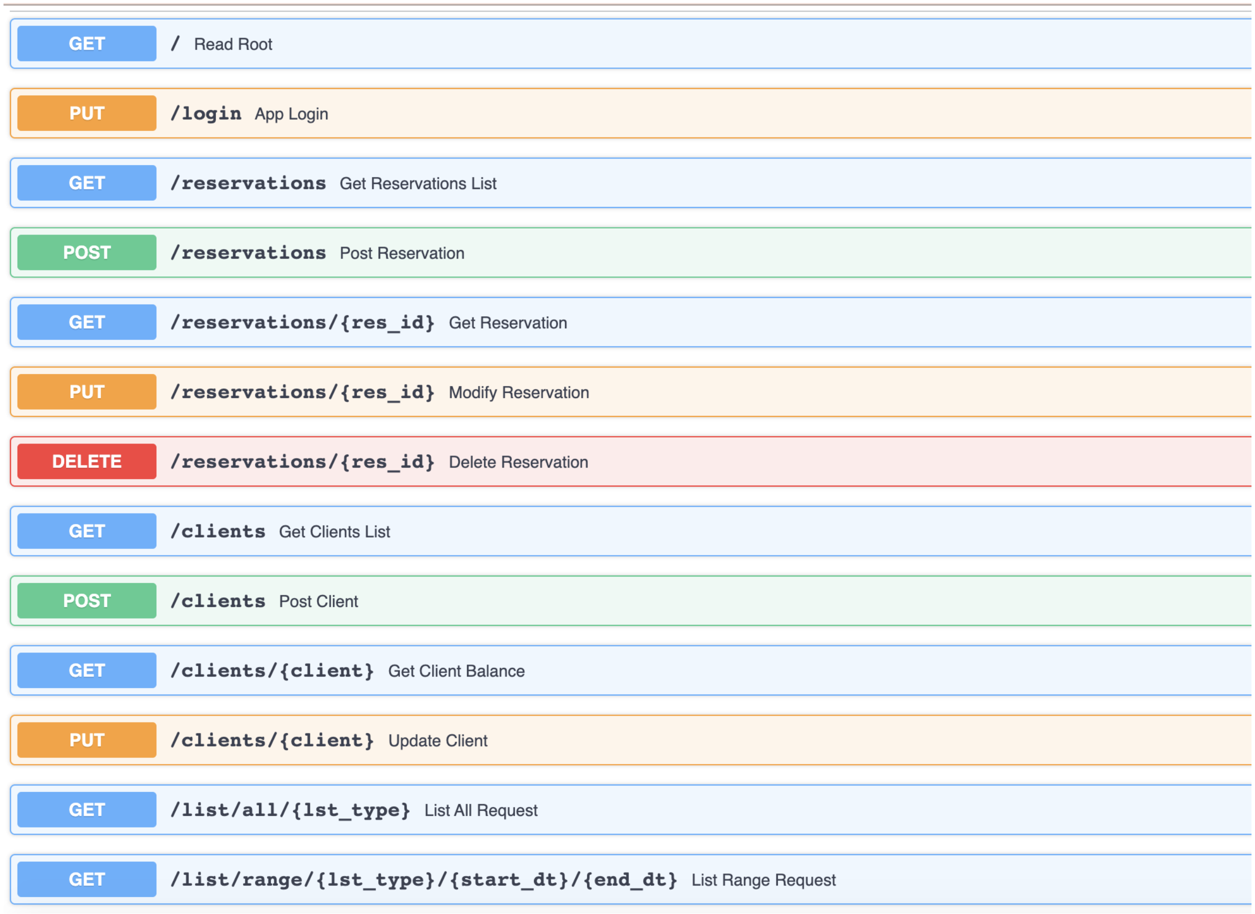
Updated API Documentation for MPCS Reservation System (Iteration 4)

