**Database Project APIs Documentation**

**Use the column names in JSON objects as specified in examples.**

1. **GET APIs**
2. localhost:3000/route/APIs/items/found

It will return all the found items in the database.

1. localhost:3000/route/APIs/items/lost

It will return all the lost items in the database.

1. localhost:3000/route/APIs/items/found?id=123

It will return all the found items by a specific user. We must send user\_id in the URL in this format “id=*required\_id*”.

1. localhost:3000/route/APIs/items/lost?id=123

It will return all lost items by a specific user. We must send user\_id in the URL in this format “id=*required\_id*”.

1. localhost:3000/route/APIs/items/found?is\_returned=0

It will return the unreturned found items. We must set the variable here “is\_returned” to zero to retrieve it.

1. localhost:3000/route/APIs/items/found?is\_returned=1

It will return the returned found items because the value of “is\_returned” in URL is 1.

1. localhost:3000/route/APIs/items/lost?is\_returned=0

It will return the unreturned lost items. We must set the variable here “is\_returned” to zero to retrieve it.

1. localhost:3000/route/APIs/items/lost?is\_returned=1

It will return the returned found items because the value of “is\_returned” in URL is 1.

1. localhost:3000/route/APIs/users?id=2

It will return all the info of a specific user in a JSON file. You have to specify the user\_id in URL. In the format “id=*required\_id*”.

1. localhost:3000/route/APIs/users/admin/claimRequests

It will return all the claim requests from the claim\_request Table.

1. localhost:3000/route/APIs/users/admin/claimRequests?id=30

It will return a specific claim request fetched using the request\_id. In the above URL request\_id

#30 is being fetched.

1. **POST APIs**
2. localhost:3000/route/APIs/items/found

It will report a found item. In the input we must send a JSON file with required attributes.

For example: {

  "item\_id": 1,

  "user\_id": "user123",

  "is\_returned": 0,

  "date\_found": "2023-12-18",

  "item\_description": "Found a valuable item",

  "image\_id": "image123",

  "title": "Valuable Item",

  "location": "City Center"

}

Warning:

The user\_id that is being sent here must be in the users table, otherwise this post request won’t work.

1. localhost:3000/route/APIs/items/lost

It will report a lost item. In the input we must send a JSON file with required attributes.

For Example: {

  "item\_id": 1,

  "user\_id": "user123",

  "is\_returned": 0,

  "date\_lost": "2023-12-18",

  "item\_description": "Found a valuable item",

  "image\_id": "image123",

  "title": "Valuable Item",

  "location": "City Center"

}

Warning:

The user\_id that is being sent here must be in the users table, otherwise this post request won’t work.

1. localhost:3000/route/APIs/users

It will register a new user in the database. We must send a JSON input with required attributes.

For Example: {

  "user\_id": "userA",

  "user\_name": "john\_doe\_ric",

  "email": "A@example.com",

  "passwords": "hashedpassword12367",

  "contact\_no": "123-45698-7890",

  "user\_role": "admin",

  "f\_name": "John",

  "l\_name": "Doe",

  "user\_dp\_id": "avatar123"

}

Warning: Consult the schema to check which are unique and not null attrubutes.

1. localhost:3000/route/APIs/users/admin/claimRequests?id=1&is\_resolved=0

It will set the variable is\_resolved value to the input value (0 above) of that specific request\_id.

1. localhost:3000/route/APIs/users/admin/claimRequests?id=1&is\_rejected=0

It will set the variable is\_rejected value to the input value (0 above) of that specific request\_id.

Warning:  
In point 4 and 5, unlike other POST requests, variable data is sent in URL itself.

1. localhost:3000/route/APIs/users/admin/makeRequest

It will make a new claim request, the required attributes must be sent using JSON format.

For Example: {

    "found\_image\_id": "image123",

    "lost\_image\_id": "image123",

    "user\_id": "userA",

    "is\_resolved": 1,

    "is\_rejected": 0

}

Warning: Consult the schema to check which are unique and not null attrubutes.

1. **DELETE APIs**
2. localhost:3000/route/APIs/users?id=user123

It will delete the specific user by its user\_id as given in the URL. It will also delete all its reported lost and found items.