Center successfully created.

Get a specific center

Token Required

GET <http://localhost:5000/distcenter/{id}>

Response:

200 - Success

```
{
    "name": "string",
    "address": "string",
    "capacity": "integer"
}
```

404 - Center not found

Note: GET methods for centers are token required. Other methods are Admin token required

Update a specific center

Admin Token Required

PUT <http://localhost:5000/distcenter/{id}>

Parameters:

```
{
    "name": "string",
    "address": "string",
    "capacity": "integer"
}
```

}

Response:

201 - Center Successfully Updated.

404 - Center not found

Note: Updating a center should cascade. Note that address should be unique, since no 2 buildings are in the exact same address.

Delete a specific center

Admin Token Required

DELETE <http://localhost:5000/distcenter/{id}>

Response:

204 - Center Successfully Deleted.

404 - Center not found