SR-API

(Service Robot API)

Solution Validation Test Guide

Table of content

[1. Document History 3](#_Toc112173016)

[2. Document Purpose 3](#_Toc112173017)

[3. Abbreviations 3](#_Toc112173018)

[4. Tested solution 4](#_Toc112173019)

[4.1. Date 4](#_Toc112173020)

[4.2. Solution Provider 4](#_Toc112173021)

[4.3. Tested solution 4](#_Toc112173022)

[5. Recommended Solution Design Flow 5](#_Toc112173023)

[6. Service Robot API Solution Validation Test Guide 6](#_Toc112173024)

[6.1. Setup 6](#_Toc112173025)

[6.2. Elevator mode check 6](#_Toc112173026)

[6.3. Elevator Call Giving 7](#_Toc112173027)

[6.4. Multiple groups and call giving 10](#_Toc112173028)

[6.5. Integration with access control and call giving 11](#_Toc112173029)

[6.6. Location control and call giving 12](#_Toc112173030)

[6.7. Elevator locks and call giving. 14](#_Toc112173031)

[6.8. Device disabling and call giving. 14](#_Toc112173032)

[6.9. Other system functionality checks 15](#_Toc112173033)

[7 Solution under test details 17](#_Toc112173034)

[7.1 Solution description:- 17](#_Toc112173035)

[7.2 Use case:- 17](#_Toc112173036)

[7.3 Sequence diagram:- 17](#_Toc112173037)

[8 Appendix 18](#_Toc112173038)

[8.1 Grading (KONE use only) 18](#_Toc112173039)

[8.2 Attribute use case 18](#_Toc112173040)

# Document History

|  |  |  |
| --- | --- | --- |
| Date | Description | Author |
| Feb 15, 2021 | KONE Service Robot API 1.0.0 | KONE API Support |
| Nov 08, 2021 | updated with new developer experience | KONE API Support |
| August 05 2022 | KONE Elevator Call API 2.0 | KONE API Support |
| July 28 2023 | Test cases improvements | KONE API Support |

# Document Purpose

Ensuring the quality and security of a solution is every developer’s responsibility. This document gives guidance on evaluating the readiness of those solutions that use Service Robot API. Practical examples of the expected testing support you in the validation. You can also use this guide as a template for documenting test results.

The test environment used in these test scenarios has been designed to resemble an elevator setup in a building. Solution validation is needed before connecting a solution to KONE equipment.

# Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| ACS | Access control server |
| COP | Car operation panel |
| DCS | Destination Control System |
| DOP | Destination operation panel |
| ETA | Estimated time of arrival. Time it will take for a robot to get to the elevator. |
| LCS | Landing call station |
| SR-API | Service Robot Application Programming Interface |

# Tested solution

## Date

|  |  |
| --- | --- |
| Test Date (dd.mm.yyyy): |  |

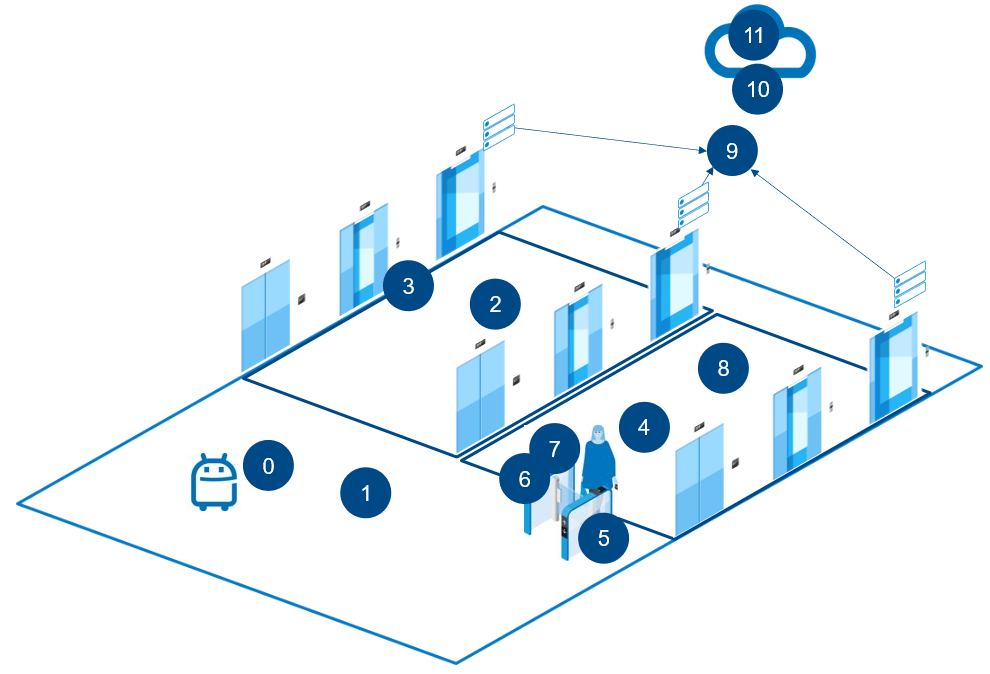
## Solution Provider

|  |  |
| --- | --- |
| Company name |  |
| Company address |  |
| Contact person name |  |
| Email |  |
| Telephone number |  |
| Tester |  |

## Tested solution

|  |  |
| --- | --- |
| System name: |  |
| System version: |  |
| Software name: |  |
| Software version: |  |
| KONE SR-API: | v2.0 |
| KONE test assistant email |  |

# Recommended Solution Design Flow



|  |  |
| --- | --- |
| **Flow** | **Description** |
| 0 | Robot is at an acceptable proximity to the elevator |
| 1 | Elevator mode check (actions taken before making a call to evaluate elevator readiness) |
| 2 | Parking orbit for automatic call |
| 3 | Multiple calls prevention |
| 4 | Robot detection behind a barrier such as turnstile |
| 5 | Barrier control such as turnstile gate if no other system operates it |
| 6 | Integration to access control system for calls generated at barrier passage by access control system |
| 7 | Integration to access control system for calls generated at barrier passage by robot |
| 8 | Integration to access control system for personalized DCS calls generated by robot |
| 9 | Multiple elevator banks support |
| 10 | Cybersecurity measures |
| 11 | Logs |

# Service Robot API Solution Validation Test Guide

These tests focus on making elevator calls with Service Robot API.

## Setup

|  |  |
| --- | --- |
| Pre-Test Setup | Get access to the equipment for testing   * Virtual equipment, available in KONE API portal * Preproduction equipment, by contacting KONE API Support (api-support@kone.com) |
| Expected result | * Test environments available for the correct KONE API organization. * Building id can be retrieved (/resource endpoint). |

|  |  |
| --- | --- |
| Test 1 | Solution initialization |
| Expected result | * Connections established by solution to test environment (Virtual or Preproduction).   + Authentication successful   + Get resources successful * Building config can be obtained.   + Response code 200   + Response code 401 in case if there is issue with API Credentials * Building actions can be obtained.   + Response code 200   + Response code 401 in case if there is issue with API Credentials |
| Test result | Result: Pass  Status: PASS  Description: Successful call to config, actions, ping APIs  Duration: 3267.1499252319336 ms  Response Data:  {  "phase": "ping\_response",  "data": {  "data": {  "request\_id": 480055306,  "time": "2025-08-17T02:33:50.914Z"  },  "callType": "ping",  "buildingId": "building:L1QinntdEOg",  "groupId": "1"  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: common-api/config  Request Parameters:  {  "type": "common-api",  "buildingId": "building:L1QinntdEOg",  "callType": "config",  "groupId": "1" }  Response Data:  [  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": "878264311",  "statusCode": 201,  "data": {  "time": "2025-08-17T02:33:49.782Z"  }  } ]  Status Code: 201  Timestamp: 2025-08-17T02:33:49.954950+00:00  Call 2:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: common-api/actions  Request Parameters:  {  "type": "common-api",  "buildingId": "building:L1QinntdEOg",  "callType": "actions",  "groupId": "1" }  Response Data:  [  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": "455268152",  "statusCode": 201,  "data": {  "time": "2025-08-17T02:33:50.302Z"  }  } ]  Status Code: 201  Timestamp: 2025-08-17T02:33:50.417988+00:00  Call 3:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: common-api/ping  Request Parameters:  {  "type": "common-api",  "buildingId": "building:L1QinntdEOg",  "callType": "ping",  "groupId": "1" }  Response Data:  [  {  "data": {  "request\_id": 480055306,  "time": "2025-08-17T02:33:50.914Z"  },  "callType": "ping",  "buildingId": "building:L1QinntdEOg",  "groupId": "1"  } ]  Status Code: 200  Timestamp: 2025-08-17T02:33:51.366287+00:00  Request Parameters:  {  "type": "common-api",  "buildingId": "building:L1QinntdEOg",  "callType": "config",  "groupId": "1" }  Request Timestamp: 1755398028.099595  Response Timestamp: 1755398031.366745 |

