**Remote Control SDK user manual**

Amy Robotics

Table of content

[1. Summary 3](#_Toc1758187)

[2. Environment requirements 3](#_Toc1758188)

[2.1. Install develop environment 3](#_Toc1758191)

[2.2. Import libraries into project 3](#_Toc1758194)

[2.3. Start remote control server on robot 4](#_Toc1758195)

[3. Remote control class 4](#_Toc1758196)

[3.1. Create remote control management object 4](#_Toc1758198)

[3.2. Initialization 4](#_Toc1758199)

[3.3. Setup map parameters 5](#_Toc1758200)

[3.4. Register 5](#_Toc1758201)

[3.5. Connection listening event 5](#_Toc1758202)

[3.6. Log setup 6](#_Toc1758203)

[3.7. Send action 7](#_Toc1758204)

[3.8. Listen to robot event 8](#_Toc1758205)

[3.9. Service start or stop 9](#_Toc1758206)

[4. Navigation 9](#_Toc1758207)

[4.1. bring up navigation 9](#_Toc1758209)

[4.2. Stop navigation 10](#_Toc1758210)

[4.3. Get navigation status 11](#_Toc1758211)

[4.4. Navigate to point 11](#_Toc1758212)

[4.5. Navigation status 12](#_Toc1758213)

[4.6. Cancel navigation 12](#_Toc1758214)

[4.7. Set default navigation map ID 12](#_Toc1758215)

[5. Map management 13](#_Toc1758216)

[5.1. Acquire map list 13](#_Toc1758218)

[5.2. Delete map 14](#_Toc1758219)

[5.3. Edit map info 14](#_Toc1758220)

[5.4. Acquire map data 15](#_Toc1758221)

[5.5. Acquire mark list 15](#_Toc1758222)

[5.6. Save mark list 16](#_Toc1758223)

[6. Task management 16](#_Toc1758224)

[6.1. Acquirement task list 16](#_Toc1758226)

[6.2. Create or edit task 18](#_Toc1758227)

[6.3. Delete task 20](#_Toc1758228)

[6.4. Start / stop task 21](#_Toc1758229)

[6.5. Stop task 21](#_Toc1758230)

[6.6. Pause task 22](#_Toc1758231)

[6.7. Execute next step 22](#_Toc1758232)

[6.8. Execute last step 22](#_Toc1758233)

[6.9. Set default task 23](#_Toc1758234)

[6.10. Task test switch 23](#_Toc1758235)

[7. Robot control command 24](#_Toc1758236)

[7.1. Go forward 24](#_Toc1758238)

[7.2. Go backward 24](#_Toc1758239)

[7.3. Turn left 24](#_Toc1758240)

[7.4. Turn right 25](#_Toc1758241)

[7.5. Stop walking 25](#_Toc1758242)

[7.6. Stop all 25](#_Toc1758243)

[7.7. Dancing 26](#_Toc1758244)

[7.8. Stop dancing 26](#_Toc1758245)

[7.9. Move 26](#_Toc1758246)

[7.10. Turn head right 27](#_Toc1758247)

[7.11. Turn head left 27](#_Toc1758248)

[7.12. Turn head up 28](#_Toc1758249)

[7.13. Turn head down 28](#_Toc1758250)

[7.14. Head reset 28](#_Toc1758251)

[7.15. Nod 29](#_Toc1758252)

[7.16. Shake 29](#_Toc1758253)

[7.17. Set LED ring 29](#_Toc1758254)

[8. System command 30](#_Toc1758255)

[8.1. Acquire media volume 30](#_Toc1758257)

[8.2. Set media volume 30](#_Toc1758258)

[8.3. Volume up 31](#_Toc1758259)

[8.4. Volume down 31](#_Toc1758260)

[8.5. Talk / stop talking 32](#_Toc1758261)

[8.6. Sing 32](#_Toc1758262)

[8.7. Stop singing 32](#_Toc1758263)

[8.8. Music-play music 33](#_Toc1758264)

[8.9. Music-pause music 33](#_Toc1758265)

[8.10. Music-resume 34](#_Toc1758266)

