Computer Network 4074 Project

Introduction

In this project we implemented an http which acts as a room reservation using Python programming language. We have created 3 server for process, roomserver:8087, activityserver:8089, rezervationserver:8088

Project Members

- Hamza Kavak 150118886
- Asaf Talha Gültekin 150119678
- Resul Akçakaya 150119663

Implementation of the servers

All operations are indicated on the flowcharts in a more clear and understandable way.

a. Implementation of the Room Server

1. /add?name=roomname

- Get the parameter value which is indicated 'name'
- Check if it is in the database
- If it is in, return 403 Forbidden and message: Room already exists in database
- Otherwise, add the database, return 200 OK and message: Room with name {} is successfully added

2. /remove?name=roomname

- Get the parameter value which is indicated 'name'
- Check if it is in the database
- If it is not in, return 403 Forbidden and message: Room does not exist in database
- Otherwise, remove the database, return 200 OK and message: Room with name {} is successfully removed

3. /reserve?name=roomname&day=x&hour=y&duration=z

- Check if parameters exist
- If parameters do not exist, return 400 Bad Request and message: Invalid parameters
- If parameters exist, check the room if it is in database

- If it is in, make reservations operations
 - o Check if the day value is between 1 and 7, otherwise return 400 Bad Request
 - o Check if the hour value is between 9 and 17, otherwise return 400 Bad Request
 - Check if the hour+duration value is greater than 17, otherwise return 400 Bad Request
 - o Check if there are any previous reservations in this day and time range.
- If all checks are ok, make the reservation and return 200 OK messages, otherwise, 400 Bad Request messages if there are problems with the parameters, 403 Forbidden messages if a reservation has been made before.

4. /checkavailability?name=roomname&day=x

- First, check that the name and day parameters are not gutted in the url. If not, return 400 Bad Request.
- If there are parameters, check the value of the day parameter, it should be between 1 and 7. If not, return 400 Bad Request.
- If there are parameters and they are valid, check whether there is such a room, if there is no room with that name, return 404 Not Found.
- If there is a room, check the hours between 9-17 and subtract the appropriate hours and return 200 OK.

b. Implementation of the Activity Server

1. /add?name=activityname

- Get the parameter value which is indicated 'name'
- Check if it is in the database
- If it exist it returns 403 Forbidden message: Activity already exists in database!
- Otherwise, add the database, return 200 OK and message: Activity Added, name {}, successfully added

2. /remove?name=activityname

- Get the parameter value which is indicated 'name'
- Check if it is in the database
- if it exists it returns 200 OK message: Activity Removed!', Activity with name {name} is successfully removed.
- Otherwise if it doesn't exist, it returns 403 Forbidden messages: Error', 'Activity does not exist!

3. /check?name=activityname

- Get the parameter value which is indicated 'name'
- Check if it is in the database

- If it exists it return 200 OK message: Activity Exists!', Activity with name {name} exists.
- Otherwise, it returns 403 Forbidden messages: Error', 'Activity does not exist!

c. Implementation of the Reservation Server

- 1. /listavailability?room=roomname
- 2. /listavailability?room=roomname&day=x
 - There are 2 different usage /listavailability methods. It contains a day parameter in one of them. If there is a day parameter, the day is controlled as day+1 with the for loop, if there is no day parameter, all days between 1-7 and 1-7 are controlled.
 - With the established loop, the room sends a request to the server for every day and splits the result from there and combines it.
 - When it exits the loop, it returns with a 200 OK message. If there is a message other than 200, the direct message is returned as a response

To connection room server

```
request = f"GET /checkavailability?name={room}&day={day} HTTP/1.1\r\n"
room_socket.sendall(request.encode())
room_socket.sendall(f"Host: localhost:{room_port}\r\n".encode())
room_socket.sendall(b"\r\n")
room_response = room_socket.recv(1024).decode()
room_socket.close()
```

3. /reserve?room=roomname&activity=activityname&day=x&hour=y&duration=z

- Get the parameter value which is indicated 'reserve'
- it checks for room, day, hour and activity:
 - 1. if any of them doesn't exist it return 400 Bad Request
 - 2. activity server connection is created using a socket request to Check, if the activity exists or not. If there is a status code other than 200, the direct message is returned to the user.
 - 4. If it has activity, it is communicating with the room server. If a 200 OK message is received from here, the reservation process is made and returns 200 OK message. If a different message is received from here (parameters may be invalid), it breaks the incoming message and returns to the user as a new message.