## Elevator mode check

|  |  |
| --- | --- |
| Test 2 | Is the elevator mode operational or not?  Note: elevator mode is set to non-operational. Use if applicable to robot use case |
| Expected result | * Elevator mode is true with any of the below   + Fire mode (FRD)   + Out of service mode (OSS)   + Attendant mode (ATS)   + Priority mode (PRC) * call is not made |
| Test result | Result: Pass  Status: PASS  Description: Subscribe lift\_+/status, check lift\_mode non-operational  Duration: 391.7839527130127 ms  Response Data:  {  "phase": "subscribe\_response",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": "1755398031367",  "statusCode": 201,  "data": {  "time": "2025-08-17T02:33:51.632Z"  }  } }  Request Parameters:  {  "type": "site-monitoring",  "buildingId": "building:L1QinntdEOg",  "callType": "monitor",  "groupId": "1",  "requestId": "1755398031367",  "payload": {  "sub": "mode\_test\_1755398031",  "duration": 60,  "subtopics": [  "lift\_+/status"  ]  } }  Request Timestamp: 1755398031.36681  Response Timestamp: 1755398031.758594 |

|  |  |
| --- | --- |
| Test 3 | Is the elevator mode operational or not?  Note: elevator is set to operational; Source (any floor) – Destination (any floor) |
| Expected result | * Elevator mode is false with all below   + Fire mode (FRD)   + Out of service mode (OSS)   + Attendant mode (ATS)   + Priority mode (PRC) * Call accepted and elevator moving   + Response code 201   + Session id returned   + Elevator destination is correct and as requested |
| Test result | Result: Pass  Status: PASS  Description: lift\_mode operational, basic elevator call successful  Duration: 3488.800048828125 ms  Response Data:  {  "phase": "call\_response",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 335420256,  "statusCode": 201,  "data": {  "request\_id": 335420256,  "success": true,  "session\_id": 283  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 283  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "6f0ac966-690d-4a2c-9c06-f99ec054d6e4",  "area": 1000,  "time": "2025-08-17T02:33:51.758915Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Response Data:  [  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 335420256,  "statusCode": 201,  "data": {  "request\_id": 335420256,  "success": true,  "session\_id": 283  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 283  } ]  Status Code: 201  Timestamp: 2025-08-17T02:33:55.247074+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "6f0ac966-690d-4a2c-9c06-f99ec054d6e4",  "area": 1000,  "time": "2025-08-17T02:33:51.758915Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Request Timestamp: 1755398031.758652  Response Timestamp: 1755398035.247452 |

## Elevator Call Giving

|  |  |
| --- | --- |
| Test 4 | Call: Basic call - > Source: any floor, Destination: any floor  Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking   + Floor markings are as expected   + Floor order is as expected   + Elevator destination is correct as requested |
| Test result | Result: Pass  Status: PASS  Description: Valid action/destination, returns 201+session\_id  Duration: 856.9767475128174 ms  Response Data:  {  "phase": "session\_id",  "data": {  "session\_id": 284  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": 425723244,  "area": 3000,  "time": "2025-08-17T02:33:55.247732Z",  "terminal": 1,  "call": {  "action": 2002,  "activate\_call\_states": [  "being\_fixed"  ],  "destination": 5000  }  } }  Response Data:  [  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 134723893,  "statusCode": 201,  "data": {  "request\_id": 134723893,  "success": true,  "session\_id": 284  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 284  } ]  Status Code: 201  Timestamp: 2025-08-17T02:33:56.104130+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": 425723244,  "area": 3000,  "time": "2025-08-17T02:33:55.247732Z",  "terminal": 1,  "call": {  "action": 2002,  "activate\_call\_states": [  "being\_fixed"  ],  "destination": 5000  }  } }  Request Timestamp: 1755398035.2475111  Response Timestamp: 1755398036.104488 |

|  |  |
| --- | --- |
| Test 5 | Call: hold open elevator door - > at Source floor, at Destination floor  Note: Use if applicable to robot use case Landing Call – Source only, Car Call – Destination only |
| Expected result | * Elevator door stays open for   + duration specified in hard time   + optionally plus duration specified in soft time |
| Test result | Result: Fail  Status: FAIL  Description: hold\_open successful, door status sequence correct  Duration: 349.40409660339355 ms  Error Message: Hold open command rejected with 403 - Token does not contain required scope for L1QinntdEOg:1  Response Data:  {  "phase": "hold\_open\_response",  "data": {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 962667662,  "statusCode": 403,  "data": {  "error": "Token does not contain required scope for L1QinntdEOg:1"  }  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action/hold\_open  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "area": 1000,  "call": {  "action": 1002,  "lift\_deck": 1000,  "hard\_time": 5,  "soft\_time": 10  }  } }  Response Data:  [  {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 962667662,  "statusCode": 403,  "data": {  "error": "Token does not contain required scope for L1QinntdEOg:1"  }  } ]  Status Code: 403  Timestamp: 2025-08-17T02:33:56.453553+00:00  Request Timestamp: 1755398036.10455  Response Timestamp: 1755398036.453954 |

|  |  |
| --- | --- |
| Test 6 | Call: Action call with action id = 200, 0 [Unlisted action (range as in action payload)] - > Source: any floor, Destination: any floor.  Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 201     - error message - " Ignoring call, unknown call action: {action id}"     - error message - " Ignoring call, unknown call action: UNDEFINED" if 0 |
| Test result | Result: Pass  Status: PASS  Description: action=200 or 0, returns unknown/undefined error  Duration: 20833.784103393555 ms  Response Data:  {  "phase": "action\_0\_result",  "data": "Action 0 (UNDEFINED): Option 2 - Call allowed and ignored (timestamp-only response)" }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "8fb7755b-e428-4abc-bd4b-4aaa87902c34",  "area": 1000,  "time": "2025-08-17T02:34:06.873587Z",  "terminal": 1,  "call": {  "action": 0,  "destination": 2000  }  } }  Response Data:  []  Status Code: 200  Timestamp: 2025-08-17T02:34:17.287387+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "d072de9c-c801-40b5-a4c0-9ed9d4f2c0fc",  "area": 1000,  "time": "2025-08-17T02:33:56.454303Z",  "terminal": 1,  "call": {  "action": 200,  "destination": 2000  }  } }  Request Timestamp: 1755398036.4540238  Response Timestamp: 1755398057.287808 |

|  |  |
| --- | --- |
| Test 7 | Call: Disabled action call with action id = 4 [listed enabled action (range as in action payload)] - > Source: any floor, Destination: any floor.  Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 201     - error message - " Ignoring call, disabled call action: {{actionid}}" |
| Test result | Result: Pass  Status: PASS  Description: action=4, returns disabled call action error  Duration: 3180.410861968994 ms  Response Data:  {  "phase": "call\_response",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 894275437,  "statusCode": 201,  "data": {  "request\_id": 894275437,  "success": true,  "session\_id": 285  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 285  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "c0010c60-19e0-4ed0-a74f-4b2ce90c9793",  "area": 1000,  "time": "2025-08-17T02:34:17.288065Z",  "terminal": 1,  "call": {  "action": 4,  "destination": 2000  }  } }  Response Data:  [  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 894275437,  "statusCode": 201,  "data": {  "request\_id": 894275437,  "success": true,  "session\_id": 285  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 285  } ]  Status Code: 201  Timestamp: 2025-08-17T02:34:18.466425+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "c0010c60-19e0-4ed0-a74f-4b2ce90c9793",  "area": 1000,  "time": "2025-08-17T02:34:17.288065Z",  "terminal": 1,  "call": {  "action": 4,  "destination": 2000  }  } }  Request Timestamp: 1755398057.287862  Response Timestamp: 1755398060.468273 |

