Postman test outputs 2

JUnit Test Run Output for methods in controller 8

# Postman test outputs

POST API endpoint testing:

1) curl --location 'http://localhost:8080/api/robotposition/createRobotPosition' \

--header 'Content-Type: application/json' \

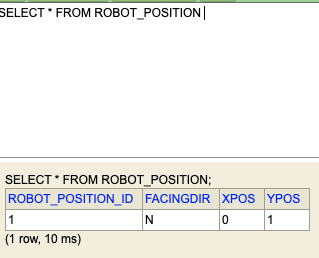
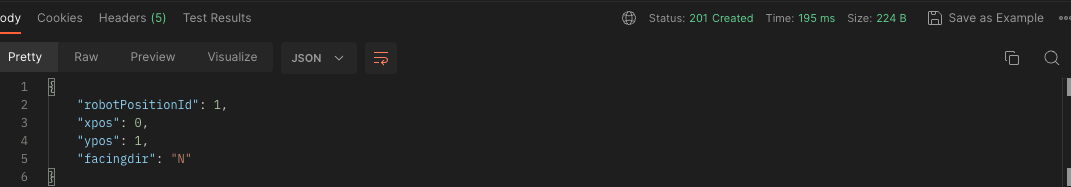
--data '{

"xpos" : 0,

"ypos" : 1,

"facingdir": "N"

}’



2) curl --location 'http://localhost:8080/api/robotposition/createRobotPosition' \

--header 'Content-Type: application/json' \

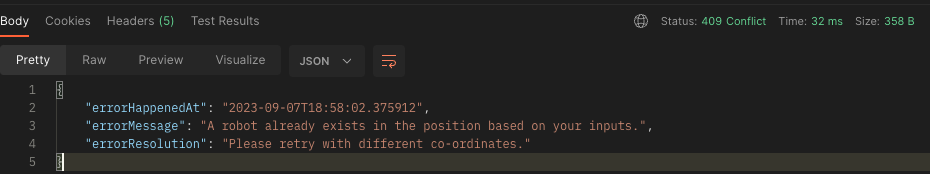
--data '{

"xpos" : 0,

"ypos" : 1,

"facingdir": "N"

}’



3) Add few more valid records.

curl --location 'http://localhost:8080/api/robotposition/createRobotPosition' \

--header 'Content-Type: application/json' \

--data '{

"xpos" : 0,

"ypos" : 0,

"facingdir": "W"

}'

curl --location 'http://localhost:8080/api/robotposition/createRobotPosition' \

--header 'Content-Type: application/json' \

--data '{

"xpos" : 4,

"ypos" : 4,

"facingdir": "W"

}'

curl --location 'http://localhost:8080/api/robotposition/createRobotPosition' \

--header 'Content-Type: application/json' \

--data '{

"xpos" : 3,

"ypos" : 4,

"facingdir": "S"

}'

curl --location 'http://localhost:8080/api/robotposition/createRobotPosition' \

--header 'Content-Type: application/json' \

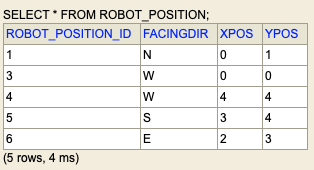
--data '{

"xpos" : 2,

"ypos" : 3,

"facingdir": "E"

}'



4) Test with invalid co-ordinates and facing direction

curl --location 'http://localhost:8080/api/robotposition/createRobotPosition' \

--header 'Content-Type: application/json' \

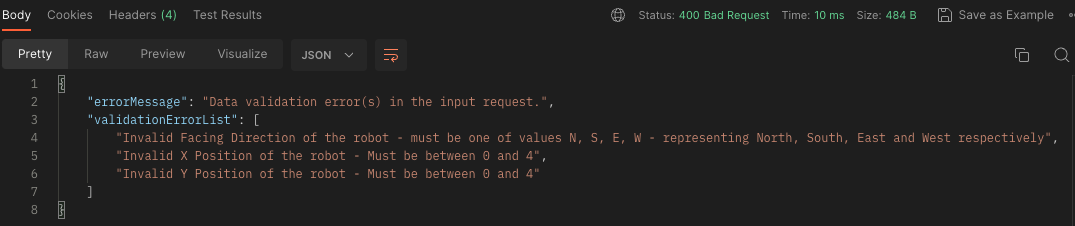
--data '{

"xpos" : 10,

"ypos" : 5,

"facingdir": "Northern"