Group 03

Group member: Pierre Pfarr, Kelly Tran, Claire Wang

I. Fast API Endpoints Overview

Updated changes on API endpoint naming convention



GET `/` Read Root

PUT `/login` App Login

GET `/reservations` Get Reservations List

POST `/reservations` Post Reservation

GET `/reservations/{res_id}` Get Reservation

PUT `/reservations/{res_id}` Modify Reservation

DELETE `/reservations/{res_id}` Delete Reservation

GET `/clients` Get Clients List

POST `/clients` Post Client

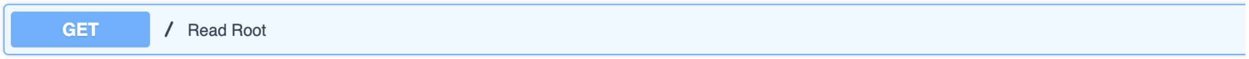
GET `/clients/{client}` Get Client Balance

PUT `/clients/{client}` Update Client

GET `/list/all/{lst_type}` List All Request

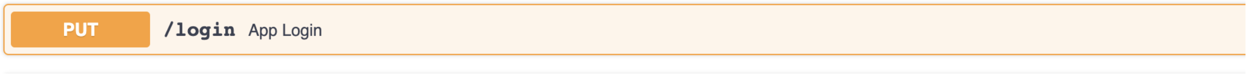
GET `/list/range/{lst_type}/{start_dt}/{end_dt}` List Range Request

II. API Endpoint details



GET `/` Read Root

- Root/home page
- Return a JSON message “Welcome to MPCS Reservations”



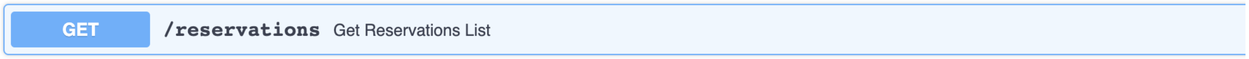
PUT `/login` App Login

- Login page
- Takes in the following parameters:
  - Username
  - Password

Example: PUT /login
Request body

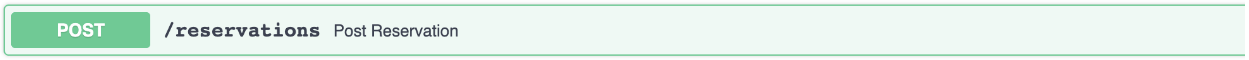
```
{
 "username": "Manager",
 "password": "admin"
}
```

- Returns a JSON format information of the result for authentication.



GET **/reservations** Get Reservations List

- Returns a JSON format information of all current reservation ID numbers in the SQL Lite database.
- Does not take in any parameters.



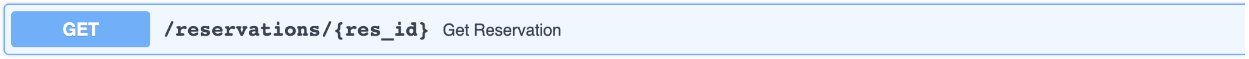
POST **/reservations** Post Reservation

- Reserves a machine or shop time.
- Takes in the following parameters:
  - client - Name of the client (string)
  - request - Machine or workshop to reserve (string)
    workshop1, workshop2, …, workshop4
    mini_microvac1, mini_microvac2
    irradiator1, irradiator2
    polymer_extruder1, polymer_extruder2
    high_velocity_crusher
    gigawatt_lightning_harvester
  - create_date - Day the reservation is made (string, ‘YYYY-MM-DD’’)
  - start_date - The first day of the reservation (string, ‘YYYY-MM-DD’)
  - res_start - The time the reservation begins (string, ‘HH:MM’)
  - res_end - The time the reservation ends (string, ‘HH:MM’)

Example: POST /reservations
Request body

```
{
  "client": "Kelly",
  "request": "workshop2",
  "create_date": "2021-05-28",
  "start_date": "2021-05-28",
  "res_start": "10:00",
  "res_end": "12:00"
}
```