|  |  |
| --- | --- |
| Test 8 | Call: Mixed action call with action id = 2002 - > Source: any floor.  Note: This is applicable for Landing Call only. Elevator at first floor and direction down. use if applicable to robot use case |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 201     - error message - "INVALID\_DIRECTION " |
| Test result | Result: Pass  Status: PASS  Description: Down call from 1F, returns INVALID\_DIRECTION error  Duration: 10424.366235733032 ms  Response Data:  {  "phase": "call\_response",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 816383181,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:34:20.765Z"  }  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "2673e9ac-dc18-449f-abe4-d5e0b700a169",  "area": 1000,  "time": "2025-08-17T02:34:20.468739Z",  "terminal": 1,  "call": {  "action": 2002  }  } }  Response Data:  [  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 816383181,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:34:20.765Z"  }  } ]  Status Code: 201  Timestamp: 2025-08-17T02:34:30.892182+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "2673e9ac-dc18-449f-abe4-d5e0b700a169",  "area": 1000,  "time": "2025-08-17T02:34:20.468739Z",  "terminal": 1,  "call": {  "action": 2002  }  } }  Request Timestamp: 1755398060.468351  Response Timestamp: 1755398070.8927171 |

|  |  |
| --- | --- |
| Test 9 | Call: Delay call with delay = 5 - > Source: any floor, Destination: any floor  Note: Landing Call – Source only, Car Call – Destination only. use if applicable to robot use case |
| Expected result | * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: delay=5, normal allocation and movement  Duration: 10425.403356552124 ms  Response Data:  {  "phase": "call\_response",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 883699663,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:34:31.190Z"  }  } }  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "6011cc69-4627-45bc-9632-d989ac359634",  "area": 1000,  "time": "2025-08-17T02:34:30.893077Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000,  "delay": 5  }  } }  Request Timestamp: 1755398070.892789  Response Timestamp: 1755398081.3181922 |

|  |  |
| --- | --- |
| Test 10 | Call: Delay call with delay = 40 [Invalid delay (range 0-30sec)] - > source: any floor, Destination: any floor  Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 201     - error message - " Invalid json payload" |
| Test result | Result: Pass  Status: PASS  Description: delay=40, returns Invalid json payload error  Duration: 0.47779083251953125 ms  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "2f39cee3-38c8-4819-976f-ec472e56696a",  "area": 1000,  "time": "2025-08-17T02:34:41.318529Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000,  "delay": 40  }  } }  Request Timestamp: 1755398081.3182652  Response Timestamp: 1755398081.318743 |

|  |  |
| --- | --- |
| Test 11 | Call: Transfer floor call - > Source: any floor, Destination: any floor  Note: The source and destination floors cannot be served by the same elevator.  Car Call – Destination only |
| Expected result | * Call accepted and elevator moving   + Response code 201   + Session id returned     - modified destination included     - modified reason included * Elevator tracking as needed   + if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: modified\_destination and modified\_reason visible  Duration: 3745.028018951416 ms  Response Data:  {  "phase": "second\_segment\_response",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 813346561,  "statusCode": 201,  "data": {  "request\_id": 813346561,  "success": true,  "session\_id": 288  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 288  } }  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "ccc6483e-cc5b-4a57-8de9-365e0ec7a249",  "area": 1000,  "time": "2025-08-17T02:34:41.318942Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 3000  }  } }  Request Timestamp: 1755398081.318781  Response Timestamp: 1755398085.063809 |

|  |  |
| --- | --- |
| Test 12 | Call: Through lift call - > Source: any floor, Destination: any floor  Note: Both source and destination floors are on the same floor but opposite side of the elevator. |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 201     - cancel Reason "SAME\_SOURCE\_AND\_DEST\_FLOOR” |
| Test result | Result: Pass  Status: PASS  Description: SAME\_SOURCE\_AND\_DEST\_FLOOR error  Duration: 10477.10108757019 ms  Response Data:  {  "phase": "same\_floor\_call",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 191089603,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:34:45.412Z"  }  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "39143cb3-525d-41bb-bb66-0d720ed6d0fb",  "area": 2000,  "time": "2025-08-17T02:34:45.064109Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Response Data:  [  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 191089603,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:34:45.412Z"  }  } ]  Status Code: 201  Timestamp: 2025-08-17T02:34:45.064135+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "39143cb3-525d-41bb-bb66-0d720ed6d0fb",  "area": 2000,  "time": "2025-08-17T02:34:45.064109Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Request Timestamp: 1755398085.0638778  Response Timestamp: 1755398095.540979 |

|  |  |
| --- | --- |
| Test 13 | Call: No travel call - > Source: any floor, Destination: same as source floor  Note: Both source and destination floors are on the same floor and same side of the elevator. |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 201     - cancel Reason "SAME\_SOURCE\_AND\_DEST\_FLOOR” |
| Test result | Result: Pass  Status: PASS  Description: Same floor same side error  Duration: 20889.889001846313 ms  Response Data:  {  "phase": "duplicate\_call",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 171845825,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:35:06.288Z"  }  } }  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "4668c9d8-6fc0-49be-8cac-5fbc7cb715f8",  "area": 2000,  "time": "2025-08-17T02:34:55.541335Z",  "terminal": 1,  "call": {  "action": 1  }  } }  Request Timestamp: 1755398095.541041  Response Timestamp: 1755398116.43093 |

|  |  |
| --- | --- |
| Test 14 | Call: Specific lift Call- > Source: any floor, Destination: any floor  Note: Allowed Lift “allowed­lifts” id to be included in the send request. Landing Call – Source only, Car Call – Destination only |
| Expected result | * Call accepted and elevator starts moving   + Response code 201   + Session id returned * Elevator tracking as needed |
| Test result | Result: Pass  Status: PASS  Description: allowed\_lifts, allocation within set  Duration: 10422.26791381836 ms  Response Data:  {  "phase": "call\_response",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 678435774,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:35:16.718Z"  }  } }  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "1e3111a9-2293-42c9-af5a-7d63636ce62c",  "area": 1000,  "time": "2025-08-17T02:35:16.431292Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000,  "allowed\_lifts": [  1  ]  }  } }  Request Timestamp: 1755398116.4310021  Response Timestamp: 1755398126.85327 |

|  |  |
| --- | --- |
| Test 15 | Call: Cancel Call- > Source: any floor, Destination: any floor  Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Request Elevator to move to any floor   + Response code 201   + Session id returned * Elevator tracking as needed * Send request cancel with payload including Session id   + Response code 201   + Elevator stop moving |
| Test result | Result: Pass  Status: PASS  Description: delete(session\_id), call\_state=canceled  Duration: 10867.506742477417 ms  Response Data:  {  "phase": "call",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 261318679,  "statusCode": 201,  "data": {  "request\_id": 261318679,  "success": true,  "session\_id": 290  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 290  } }  Request Timestamp: 1755398126.853342  Response Timestamp: 1755398137.7208488 |

|  |  |
| --- | --- |
| Test 16 | Call: Null call - > Source: any floor, Destination: any floor  Note: Destination floor is invalid and not part of building config. Car Call – Destination only |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 201     - error message - "Ignoring call, unable to resolve destination: area:\*\*\*\*” |
| Test result | Result: Pass  Status: PASS  Description: unable to resolve destination error  Duration: 10458.096981048584 ms  Response Data:  {  "phase": "call\_response",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 897289224,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:35:38.035Z"  }  } }  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "917a4c59-7341-4726-b4d0-47fdd1509825",  "area": 1000,  "time": "2025-08-17T02:35:37.721115Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 9999  }  } }  Request Timestamp: 1755398137.720905  Response Timestamp: 1755398148.179002 |