}'

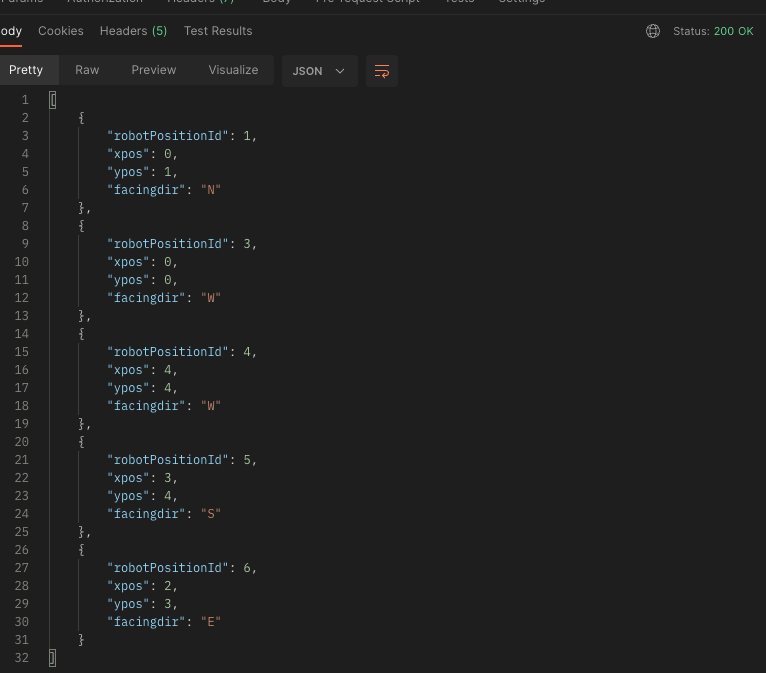


Get api endpoints testing:

1. Retrieve a report of all robot positions:

curl --location 'http://localhost:8080/api/robotposition/report' \

--data ‘'



2) Invalid robot position id in the URL

curl --location --request GET 'http://localhost:8080/api/robotposition/1.2' \

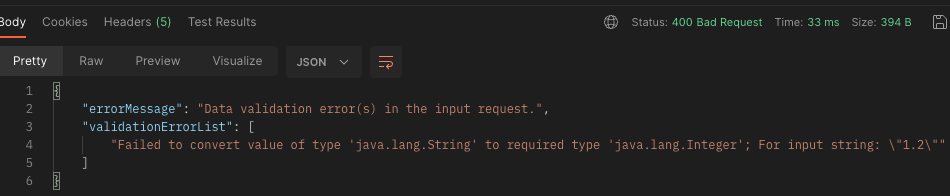
--header 'Content-Type: application/json' \

--data '{

"robotPositionId": 3,

"robotPositionCommands" : "MOVE"

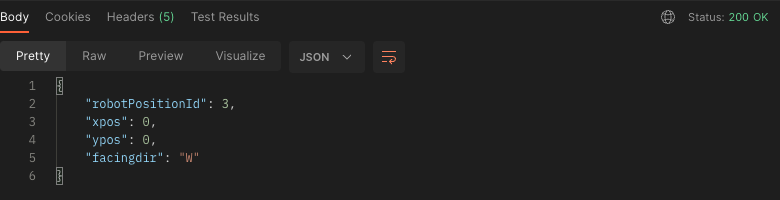
}'



3) Valid robot position id in the URL

curl --location 'http://localhost:8080/api/robotposition/3' \

--data ''

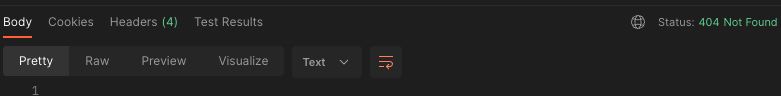


Delete api endpoint testing:

1) Delete an id that doesn’t exist.

curl --location --request DELETE 'http://localhost:8080/api/robotposition/10' \

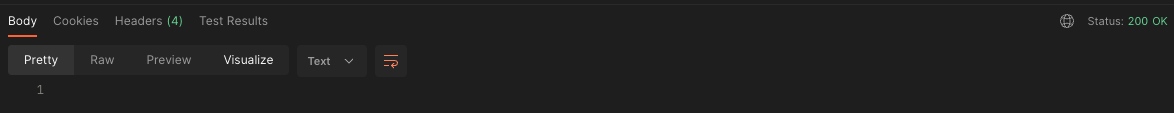
--data ‘'



2) Delete an id that exists.