To connection activity server

```
# istek gönder
request = f"GET /check?name={activity} HTTP/1.1\r\n"
activity_socket.sendall(request.encode())
activity_socket.sendall(f"Host: localhost:{activity_port}\r\n".encode())
activity_socket.sendall(b"\r\n")
activity_response = activity_socket.recv(1024).decode()
activity_socket.close()
```

To connection room server

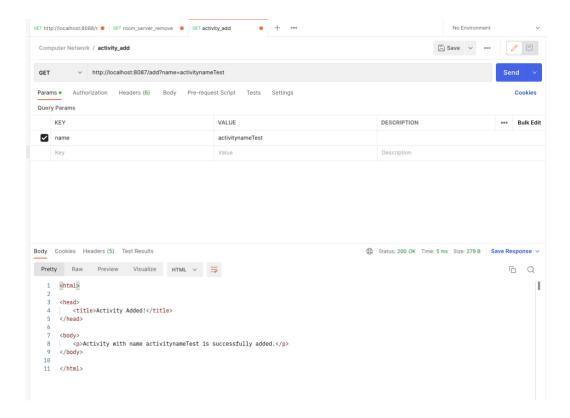
```
# istek gönder
request = f"GET /reserve?name={room}&day={day}&hour={hour}&duration={duration} HTTP/1.1\r\n"
room_socket.sendall(request.encode())
room_socket.sendall(f"Host: localhost:8089\r\n".encode())
room_socket.sendall(b"\r\n")
# cevap al
room_response = room_socket.recv(1024).decode()
# bağlantıyı kapat
room_socket.close()
```

4. /display?id=reservation_id

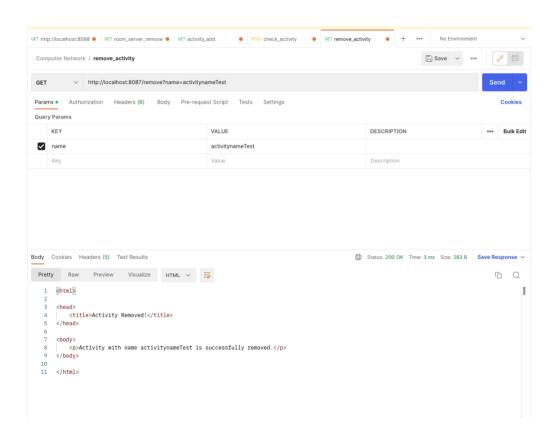
- Get the parameter value which is indicated 'display'
- Checks for id parameter if there isn't in the query string it returns 400 Bad request.
- The id from the reservations is searched, if there is an incoming id, the reservation details are returned with a 200 OK message, if there is no reservation related to that id, 404 Not Found is returned.
- If any other request made using HTTP method other than GET or POST, than it returns 405 Method not allowed message.