|  |  |
| --- | --- |
| Test 17 | Call: Null call - > Source: any floor, Destination: -  Note: Destination floor is not defined. Car Call – Destination only |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 201     - error message - " Ignoring call, destination not defined |
| Test result | Result: Pass  Status: PASS  Description: unable to resolve destination error  Duration: 10416.5678024292 ms  Response Data:  {  "phase": "call\_response",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 398876182,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:35:48.472Z"  }  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "722c19f2-ea7d-472e-835e-db3b4504b9eb",  "area": 1000,  "time": "2025-08-17T02:35:48.179357Z",  "terminal": 1,  "call": {  "action": 2  }  } }  Response Data:  [  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 398876182,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:35:48.472Z"  }  } ]  Status Code: 201  Timestamp: 2025-08-17T02:35:48.179395+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "722c19f2-ea7d-472e-835e-db3b4504b9eb",  "area": 1000,  "time": "2025-08-17T02:35:48.179357Z",  "terminal": 1,  "call": {  "action": 2  }  } }  Request Timestamp: 1755398148.179075  Response Timestamp: 1755398158.5956428 |

|  |  |
| --- | --- |
| Test 18 | Call: Null call - > Source: any floor, Destination: any floor  Note: Source floor is invalid and not part of building config. Landing Call – Source only |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 201     - error message - "Ignoring call, unable to resolve area: area:\*\*\*\*” |
| Test result | Result: Pass  Status: PASS  Description: Connection and event subscription test  Duration: 1592.7560329437256 ms  Response Data:  {  "phase": "events",  "data": {  "data": {  "request\_id": 131199781,  "success": false,  "error": "Ignoring call, unknown call action: 200"  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1"  } }  Request Timestamp: 1755398158.5957088  Response Timestamp: 1755398160.1884649 |

|  |  |
| --- | --- |
| Test 19 | Call: Null call - > Source: any floor, Destination: any floor  Note: Source floor and Destination floor are both invalid and not part of building config. |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 201     - error message - "Ignoring call, unable to resolve area: area:\*\*\*\*" |
| Test result | Result: Pass  Status: PASS  Description: System ping test  Duration: 1321.4521408081055 ms  Response Data:  {  "phase": "ping\_response",  "data": {  "data": {  "request\_id": 806543029,  "time": "2025-08-17T02:36:00.884Z"  },  "callType": "ping",  "buildingId": "building:L1QinntdEOg",  "groupId": "1"  } }  Request Timestamp: 1755398160.1884868  Response Timestamp: 1755398161.509939 |

|  |  |
| --- | --- |
| Test 20 | Call: Misplaced call to Building ID: a4KrX2cei - > Source: any floor, Destination: any floor  Note: Building ID is invalid and not part of used resources. Landing Call – Source only, Car Call – Destination only |
| Expected result | * Option 1   + Illegal call prevented by robot controller * Option 2   + Call allowed and Call cancelled     - Response code 404     - error message - "Building data not found for ID building:a4KrX2cei” |
| Test result | Result: Pass  Status: PASS  Description: Call to invalid building ID a4KrX2cei  Duration: 997.0018863677979 ms  Response Data:  {  "phase": "misplaced\_call\_response",  "data": {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 250613106,  "statusCode": 403,  "data": {  "error": "Token does not contain required scope for a4KrX2cei:1"  }  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:a4KrX2cei",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "4d1ee6fb-ef78-49b2-9cf2-1a601e8a799f",  "area": 1000,  "time": "2025-08-17T02:36:01.510264Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Response Data:  [  {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 250613106,  "statusCode": 403,  "data": {  "error": "Token does not contain required scope for a4KrX2cei:1"  }  } ]  Status Code: 403  Timestamp: 2025-08-17T02:36:01.510288+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:a4KrX2cei",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "4d1ee6fb-ef78-49b2-9cf2-1a601e8a799f",  "area": 1000,  "time": "2025-08-17T02:36:01.510264Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Request Timestamp: 1755398161.5100071  Response Timestamp: 1755398162.507009 |

## Multiple groups and call giving

|  |  |
| --- | --- |
| Functionality declaration 1 | Specify how *multiple groups* (at least two building ids) or (one building id with multiple suffix :1, :2, …) in a building (physical building) are selected for placing calls. |
| Comment |  |

|  |  |
| --- | --- |
| Test 21 | Group 2 (second building id) of physical building’s lobby 2 access is provided.  lobby 2 call - > Source: any floor, Destination: any floor Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Possible to select between groups (group 1 or group 2) * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: 404+Building data not found  Duration: 418.38884353637695 ms  Response Data:  {  "phase": "call\_response",  "data": {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 972805408,  "statusCode": 403,  "data": {  "error": "Token does not contain required scope for invalid123:1"  }  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:invalid123",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "3668e4b0-04d7-4d98-b9cc-c4bcb16aebef",  "area": 1000,  "time": "2025-08-17T02:36:02.507354Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Response Data:  [  {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 972805408,  "statusCode": 403,  "data": {  "error": "Token does not contain required scope for invalid123:1"  }  } ]  Status Code: 403  Timestamp: 2025-08-17T02:36:02.507382+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:invalid123",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "3668e4b0-04d7-4d98-b9cc-c4bcb16aebef",  "area": 1000,  "time": "2025-08-17T02:36:02.507354Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Request Timestamp: 1755398162.507071  Response Timestamp: 1755398162.9254599 |

|  |  |
| --- | --- |
| Test 22 | Group 2 (suffix :2) of physical building’s lobby 2 access is provided.  lobby 2 call - > Source: any floor, Destination: any floor Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Possible to select between groups (group 1 or group 2) * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: Same success flow as #4  Duration: 443.71604919433594 ms  Response Data:  {  "phase": "second\_building\_call",  "data": {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 523599284,  "statusCode": 403,  "data": {  "error": "Token does not contain required scope for demo02:2"  }  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:demo02",  "callType": "action",  "groupId": "2",  "payload": {  "request\_id": "2c518933-9b82-4407-8e17-44d8e1a1e2e0",  "area": 1000,  "time": "2025-08-17T02:36:02.925619Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Response Data:  [  {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 523599284,  "statusCode": 403,  "data": {  "error": "Token does not contain required scope for demo02:2"  }  } ]  Status Code: 403  Timestamp: 2025-08-17T02:36:02.925632+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:demo02",  "callType": "action",  "groupId": "2",  "payload": {  "request\_id": "2c518933-9b82-4407-8e17-44d8e1a1e2e0",  "area": 1000,  "time": "2025-08-17T02:36:02.925619Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Request Timestamp: 1755398162.925489  Response Timestamp: 1755398163.369205 |

## Integration with access control and call giving

|  |  |
| --- | --- |
| Functionality declaration 2 | Specify how robot’s access permission to *specific floor(s)* is obtained from access control system for calls by solution. |
| Comment |  |

|  |  |
| --- | --- |
| Test 23 | Call: Access control call - > Source: any floor, Destination: any floor  Note: floors are as defined in the access control permissions. Landing Call – Source only, Car Call – Destination only |
| Expected result | * Robot have access only to the floors specified in the access rights * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: Robot access control permissions  Duration: 1011.1100673675537 ms  Response Data:  {  "phase": "access\_control\_call",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 841101449,  "statusCode": 201,  "data": {  "request\_id": 841101449,  "success": true,  "session\_id": 291  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 291  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "0f76e407-8fb7-4c93-a866-966973383755",  "area": 1000,  "time": "2025-08-17T02:36:03.369512Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Response Data:  [  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 841101449,  "statusCode": 201,  "data": {  "request\_id": 841101449,  "success": true,  "session\_id": 291  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 291  } ]  Status Code: 201  Timestamp: 2025-08-17T02:36:03.369535+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "0f76e407-8fb7-4c93-a866-966973383755",  "area": 1000,  "time": "2025-08-17T02:36:03.369512Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Request Timestamp: 1755398163.369267  Response Timestamp: 1755398164.380377 |

|  |  |
| --- | --- |
| Test 24 | Call: Access control call with media id - > Source: any floor, Destination: any floor  Note: floors are already defined in the access control system linked with the media id (with and without company code). Landing Call – Source only, Car Call – Destination only |
| Expected result | * Optional Robot have access only to the floors specified in the access rights * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: Format error rejection  Duration: 421.2219715118408 ms  Response Data:  {  "phase": "invalid\_call",  "data": {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 914519478,  "statusCode": 400,  "data": {  "error": "'payload.area' must be a number"  }  } }  Request Parameters:  {  "type": "invalid-type",  "buildingId": "building:L1QinntdEOg",  "invalid\_field": "invalid\_value" }  Request Timestamp: 1755398164.380434  Response Timestamp: 1755398164.801656 |