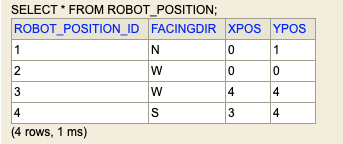
curl --location --request DELETE 'http://localhost:8080/api/robotposition/6' \

--data ‘'



UPDATE API ENDPOINT TESTING:

Fresh data setup again for Update API testing



1) Invalid data in the request:

curl --location --request PUT 'http://localhost:8080/api/robotposition/updateRobotPosition' \

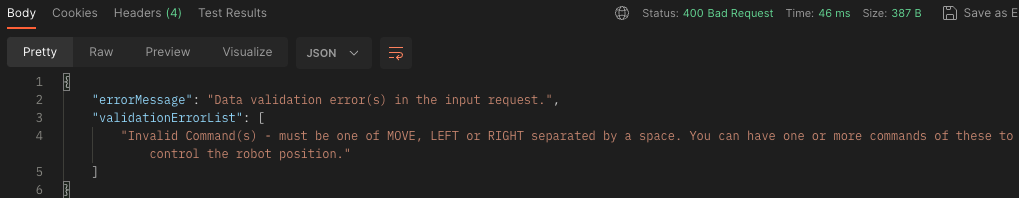
--header 'Content-Type: application/json' \

--data '{

"robotPositionId": 1,

"robotPositionCommands" : "RIGHT MOE LET MOVE MOVE RITE"

}'



2) Robot position id not present in the table:

curl --location --request PUT 'http://localhost:8080/api/robotposition/updateRobotPosition' \

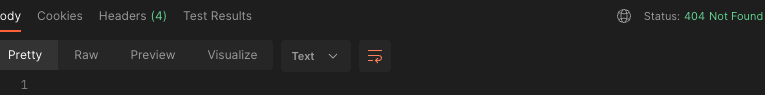
--header 'Content-Type: application/json' \

--data '{

"robotPositionId": 10,

"robotPositionCommands" : "RIGHT"

}'



3) Try to update the robot position id 2 with MOVE command. This is illegal as the robot is west facing and cannot fall off the table.

curl --location --request PUT 'http://localhost:8080/api/robotposition/updateRobotPosition' \

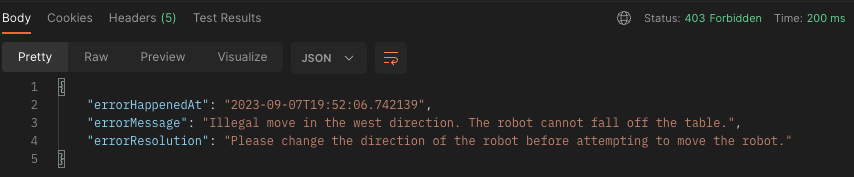
--header 'Content-Type: application/json' \

--data '{

"robotPositionId": 2,

"robotPositionCommands" : "MOVE"

}'



4) Try to update the robot position id 2 with MOVE command, but change the direction with RIGHT before that. This should make the robot north facing and then is moved. However robot id 1 is already in that position. Hence error.

curl --location --request PUT 'http://localhost:8080/api/robotposition/updateRobotPosition' \

--header 'Content-Type: application/json' \

--data '{

"robotPositionId": 2,

"robotPositionCommands" : "RIGHT MOVE"

}'



5) Try to update the robot position id 2 with MOVE command, but change the direction with RIGHT and RIGHT commands before that. After moving, turn the robot left

curl --location --request PUT 'http://localhost:8080/api/robotposition/updateRobotPosition' \

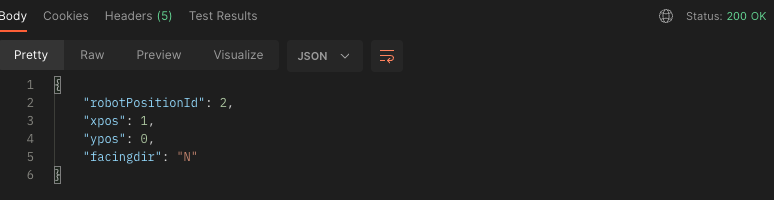
--header 'Content-Type: application/json' \

--data '{

"robotPositionId": 2,

"robotPositionCommands" : "RIGHT RIGHT MOVE LEFT"

}'



DB record #2 updated as well.

# JUnit Test Run Output for methods in controller

# pasted-image.png

