USER MANUAL

After deploy the project on a server, we can test the services offered by the projects.

User CURL to post request and receive response detail directly on terminal (you can also use either SOAP UI or POSTMAN as an alternative for testing REST API), open terminal and use these syntaxes for each service

Some notes: secret.key=q1w2e3r4 // this is the key set up by administrator

1. Find All User

Syntax: curl –v localhost:8080/users

Result: Response’s body is an array of all user in JSON format

Header status code: 200

1. Delete a user

Syntax: curl -v -X DELETE localhost:8080/user/{secret.key}/{login}

Result: success with status code 200 and empty body

Result: error in the case of no login found. status code 500, body with message contain the login not found are returned

1. Creating new user

Syntax: curl -v -X POST localhost:8080/users/{secret.key} -H "Content-type:application/json" -d "{\"login\":\"loginString\",\"name\":\"nameString\",\"surname\":\"surnameString\",\"password\":\"passwordString\"}"

NOTE: Body of the syntax contain the user object in JSON

Result: success with status code 200, response’s body is the user object in JSON

Result: error in the case of a login found. status code 500, body with message contain the login found are returned

1. Update a user

Syntax: curl -v -X PATCH localhost:8080/users/{secret.key}/{login} -H "Content-type:application/json" -d "{\"login\":\"jsmith\",\"name\":\"tu\",\"surname\":\"Nguyen\"}"

NOTE: Body of the syntax contain the updating user object in JSON

Result: success with status code 200, response’s body is the updating user object in JSON

Result: error in the case of no login found. status code 500, body with message contain the login not found are returned

1. Find All Room

Syntax: curl –v localhost:8080/rooms

Result: Response’s body is an array of all room in JSON format

Header status code: 200

1. Delete a Room

Syntax: curl -v -X DELETE localhost:8080/rooms/{secret.key}/{name}

Result: success with status code 200 and empty body

Result: error in the case of no room name found. status code 500, body with message contain the room name not found are returned

Note: you cant pass space in url instead you must replace space by %20 in url

For example: curl -v -X DELETE localhost:8080/rooms/q1w2e3r4/Small%20Room

1. Creating new Room

Syntax: curl -v -X POST localhost:8080/rooms/{secret.key} -H "Content-type:application/json" -d "{\"name\":\"ROOMNAME\",\"location\":\"LOCATION\",\"seats\":\"NUMBEROFSEATS\",\"projector\":\"PROJECTOR\", \"phonenumber\":\"THEPHONENUMBER\" }"

NOTE: Body of the syntax contain the room object in JSON, capital letters are variables and must be replaced in above url, NUMBEROFSEATS should be a number and PROJECTOR should be yes or no, otherwise and error will rise and a status code 404 will be returned

Result: success with status code 200, response’s body is the room object in JSON

Result: error in the case of a room with the same name found. status code 500, body with message contain the room name found are returned

1. Update a user

Syntax: curl -v -X PATCH localhost:8080/rooms/{secret.key}/{name} -H "Content-type:application/json" -d "{\"name\":\"ROOMNAME\",\"location\":\"LOCATION\",\"seats\":\"NUMBEROFSEATS\",\"projector\":\"PROJECTOR\", \"phonenumber\":\"THEPHONENUMBER\" }"

NOTE: Body of the syntax contain the updating user object in JSON

Result: success with status code 200, response’s body is the updating room object in JSON

Result: error in the case of no room name found. status code 500, body with message contain the room name not found are returned

1. Booking a reservation

Syntax: curl -v –x POST localhost:8080/booking/book -H "Content-type:application/json" -d "{\"roomName\":\"ROOMNAME\",\"login\":\"LOGIN\",\"password\":\"PASSWORD\",\"from\":\"YYYY-MM-DD HH:MM:SS\",\"to\":\"YYYY-MM-DD HH:MM:SS\"}"

For example: curl -v –x POST localhost:8080/booking/book -H "Content-type:application/json" -d "{\"roomName\":\"Small Room\",\"login\":\"jsmith\",\"password\":\"qwerty\",\"from\":\"2019-10-28 11:11:11\",\"to\":\"2019-10-29 00:00:00\"}"

Note: the body is the reservation object.

1. List all reservation

Syntax: curl –v localhost:8080/booking

Result: Response’s body is an array of all reservation in JSON format

Header status code: 200

1. Getting booking schedule for all rooms

Syntax: curl –v –X POST localhost:8080/booking/allrooms -H "Content-type:application/json" -d "{\"from\":\"FROMDATE\",\"to\":\"TODATE\"}"

For example: curl –v –X POST localhost:8080/booking/allrooms -H "Content-type:application/json" -d "{\"from\":\"2019-10-28 10:10:10\",\"to\":\"2019-10-29 00:00:00\"}"

Note: call consist of a time frame with from or to, they maybe empty

Date must be in format: yyyy-MM-dd HH:mm:ss for example: 2019-10-28 10:10:10

1. Getting booking schedule for single room

Syntax: curl –v –X POST localhost:8080/booking/listbyroom/{roomName} -H "Content-type:application/json" -d "{\"from\":\"FROMDATE\",\"to\":\"TODATE\"}"

For example: curl –v –X POST localhost:8080/booking/listbyuser/jsmith -H "Content-type:application/json" -d "{\"from\":\"2019-10-28 10:10:10\",\"to\":\"2019-10-29 00:00:00\"}"

Note: call consist of a time frame with from or to, they maybe empty

Date must be in format: yyyy-MM-dd HH:mm:ss

1. Getting booking schedule for a user

Syntax: curl –v –X POST localhost:8080/booking/listbyuser/{login} -H "Content-type:application/json" -d "{\"from\":\"FROMDATE\",\"to\":\"TODATE\"}"

For example: curl –v –X POST localhost:8080/booking/listbyuser/jsmith -H "Content-type:application/json" -d "{\"from\":\"2019-10-28 10:10:10\",\"to\":\"2019-10-29 00:00:00\"}"

Note: call consist of a time frame with from or to, they maybe empty

Date must be in format: yyyy-MM-dd HH:mm:ss