## Location control and call giving

|  |  |
| --- | --- |
| Functionality declaration 3 | Specify how a robot is *geographically placed* in the building and is preventing calls from long distances, away from the building (e.g., beacon, geolocation…). |
| Comment |  |

|  |  |
| --- | --- |
| Test 25 | Call: Geographically controlled call - > Source: any floor, Destination: any floor  Note: users are only allowed to place a call within the building they are located. Landing Call – Source only, Car Call – Destination only |
| Expected result | * While out of range, solution disables all calls   + Calls cannot be made * While in range, solution enable all calls   + Call accepted and elevator moving     - Response code 201     - Session id returned   + Elevator tracking as needed     - if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: Concurrent request handling  Duration: 2671.6761589050293 ms  Response Data:  {  "phase": "concurrent\_calls",  "data": [  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 383161174,  "statusCode": 201,  "data": {  "request\_id": 383161174,  "success": true,  "session\_id": 292  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 292  },  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 857083232,  "statusCode": 201,  "data": {  "request\_id": 857083232,  "success": true,  "session\_id": 294  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 294  },  {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 700505174,  "statusCode": 201,  "data": {  "request\_id": 700505174,  "success": true,  "session\_id": 293  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 293  }  ] }  Request Timestamp: 1755398164.801705  Response Timestamp: 1755398167.473381 |

|  |  |
| --- | --- |
| Functionality declaration 4 | Specify how solution controls *barriers* (e.g., turnstiles, speed gate…).  In situations where there is no separate system (e.g., by access control system), specify how barrier will be controlled. |
| Comment |  |

|  |  |
| --- | --- |
| Functionality declaration 5 | Specify how a robot is *geographically placed next to a barrier* in the building and how it is preventing calls from being made before barrier is crossed (e.g., beacon, geolocation…). |
| Comment |  |

|  |  |
| --- | --- |
| Functionality declaration 6 | Specify how robot *driving direction is obtained* while heading towards or away from the elevator lobby (typically for automatic calls). |
| Comment |  |

|  |  |
| --- | --- |
| Test 26 | Call: Barrier call - > Source: any floor, Destination: any floor  Note: user is only allowed to place a call after crossing the barrier. Landing Call – Source only, Car Call – Destination only |
| Expected result | * Before barrier, solution disables all calls   + Calls cannot be made * After barrier, solution enables all calls   + Call accepted and elevator moving     - Response code 201     - Session id returned   + Elevator tracking as needed     - if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: Event subscription test  Duration: 2970.6060886383057 ms  Response Data:  {  "phase": "events",  "data": {  "data": {  "request\_id": 973178331,  "success": true,  "session\_id": 295  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1"  } }  Request Timestamp: 1755398167.473431  Response Timestamp: 1755398170.4440372 |

|  |  |
| --- | --- |
| Test 27 | Call: Barrier control and call - > Source: any floor, Destination: any floor  Note: user is only allowed to place a call after crossing the barrier. Landing Call – Source only, Car Call – Destination only |
| Expected result | * Before barrier, solution disables all calls   + Calls cannot be made * At barrier, solution releases barrier for passage * After barrier, solution enables all calls   + Call accepted and elevator moving     - Response code 201     - Session id returned   + Elevator tracking as needed     - if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: Load handling capability  Duration: 5280.970096588135 ms  Response Data:  {  "phase": "load\_test",  "data": {  "total\_calls": 10,  "successful\_calls": 10,  "duration": 5.2803168296813965,  "calls\_per\_second": 1.8938257537480718  } }  Request Timestamp: 1755398170.444105  Response Timestamp: 1755398175.725075 |

|  |  |
| --- | --- |
| Functionality declaration 7 | Specify control method for *avoiding multiple calls* when in range and heading in the right direction. And while in elevator lobby (while waiting for allocated elevator). |
| Comment |  |

|  |  |
| --- | --- |
| Test 28 | Call: Multiple automatic calls prevention - > Source: any floor, Destination: any floor  Note: user should only place single call for a journey. Landing Call – Source only, Car Call – Destination only |
| Expected result | * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door * No other calls generated while elevator approaches |
| Test result | Result: Pass  Status: PASS  Description: System error recovery capability  Duration: 11463.08708190918 ms  Response Data:  {  "phase": "recovery\_request",  "data": {  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 478299057,  "statusCode": 201,  "data": {  "request\_id": 478299057,  "success": true,  "session\_id": 296  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 296  } }  Request Timestamp: 1755398175.725149  Response Timestamp: 1755398187.188236 |

|  |  |
| --- | --- |
| Test 29 | Call: Multiple automatic call prevention for Group (Lobby) 2 - > Source: any floor, Destination: any floor  Note: user should only place single call for a journey to the correct building id. Landing Call – Source only, Car Call – Destination only |
| Expected result | * Possible to automatically select between groups (building id 1 or building id 2) * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door * No other calls generated while elevator approaches |
| Test result | Result: Pass  Status: PASS  Description: Input data validation  Duration: 707.8561782836914 ms  Response Data:  {  "phase": "validation\_tests",  "data": [  {  "test\_data": {  "area": "string\_instead\_of\_int",  "action": 2,  "destination": 2000,  "description": "Invalid area type"  },  "response": {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 992325656,  "statusCode": 400,  "data": {  "error": "'payload.area' must be a number"  }  },  "validated": true,  "description": "Invalid area type"  },  {  "test\_data": {  "area": 1000,  "action": 2,  "destination": "invalid",  "description": "Invalid destination type"  },  "response": {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 896725465,  "statusCode": 400,  "data": {  "error": "'payload.call.destination' must be a number"  }  },  "validated": true,  "description": "Invalid destination type"  }  ] }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "ae2abce8-9998-4d2c-a02f-c0e16893c424",  "area": "string\_instead\_of\_int",  "time": "2025-08-17T02:36:27.188598Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Response Data:  [  {  "test\_data": {  "area": "string\_instead\_of\_int",  "action": 2,  "destination": 2000,  "description": "Invalid area type"  },  "response": {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 992325656,  "statusCode": 400,  "data": {  "error": "'payload.area' must be a number"  }  },  "validated": true,  "description": "Invalid area type"  },  {  "test\_data": {  "area": 1000,  "action": 2,  "destination": "invalid",  "description": "Invalid destination type"  },  "response": {  "type": "error",  "connectionId": "PbbeHfGMjoECFwA=",  "requestId": 896725465,  "statusCode": 400,  "data": {  "error": "'payload.call.destination' must be a number"  }  },  "validated": true,  "description": "Invalid destination type"  } ]  Status Code: 200  Timestamp: 2025-08-17T02:36:27.188628+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "ae2abce8-9998-4d2c-a02f-c0e16893c424",  "area": "string\_instead\_of\_int",  "time": "2025-08-17T02:36:27.188598Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Request Timestamp: 1755398187.188318  Response Timestamp: 1755398187.8961742 |