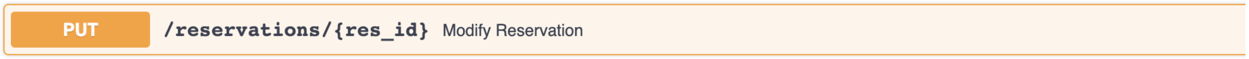
- Returns a JSON that confirms the reservation if it is available, or notifies the user if the reservation is unavailable



GET /reservations/{res_id} Get Reservation

- Get reservation details by the reservation_id
- Take in one parameter: res_id
  Example: GET /reservations/{res_id}
  Parameters
  res_id: 202154385
-

- Returns a JSON format information that has reservation details including client name, request machine, reservation create date, reservation start date, start time, end date, end time, status and total cost
- If the reservation ID is not valid, it will return “no reservation under that id”.



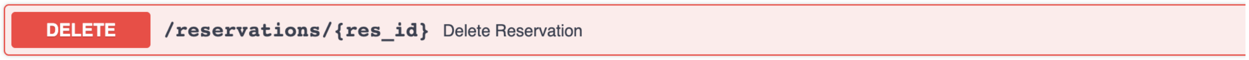
PUT /reservations/{res_id} Modify Reservation

- Modify reservation details by the reservation_id
- Takes in the following parameters:
  - res_id
  - client - Name of the client (string)
  - request - Machine or workshop to reserve (string)
    workshop1, workshop2, …, workshop4
    mini_microvac1, mini_microvac2
    irradiator1, irradiator2
    polymer_extruder1, polymer_extruder2
    high_velocity_crusher
    gigawatt_lightning_harvester
  - create_date - Day the reservation is made (string, ‘YYYY-MM-DD’’)
  - start_date - The first day of the reservation (string, ‘YYYY-MM-DD’)
  - res_start - The time the reservation begins (string, ‘HH:MM’)
  - res_end - The time the reservation ends (string, ‘HH:MM’)

  Example: PUT /reservations/{res_id}
  res_id: 20214923
  Request body

```
{
 "client": "Kelly",
 "request": "workshop3",
 "create_date": "2021-05-28",
 "start_date": "2021-05-28",
 "res_start": "10:00",
  "res_end": "12:00"
}
```

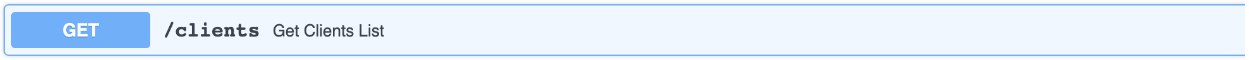
- Returns a JSON format information that has reservation details including client name, request machine, reservation create date, reservation start date, start time, end time, status and total cost
- If the reservation ID is not valid, it will return “no reservation under that id”.



DELETE `/reservations/{res_id}` Delete Reservation

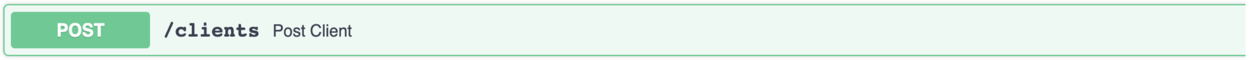
- Deletes a reservation from the internal list of unavailable reservations
- Takes in a reservation ID number
  Example: DELETE /reservations/{res_id}
  Parameters
  res_id: 20214923

- Returns a JSON with the details of the deleted reservation, including the refund cost, client, and reservation times. If the reservation is not in the system, the JSON will say ‘none’



GET `/clients` Get Clients List

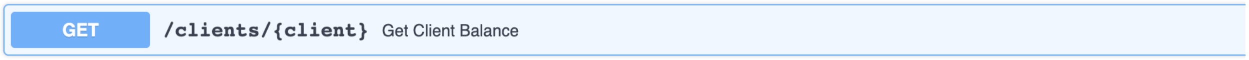
- Get client list
- Does not take in any parameters.
- Returns JSON format information of a list of clients.
- If there is no client available, it will return “{'ids_lst': 'No client'}”