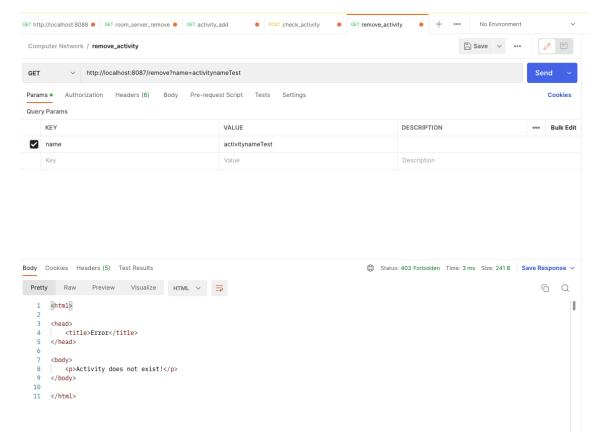
Outputs

Activity Add

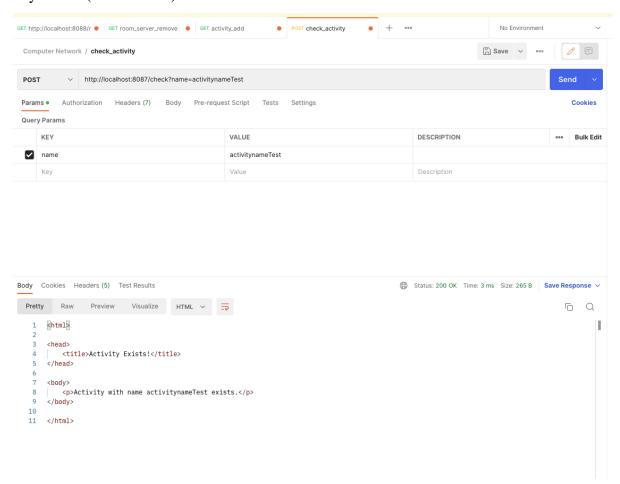


Activity Remove

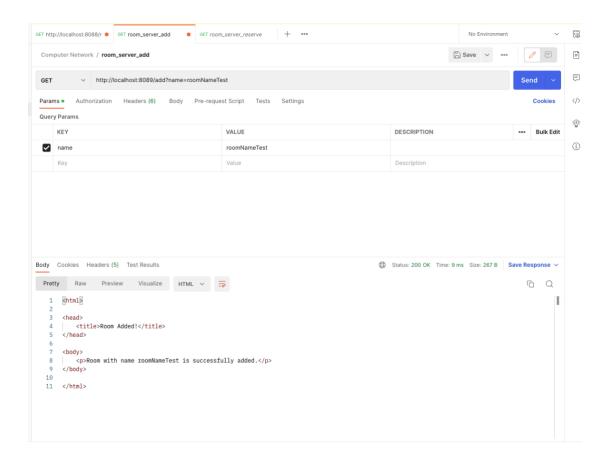




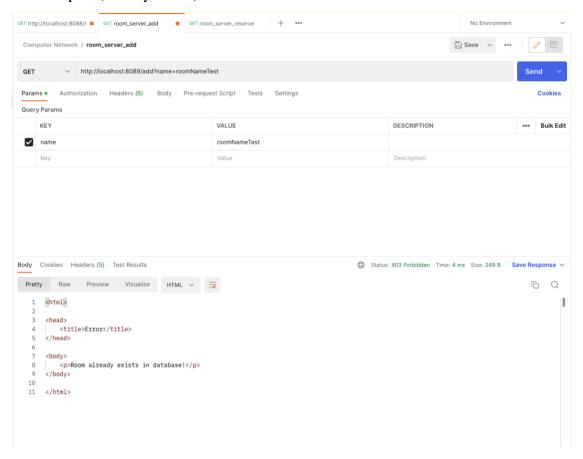
Activity Check (Post method)

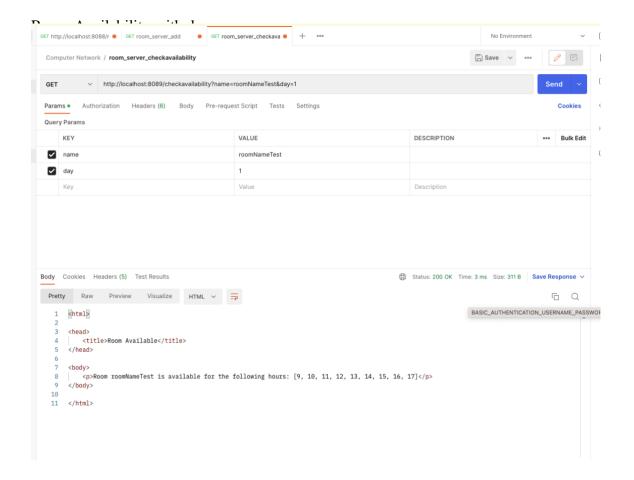


Room Add

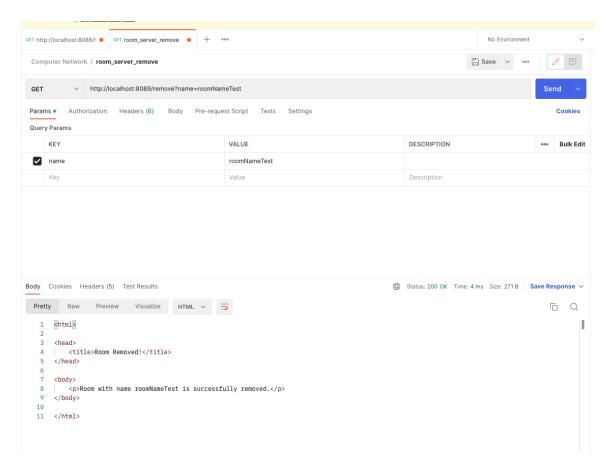


Room Add – Repeat (already added)