|  |  |
| --- | --- |
| Test 30 | Call: Multiple automatic call prevention for Group (Lobby) 2 - > Source: any floor, Destination: any floor  Note: user should only place single call for a journey to the correct group suffix. Landing Call – Source only, Car Call – Destination only |
| Expected result | * Possible to automatically select between groups (group suffix 1 or group suffix 2) * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door * No other calls generated while elevator approaches |
| Test result | Result: Pass  Status: PASS  Description: Token validation test  Duration: 983.8640689849854 ms  Response Data:  {  "phase": "token\_validation",  "data": {  "data": {  "request\_id": 127114326,  "time": "2025-08-17T02:36:28.554Z"  },  "callType": "ping",  "buildingId": "building:L1QinntdEOg",  "groupId": "1"  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/ping  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "ping",  "groupId": "1",  "payload": {  "request\_id": "1c7a9137-b450-4093-9bfe-e09f9caa2ea2",  "time": "2025-08-17T02:36:27.896453Z"  } }  Response Data:  [  {  "data": {  "request\_id": 127114326,  "time": "2025-08-17T02:36:28.554Z"  },  "callType": "ping",  "buildingId": "building:L1QinntdEOg",  "groupId": "1"  } ]  Status Code: None  Timestamp: 2025-08-17T02:36:27.896474+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "ping",  "groupId": "1",  "payload": {  "request\_id": "1c7a9137-b450-4093-9bfe-e09f9caa2ea2",  "time": "2025-08-17T02:36:27.896453Z"  } }  Request Timestamp: 1755398187.896236  Response Timestamp: 1755398188.8801 |

## Elevator locks and call giving.

This test is done with test assistance. Contact KONE API Support (api-support@kone.com) to book a test assistant.

|  |  |
| --- | --- |
| Test 31 | Call: Enable locks - > Source: any floor, Destination: any floor  Note: in this use case source floor is locked by ACS solution. Landing Call – Source only, Car Call – Destination only |
| Expected result | * Call accepted and cancelled   + Response code 201   + cancel Reason - "FLOOR\_IS\_LOCKED" |
| Test result | Result: Pass  Status: PASS  Description: Rate limiting test  Duration: 8248.293161392212 ms  Response Data:  {  "phase": "rapid\_calls",  "data": [  {  "request\_num": 1,  "status": 201,  "timestamp": 0.4262359142303467  },  {  "request\_num": 2,  "status": 201,  "timestamp": 0.8303449153900146  },  {  "request\_num": 3,  "status": 201,  "timestamp": 1.2616651058197021  },  {  "request\_num": 4,  "status": 201,  "timestamp": 1.6856200695037842  },  {  "request\_num": 5,  "status": 201,  "timestamp": 2.0805811882019043  },  {  "request\_num": 6,  "status": 201,  "timestamp": 2.4912209510803223  },  {  "request\_num": 7,  "status": 201,  "timestamp": 2.910090923309326  },  {  "request\_num": 8,  "status": 201,  "timestamp": 3.3340868949890137  },  {  "request\_num": 9,  "status": 201,  "timestamp": 3.753546953201294  },  {  "request\_num": 10,  "status": 201,  "timestamp": 4.159578084945679  },  {  "request\_num": 11,  "status": 201,  "timestamp": 4.5752739906311035  },  {  "request\_num": 12,  "status": 201,  "timestamp": 4.98127007484436  },  {  "request\_num": 13,  "status": 201,  "timestamp": 5.381802082061768  },  {  "request\_num": 14,  "status": 201,  "timestamp": 5.772985219955444  },  {  "request\_num": 15,  "status": 201,  "timestamp": 6.182934045791626  },  {  "request\_num": 16,  "status": 201,  "timestamp": 6.589295148849487  },  {  "request\_num": 17,  "status": 201,  "timestamp": 7.017275810241699  },  {  "request\_num": 18,  "status": 201,  "timestamp": 7.421535968780518  },  {  "request\_num": 19,  "status": 201,  "timestamp": 7.816069841384888  },  {  "request\_num": 20,  "status": 201,  "timestamp": 8.247756004333496  }  ] }  Request Timestamp: 1755398188.880147  Response Timestamp: 1755398197.1284401 |

|  |  |
| --- | --- |
| Test 32 | Call: Disable locks - > Source: any floor, Destination: any floor  Note: in this use case source floor is unlocked by ACS solution. Landing Call – Source only, Car Call – Destination only |
| Expected result | * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: Connection reconnection test  Duration: 3409.682035446167 ms  Response Data:  {  "phase": "reconnect",  "data": {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": "1755398198846",  "statusCode": 201,  "data": {  "time": "2025-08-17T02:36:40.412Z"  }  } }  Request Timestamp: 1755398197.128509  Response Timestamp: 1755398200.538191 |

## Device disabling and call giving.

This test is done with test assistance. Contact KONE API Support (api-support@kone.com) to book a test assistant.

|  |  |
| --- | --- |
| Test 33 | Call: Elevator’s allocation interrupted (all elevators disabled) - > Source: any floor, Destination: any floor  Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Call accepted and cancelled   + Response code 201   + cancel Reason - "NO\_LIFT\_AVAILABLE "   + Indicate call failure to user (such as timeout) |
| Test result | Result: Pass  Status: PASS  Description: System status check  Duration: 2275.212049484253 ms  Response Data:  {  "phase": "ping\_check",  "data": {  "data": {  "request\_id": 430729483,  "time": "2025-08-17T02:36:42.250Z"  },  "callType": "ping",  "buildingId": "building:L1QinntdEOg",  "groupId": "1"  } }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/config  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "config",  "groupId": "1",  "payload": {  "request\_id": "9438c3af-5a7e-4ba9-ba80-13a6f9b318b0",  "time": "2025-08-17T02:36:40.538514Z"  } }  Response Data:  [  {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": "985670715",  "statusCode": 201,  "data": {  "time": "2025-08-17T02:36:40.901Z"  }  },  {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": "853121897",  "statusCode": 201,  "data": {  "time": "2025-08-17T02:36:41.420Z"  }  } ]  Status Code: 200  Timestamp: 2025-08-17T02:36:40.538539+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "config",  "groupId": "1",  "payload": {  "request\_id": "9438c3af-5a7e-4ba9-ba80-13a6f9b318b0",  "time": "2025-08-17T02:36:40.538514Z"  } }  Request Timestamp: 1755398200.538259  Response Timestamp: 1755398202.813471 |

|  |  |
| --- | --- |
| Test 34 | Call: Elevator’s allocation interrupted (all elevators enabled) - > Source: any floor, Destination: any floor  Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: Boundary value handling  Duration: 21198.906898498535 ms  Response Data:  {  "phase": "edge\_cases",  "data": [  {  "case": "empty\_building\_id",  "response": {  "type": "error",  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": 691128425,  "statusCode": 400,  "data": {  "error": "'buildingId' is not allowed to be empty"  }  },  "handled": true  },  {  "case": "large\_area",  "response": {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": 173136797,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:36:43.451Z"  }  },  "handled": true  },  {  "case": "negative\_area",  "response": {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": 840630824,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:36:53.882Z"  }  },  "handled": true  }  ] }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "decd274c-5935-4db5-94cb-7822362306d8",  "area": 1000,  "time": "2025-08-17T02:36:42.813720Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Response Data:  [  {  "case": "empty\_building\_id",  "response": {  "type": "error",  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": 691128425,  "statusCode": 400,  "data": {  "error": "'buildingId' is not allowed to be empty"  }  },  "handled": true  },  {  "case": "large\_area",  "response": {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": 173136797,  "statusCode": 201,  "data": {  "time": "2025-08-17T02:36:43.451Z"  }  },  "handled": true  } ]  Status Code: 200  Timestamp: 2025-08-17T02:36:42.813740+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "decd274c-5935-4db5-94cb-7822362306d8",  "area": 1000,  "time": "2025-08-17T02:36:42.813720Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Request Timestamp: 1755398202.813518  Response Timestamp: 1755398224.012425 |

