| Use cases | REST API realisation | REST API endpoints | API request body inputs | Data model | Processing  Exceptions | Success response body | HTTP Status response |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PLACE x,y, facingdirection  e.g. PLACE 0,0,NORTH | This command will **create** a record in the robot\_position table and return a unique id for the given robot position. | **POST** /api/robotposition/createRobotPosition | •XPOS  •YPOS  •FACINGDIR | ROBOT\_POSITION | If an identical record already exists, it is reported as duplicate robot position. | •ROBOT\_POSITION.ROBOT\_POSITION\_ID  •ROBOT\_POSITION.XPOS  •ROBOT\_POSITION.YPOS  •ROBOT\_POSITION.FACINGDIR | **201** (Record created)  **409**  (Conflict - when identical record exists) |
| REPORT | This command will **get** **all** the robot positions from the table. | **GET** /api/robotposition/report | None | ROBOT\_POSITION |  | •ROBOT\_POSITION.ROBOT\_POSITION\_ID  •ROBOT\_POSITION.XPOS  •ROBOT\_POSITION.YPOS  •ROBOT\_POSITION.FACINGDIR | **200** (Records retrieved)  **404**  (No record found) |
| PLACE x,y, facingdirection, MOVE, LEFT, RIGHT | This command will **update** the robot position based on the MOVE. LEFT, RIGHT actions. | **PUT** /api/robotposition/updateRobotPosition | * ROBOT\_POSITION\_ID * Robot position commands (LEFT, RIGHT, MOVE) | ROBOT\_POSITION | 1. During the processing of the input robot position commands, if the new calculated position is already in the table, then it is reported as duplicate robot position. 2. During the processing of the MOVE command, if the robot position is going out of the 5\*5 grid, then it is reported as illegal move. | Updated record with new position values:  •ROBOT\_POSITION.ROBOT\_POSITION\_ID  •ROBOT\_POSITION.XPOS  •ROBOT\_POSITION.YPOS  •ROBOT\_POSITION.FACINGDIR | **200**  (Successful update)  **409**  (Conflict - when identical record exists)  **403**  (Illegal move - the robot cannot fall off the table in any direction  )  **404**  (Record not found for the given id) |
|  | **Get** a robot position using its id | **GET** /api/robotposition/{id} | None | ROBOT\_POSITION |  | •ROBOT\_POSITION.ROBOT\_POSITION\_ID  •ROBOT\_POSITION.XPOS  •ROBOT\_POSITION.YPOS  •ROBOT\_POSITION.FACINGDIR | **200** (Record retrieved)  **404**  (No record found) |
|  | **Delete** a robot position using its id | **DELETE** /api/robotposition/{id} | None | ROBOT\_POSITION |  | None | **200** (Record deleted)  **404**  (No record found) |

| REST API endpoint | Request body | Validation errors on the input fields in the request body | Processing errors with resolution |
| --- | --- | --- | --- |
| **POST** /api/robotposition/createRobotPosition | •XPOS  •YPOS  •FACINGDIR | * Invalid X Position of the robot - Must be between 0 and 4 * Invalid Y Position of the robot - Must be between 0 and 4 * Invalid Facing Direction of the robot - must be one of values N, S, E, W - representing North, South, East and West respectively | * A robot already exists in the position based on your inputs. Please retry with different co-ordinates. |
| **PUT** /api/robotposition/updateRobotPosition | * ROBOT\_POSITION\_ID * Robot position commands (LEFT, RIGHT, MOVE) | • Invalid Command(s) - must be one of MOVE, LEFT or RIGHT separated by a space. You can have one or more commands of these to control the robot position. | * A robot already exists in the position based on your inputs. Please retry with different co-ordinates. * Illegal move in the {north, south, east or west} direction. The robot cannot fall off the table. Please change the direction of the robot before attempting to move the robot. |

**DATABASE DETAILS:**

|  |  |
| --- | --- |
| Database name | H2 (in-memory DB). Automatically set up by Spring Boot when starting the application |
| DB Table | robot\_position |
| Table constraints | Automatically set up based on the JPA configuration:  alter table if exists robot\_position add constraint UniqueRobotPosition unique (xpos, ypos).  This is necessary to avoid the creation/update of more than one robot in the same position |
| DB console to run SQL, etc | http://localhost:8080/h2-console/  JDBC URL: jdbc:h2:mem:testdb  Username:sa  Leave the password field blank |