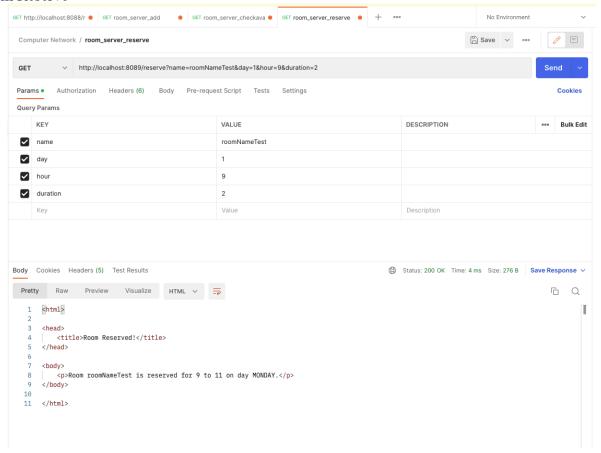




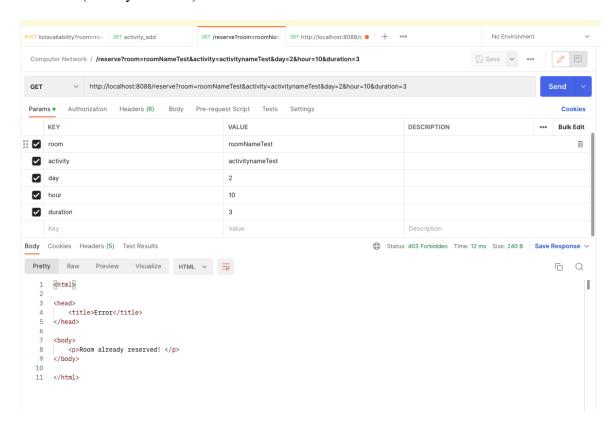
Room Remove



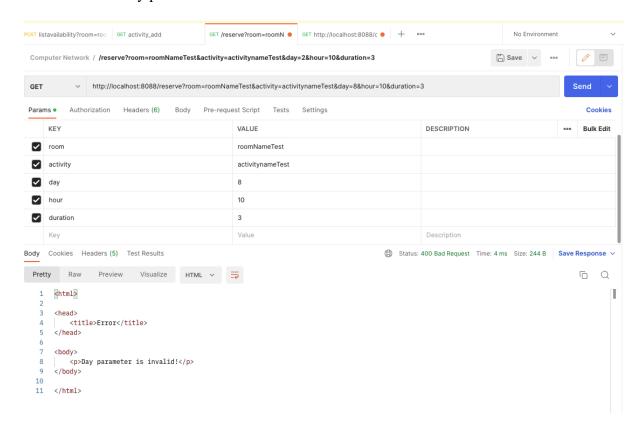
Room Reserve



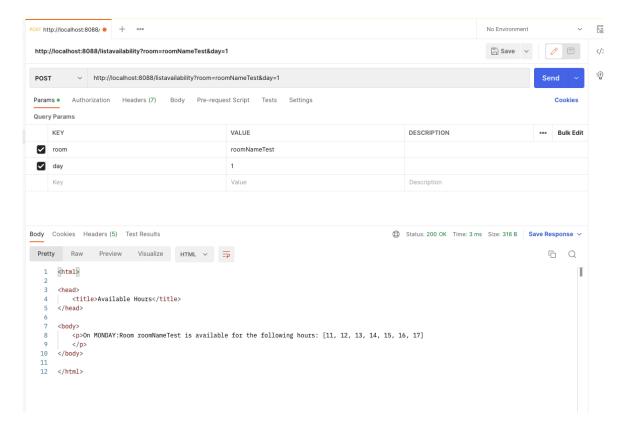
Room Reserve (already reserved)