|  |  |
| --- | --- |
| Test 35 | Call: End-to-end communication interrupted (DTU disconnected) - > Source: any floor, Destination: any floor  Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Call stuck on created   + Response code 201   + error: 1005 * Indicate call failure to user (such as timeout) |
| Test result | Result: Pass  Status: PASS  Description: Performance benchmark test  Duration: 2053.074836730957 ms  Response Data:  {  "phase": "performance\_data",  "data": [  {  "request\_num": 1,  "duration": 0.4168252944946289,  "status": 201,  "success": true  },  {  "request\_num": 2,  "duration": 0.3997020721435547,  "status": 201,  "success": true  },  {  "request\_num": 3,  "duration": 0.420698881149292,  "status": 201,  "success": true  },  {  "request\_num": 4,  "duration": 0.39817023277282715,  "status": 201,  "success": true  },  {  "request\_num": 5,  "duration": 0.4170999526977539,  "status": 201,  "success": true  }  ] }  Request Timestamp: 1755398224.01247  Response Timestamp: 1755398226.0655448 |

|  |  |
| --- | --- |
| Test 36 | Call: Call failure, communication interrupted - > Ping building or group  Note: Start a ping sequence and stop pinging after positive response is obtained |
| Expected result | * Ping failed * Communication restored   + Ping Successful |
| Test result | Result: Pass  Status: PASS  Description: End-to-end integration test  Duration: 4128.957748413086 ms  Response Data:  {  "phase": "integration\_steps",  "data": [  {  "step": "get\_config",  "response": {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": "250534799",  "statusCode": 201,  "data": {  "time": "2025-08-17T02:37:06.442Z"  }  },  "success": true  },  {  "step": "subscribe",  "response": {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": "1755398226744",  "statusCode": 201,  "data": {  "time": "2025-08-17T02:37:07.029Z"  }  },  "success": true  },  {  "step": "call\_action",  "response": {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": 704111367,  "statusCode": 201,  "data": {  "request\_id": 704111367,  "success": true,  "session\_id": 297  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 297  },  "success": true  },  {  "step": "get\_events",  "event\_received": true,  "success": true  },  {  "step": "ping",  "response": {  "data": {  "request\_id": 953512167,  "time": "2025-08-17T02:37:09.906Z"  },  "callType": "ping",  "buildingId": "building:L1QinntdEOg",  "groupId": "1"  },  "success": true  }  ] }  API Calls:  Call 1:  Interface Type: WebSocket  URL: wss://dev.kone.com/stream-v2  Method: lift-call-api-v2/action  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "5f6f1c17-be25-486e-a479-322ad8db8fd0",  "area": 1000,  "time": "2025-08-17T02:37:06.065851Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Response Data:  [  {  "step": "get\_config",  "response": {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": "250534799",  "statusCode": 201,  "data": {  "time": "2025-08-17T02:37:06.442Z"  }  },  "success": true  },  {  "step": "subscribe",  "response": {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": "1755398226744",  "statusCode": 201,  "data": {  "time": "2025-08-17T02:37:07.029Z"  }  },  "success": true  } ]  Status Code: 200  Timestamp: 2025-08-17T02:37:06.065873+00:00  Request Parameters:  {  "type": "lift-call-api-v2",  "buildingId": "building:L1QinntdEOg",  "callType": "action",  "groupId": "1",  "payload": {  "request\_id": "5f6f1c17-be25-486e-a479-322ad8db8fd0",  "area": 1000,  "time": "2025-08-17T02:37:06.065851Z",  "terminal": 1,  "call": {  "action": 2,  "destination": 2000  }  } }  Request Timestamp: 1755398226.0656111  Response Timestamp: 1755398230.1945689 |

|  |  |
| --- | --- |
| Test 37 | Call: End-to-end communication enabled (DTU connected) - > Source: any floor, Destination: any floor  Note: Landing Call – Source only, Car Call – Destination only |
| Expected result | * Call accepted and elevator moving   + Response code 201   + Session id returned * Elevator tracking as needed   + if applicable hold open elevator door |
| Test result | Result: Pass  Status: PASS  Description: DTU connected communication test  Duration: 890.1529312133789 ms  Response Data:  {  "phase": "tracking\_error",  "error": "'KoneDriverV2' object has no attribute 'get\_call\_status'" }  Request Timestamp: 1755398230.194603  Response Timestamp: 1755398231.084756 |

|  |  |
| --- | --- |
| Test 38 | Custom case |
| Expected result |  |
| Test result | Result: Pass  Status: PASS  Description: Custom case test  Duration: 851.0928153991699 ms  Response Data:  {  "phase": "custom\_call\_response",  "data": {  "connectionId": "Pbb4zdAHDoECGwg=",  "requestId": 887075556,  "statusCode": 201,  "data": {  "request\_id": 887075556,  "success": true,  "session\_id": 299  },  "callType": "action",  "buildingId": "building:L1QinntdEOg",  "groupId": "1",  "sessionId": 299  } }  Request Timestamp: 1755398231.0848262  Response Timestamp: 1755398231.935919 |

## Other system functionality checks

|  |  |
| --- | --- |
| Functionality declaration 8 | Specify how *logs* for access permission and call giving are handled. |
| Comment |  |

|  |  |
| --- | --- |
| Functionality declaration 9 | Self-assessment *cybersecurity* form filled |
| Comment |  |

|  |  |
| --- | --- |
| Functionality declaration 10 | Specify how *connectivity* is handled inside and outside the elevator car. |
| Comment |  |

Legend:

|  |  |
| --- | --- |
|  | Ok /Passed |
|  | pending implementation |
|  | not tested / not passed |

# Solution under test details

## Solution description: -

## Use case:-

## Sequence diagram:-

# Appendix

## Grading (KONE use only)

Grading provides support in evaluating the functionality readiness

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Grade | | | | | |
| Functionality | 5 | 4 | 3 | 2 | 1 | 0 |
| Log | x | x | x | x | x |  |
| Elevator mode check | x | x | x | x | x |  |
| Barrier control | (x) | (x) | (x) | (x) |  |  |
| Direction detection (for automatic call) | x\* | x\* | x\* | x\* |  |  |
| Multiple group support | x | x | x |  |  |  |
| Cyber security | x | x | x |  |  |  |
| Geolocation | x | x |  |  |  |  |
| Barrier side location | x | x |  |  |  |  |
| Direct call destination (home floor) | x |  |  |  |  |  |
| Access Control Integration | x |  |  |  |  |  |

(x) – optional

x\* – if applicable

## Attribute use case

This is to explain possible scenarios covered by each test case and functional declaration.