POST `/clients` Post Client

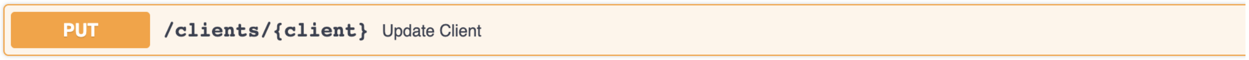
- Create new client
- Take in input parameters *client*, from Client class
  Example: POST /clients
  ```
  {
  "name": "Kelly",
  "password": "hello123",
  "status": "active",
  "balance": 0,
  "role": "admin"
  }
  ```
- Return the status confirmation of “{"Status": "Client Created"}”

- If we cannot create a new client, it will update the status to “{"Status": "Your Reservation is Unavailable"}”



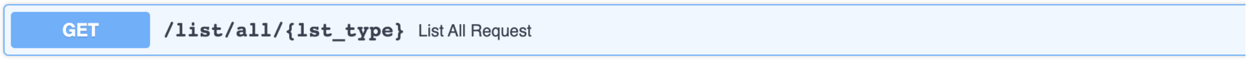
GET `/clients/{client}` Get Client Balance

- Get Client Balance
- Take in parameter:
    - Required path parameter: {client}
    - Optional query parameter: *balance*, “None” by default
        - If the balance is “true”, the system will return the balance for the client
- Returns a JSON message of the result of the client balance in the system
- If the input client name is invalid or not exists, it will return {'Message': 'No client'}



PUT `/clients/{client}` Update Client

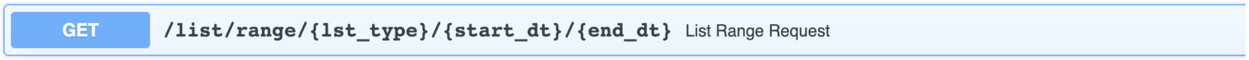
- Update Client information, functions including changing to a new name, load balance to the account, subtract balance from the account, change the status to activate or deactivate of the account
- Take in parameter:
    - Required path parameter: {client}
      (Only one of the four optional options can be used each time)
    - Optional query parameter: {new_name} - can help the client to update to a new name
    - Optional query parameter: {add} - can help client to add balance to the account (limit to 1 - 25,000 balance adding)
    - Optional query parameter: {sub} - subtract this amount from the client’s existing account balance
    - Optional query parameter: {status} - can be “Activate” or “Deactivate”
- Returns a JSON message of confirmation of {"Message": "Client updated"}
- If the input client name is invalid or not exists, it will return {"Message": "Client cannot be updated"}



GET `/list/all/{lst_type}` List All Request

- List all client or list all transaction
- Take in parameter:
    - Required path parameter: {lst_type} - can be two options: “client” or “transactions”
    - Optional query parameter: {client} - if want a list of client, then can take input of client
- If the list type is client:
    - Returns the list of the client result
    - If the client is valid but have no result, it will return {"Message":"No results found"}

    - If the client is invalid, it will return {"Message":"No client was listed"}
- If the list type is transaction:
    - Returns a result list of transaction results
    - If invalid input, returns {"Message":"No Results Found"}



GET `/list/range/{lst_type}/{start_dt}/{end_dt}` List Range Request

- Generates a report for all reservations or clients or transactions within the specified date range
- Takes in three parameters:
    - Lst_type - Type of report. Options can be “client”, “transactions”, or “reservations”
    - Start - Beginning date in the date range (string, ‘YYYY-MM-DD’)
    - End - Ending date in the date range (string, ‘YYYY-MM-DD’)

  Example: GET /list/{lst_type}/{start}/{end}
  Parameters
  lst_type: reservations
  start: 2021-05-10
  end: 2021-05-15

- Returns a JSON that lists all the reservations or transactions between the date range. Transactions will include cancelled reservations and refunds.
- If any type is invalid, it will return a message of {"Message":"No Results Found"}