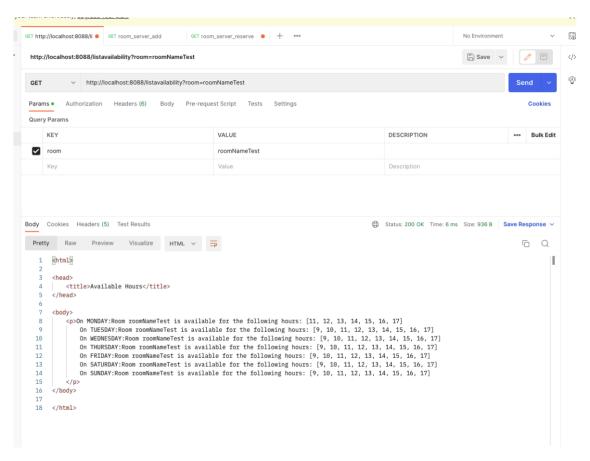
Room Server Error day parameter



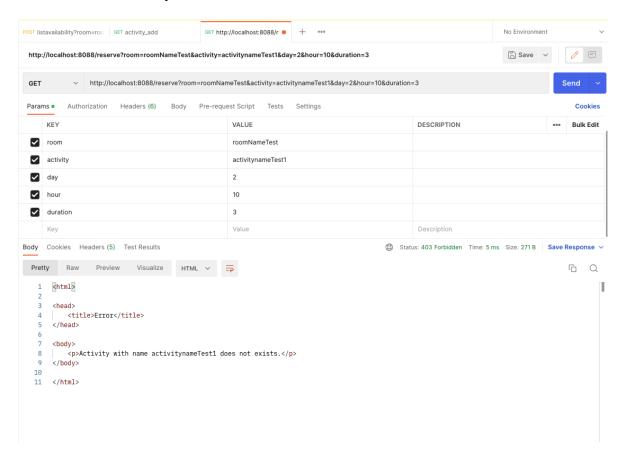
Reservation Server list availability with day parameter



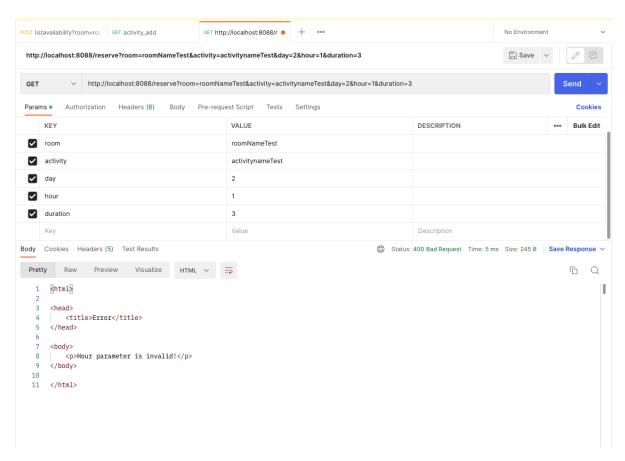
Reservation Server list availability without day parameter



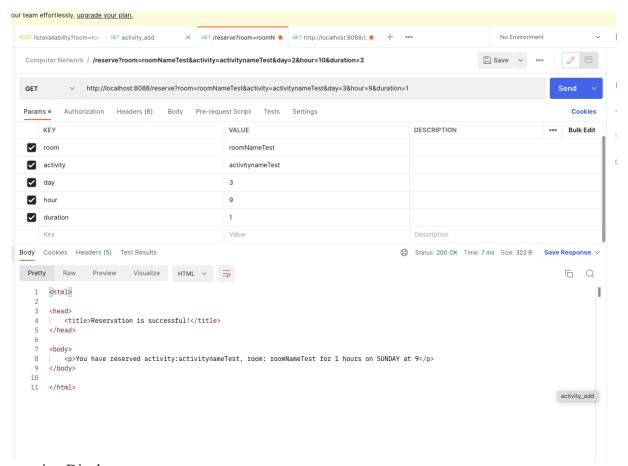
Reservation Server if activity does not exist



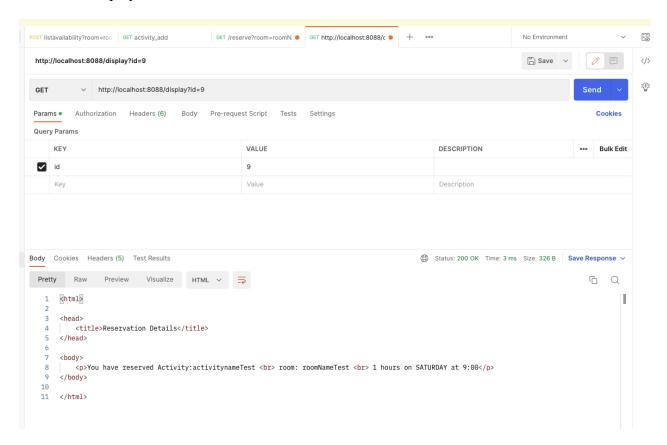
Reservation Server if any of them is/are invalid



Reservation Success



Reservation Display



Extras & Genius Ideas

Project also supports POST method together with the GET method.