|  |  |  |
| --- | --- | --- |
| Functional Declaration | Test case | Attributed use case |
|  | 1 | Basic connection setup needed for all use cases. |
|  | 2 | Robot can check if elevator is not operational before making a call for example if elevator is undergoing maintenance or used by a VIP at which time robot calls will rejected and robot is not allowed to use the elevator until set to normal operation. |
|  | 3 | reset injected error above and test again that all is fine. |
|  | 4 | General use case for robot to be able to call the elevator from one floor to another. |
|  | 5 | This is to check that robot is able to keep the elevator door open for extra amount of time needed for it to enter or exit the elevator. |
|  | 6 | This checks that solution can handle errors generated by using action id that is not supported by the elevator. There is a list of actions returned in the action command but if other outside this list is used, users should be instructed to use the correct value |
|  | 7 | This is to handle situations that previously accepted action is now disabled, and users need to be informed properly to switch their action selection. |
|  | 8 | There comes with the action file description of actions, if used incorrectly, users should be informed and directed. For example, if making a landing call at the last floor and selecting an action indication going down, as this is not physically possible, such error will occur. |
|  | 9 | To get maximum door open time so that robot can enter lift, use of delay is important. For example, normal elevator door is opened for 5 seconds but robot needs more time to enter the elevator, with delay data, robot can get more time to enter the elevator. |
|  | 10 | The delay is maximum 30 sec, for example if robot want to set 25 second and then set 40sec for the delay, this will cause call to be rejected. |
|  | 11 | This is for building setup where two elevator rides are needed to get to the destination. For example if robot is on floor 5 and wants to go to floor 15, in the building elevator A can serve from floor 1 to 10 and elevator B can serve from floor 10 to 20 (in a 20 floors building), and the robot makes a call with source 5 and destination 15. First elevator A will come to floor 5 and take the robot to floor 10. In this case, the robot needs to know that it is at floor 10 and not 15 and also that it needs to make another call with source 10 and destination 15 in order to use elevator B to get to its final destination. Else robot will be on floor 10 with to understanding what to do next. |
|  | 12 | For buildings with elevators that have two doors (front and rear), if a call is made from front side to rear side of the same floor for example floor 5 front and floor 5 rear for both front and rear door to open and robot can drive through the elevator, this is not allowed and will result in call rejection. |
|  | 13 | If the solution user makes a call to the same floor where it is located or for example it is on floor 5 and wants to go to floor 10 but set that it is on floor 10 and want to go to floor 10, this will result in call rejection. |
|  | 14 | In a situation that there is more than one elevator in the elevator lobby, it is possible to specify which of the elevator to serve the call being made. This use case provides the chance to selector among the elevator with the correct parameter. |
|  | 15 | This helps to cancel a call made that is no longer needed |
|  | 16 | For a building with 20 floors, if the destination is set to 21 or above, for example robot is on floor 5 and want to go to floor 11 but set destination to floor 21 instead, this will result in error and robot needs to know of the error and make correct call. |
|  | 17 | For a building with 20 floors, if the source is set to 21 or above, for example robot is on floor 5 and set that it is on floor 25, this will result in error and robot needs to know of the error and make correct call. |
|  | 18 | If the robot makes a call to the same floor where it is located or for example it is on floor 5 and wants to go to floor 10 but set that it is on floor 10 and want to go to floor 10, this will result in call rejection. |
|  | 19 | For a building with 20 floors, if the source is set to 21 and destination is set to 25 or above, for example solution user is on floor 5 and want to go to floor 11 but set the source to 21 and destination to floor 25 instead, this will result in error and solution user needs to know of the error and make correct call. |
|  | 20 | When there is mistake in building id where the robot is located for example if building id is abcdefg and robot makes a call to building cdefgab then call will be rejected. |
| 1  Explain how it is done. |  | In a physical building, there can be multiple groups for example in a building lobby to the left can be elevator West group and to the right elevator East group, when the robot is in the lobby and riding towards the West group elevator, it has to know about both groups and about the current group it is located so as to make call to the West group and not to the East group. |
|  | 21 | Like above, this is to check making call to second group, for example while at the lobby and heading towards the East group elevator, robot should not make call to West group elevator. |
| 2  Explain how it is done. |  | This applies when building have access control system. For example, if access control system in a building specify where the robot can go and it specifies that the robot can only got to floors 5 – 10. How will the robot know about this limited access? The robot should connect to access control system and get this specific floor so that it knows that it can only go to floors 5 – 10. |
|  | 22 | Like above, robot connects to access control system and now knows it can only make call to floors 5 – 10. Calls to other floors by the robot must not be allowed. As the api doesn’t prevent robot from making calls to other floors, the restriction must be done within the robot to follow the access control rules in the building. |
| 3  Explain how it is done. |  | This checks if robot is in the right building, for example if the robot is configured to work in Building 1 in a building complex and then moved to Building 2 to start operation, the robot must not be allowed to make calls to Building 1 and must identify itself to be in Building 2. |
|  | 23 | Like above this checks for example when robot is in Building 1 calls can’t be made and when the robot is moved to Building 2 it is then able to make calls. |
| 4  Explain how it is done. |  | This handles for example when there is a turnstile in the building but no means to open and close the turnstile gate. If robot have system to open and close the gate when it goes through then specify it here. Else customer needs to find another system to control the gate. |
| 5  Explain how it is done. |  | If there is a building turnstile regardless how it is controlled, how can robot for example when before the turnstile it is prevented from making any call and when after the turnstile its calling ability is enabled. If robot is going to floor 5, while before the turnstile and tries to make a call, the call making is disabled and after going pasted the turnstile the call making is enabled and it can call floor 5. |
| 6  Explain how it is done. |  | If robot makes a call at regular interval, for example when entering or exiting an elevator lobby, how to determine based on robot driving direction that call should be made and if robot is leaving the elevator lobby, calling is prevented. If robot is going from floor 5 to floor 10, when driving in the direction to lobby of floor 5, call can be made but when driving away from lobby on floor 10 then call must be disabled. |
|  | 24 | Like above, this checks for example if calls are prevented before a turnstile while the robot is driving toward the elevator lobby and once the robot crosses the turnstile, elevator calling becomes active. It can be assumed that some other system like access control system will open the turnstile for the robot, so before the turnstile is opened and when robot is on floor 5 going to floor 10, call cant be made but after access control opens the turnstile and robot crosses, then call to floor 5 is possible. |
|  | 25 | Like above, this checks for example if calls are prevented before a turnstile while the robot is driving toward the elevator lobby, the robot is able to open the turnstile and once the robot crosses the turnstile, elevator calling becomes active. In this case, it can be assumed that robot / robot system will open the turnstile for the robot, so before the turnstile is opened and when robot is on floor 5 going to floor 10, call can’t be made but after robot / robot system opens the turnstile and robot crosses, then call to floor 5 is possible. |
| 7  Explain how it is done. |  | This handles unwanted calls for example a call is already made and robot is waiting for the elevator in the lobby, while waiting, other calls should not be made. Meaning if robot is going from floor 5 to 10, and first call is made, while robot is waiting for elevator to come to floor 5, it should not make any more calls for floor 5 to 10. |
|  | 26 | Like above, this checks for example that calls are not made from floor 5 to floor 10 again while robot is still waiting for elevator A to come to floor 5. |
|  | 27 | Like above, this checks for example if calls are prevented before a turnstile while the solution user is walking toward the elevator lobby, the solution user can open the turnstile and once the solution user crosses the turnstile, elevator calling becomes active. In this case, it can be assumed that solution user / solution user system will open the turnstile for the solution user, so before the turnstile is opened and when solution user is on floor 5 going to floor 10, call can’t be made but after solution user / solution user system opens the turnstile and solution user crosses, then call to floor 5 is possible. |
|  | 28 | This is like test 10. While test 10 accepts the possibility that from the robot, calls from source to destination can be manually selected, this checks for automatic calls and that automatic call is made for the right elevator group and no further calls made afterwards. for example while at the lobby and heading towards the East group elevator, robot should not make call to West group elevator automatically. |
|  | 29 | Like above, this checks for example that calls are not made from floor 5 to floor 10 again after solution user / solution user system opens the turnstile for passage while solution user is still waiting for elevator A to come to floor 5 |
|  | 30 | This checks for local lock error handling. for example if there is local lock on floor 5 and robot make a call setting it source to 5, it will reject the call. Robot should be able to handle the rejection. |
|  | 31 | This checks for local lock error handling. for example, if there is local lock on floor 5 and solution user make a call setting it source to 5, it will reject the call. Solution user should be able to handle the rejection. |
|  | 32 | reset injected error above and test again that all is fine |
|  | 33 | This checks how un-servable call is handled for example if calls is made from floor 5 to floor 10 but the call response is that no elevator can serve therefore call cancelled, how to handle such cancellation and make another call. |
|  | 34 | reset injected error above and test again that all is fine |
|  | 35 | This checks how un-servable call is handled for example if calls is made from floor 5 to floor 10 but the call status never changes, who to handle such freeze in communication. |
|  | 36 | When a call is placed and the call fails, it is possible to ping the elevator to determine if the communication is alive before proceeding to making more calls. This pinging can stop once successful and call making resumes. |
|  | 37 | reset injected error above and test again that all is fine. |
| 8 |  | Giving explanation how logs especially access call logs to different floors are stored. |
| 9 |  | This is to indicate if the cyber security self-test has been taken. |
| 10 |  | This is to indicate how communication between the robot and either an intermediate controller at site or own cloud backend is established for example with 4G or wifi when the robot is outside and inside the elevator car. As it is known that connectivity strengths drops when robot is inside the elevator car, how is this managed to ensure continuous control of the robot. |