[8.11. Music-stop 34](#_Toc1758267)

[8.12. Video-play 34](#_Toc1758268)

[8.13. Video-pause 35](#_Toc1758269)

[8.14. Video-resume 35](#_Toc1758270)

[8.15. Video-stop 36](#_Toc1758271)

[8.16. Show face 36](#_Toc1758272)

[8.17. Hide face 36](#_Toc1758273)

[8.18. Slide show-start 37](#_Toc1758274)

[8.19. Slide show-stop 37](#_Toc1758275)

[9. Error code description 38](#_Toc1758276)

Version history：

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Comment | Date | PIC |
| V1.0 | Initial commit | 2019-02-19 | mudeyu |
| V1.1 | Correct Names & params | 2019-02-22 | mudeyu  zhongjianhui |
| V1.2 | English translation & minor corrections | 2019-02-22 | zyu |

## Summary

Remote Control SDK documentation for all AmyRobotics robots.

## Environment requirements



### Install develop environment

Install JDK with version number >= 1.7



### Import libraries into project

Import library “AmyRobotRemoteClientLib.jar” into project.

Import dependent libraries:

<dependency>

<groupId>com.googlecode.protobuf-java-format</groupId>

<artifactId>protobuf-java-format</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>com.google.protobuf</groupId>

<artifactId>protobuf-java</artifactId>

<version>3.0.0</version>

</dependency>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>fastjson</artifactId>

<version>1.2.18</version>

</dependency>

<dependency>

<groupId>io.netty</groupId>

<artifactId>netty-all</artifactId>

<version>4.1.27.Final</version>

</dependency>

### Start remote control server on robot

Start remote control service on robot and make sure version number is >= 1.21.1.

### Remote control over LAN

Use sendAction method to send commands after successful connection and registration 

### Remote control over cloud

**dstcid** set to null, use sendRegister to send registration after successful connection, use sendAction method to send commands



## Remote control class



### Create remote control management object

RobotClientMgr mRobotClientMgr = RobotClientMgr.getInstance();

RobotClientMgr provides methods for server and robot communication. See following sections.

### Initialization

**Name:** setClientType(String clientType)

**Description: setup client type, incorrect type will not get through**

**Parameters：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| CLIENT\_TYPE\_WEB | String | Web client |
| CLIENT\_TYPE\_MOBILE | String | Mobile client |

**Example:**

mRobotClientMgr.setClientType(RemoteClientType.CLIENT\_TYPE\_WEB);

**Name:** init (serverIP, serverPort, userName, cid, dstCid, passwd)

**Description: initialize server IP addres,s, port etc.**

**Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| serverIP | String | Y | Server IP |
| serverPort | String | Y | Server port |
| userName | String | N | User name |
| cid | String | Y | Sender id |
| dstCid | String | Y | Robot id |
| passwd | String | Y | password |

### Setup map parameters

**Name:** setUseMapDir (String mapRootDir);

**Description: set map save path**

**Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| mapRootDir | String | Y | Map save path. Can be relative or absolute. |

**Example:**

String mapRootDir= "amyrobot//map/";

mRobotClientMgr.setUseMapDir(mapRootDir);

**Name:** String getUseMapDir(String mapRootDir);

**Description：get map saving path**

**Return：return map saving path**

### Register

**Name：**void sendRegister(String cid, String pwd);

**Description：**Send registration

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| cid | String | Y | Robot id |
| pwd | String | Y | password |

**Return：**

Will call onRegisterResult() from RobotEventListener class

### Connection listening event

**Name：**

addDataClientListener (String obj, RobotEventListener listener);

removeDataClientListener(String obj);

**Description：**

addDataClientListener adds event listener

removeDataClientListener removes event listener

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| obj | String | Y | String object, used to differentiate multiple different listeners |
| listener | RobotEventListener | Y | Event callback, see RobotEventListener for detail |

**RobotEventListener interface description：**

|  |  |
| --- | --- |
| **Name** | **Description** |
| onConnected | Callback for successful connection |
| onRegisterResult | Callback for registration results  **srcCId** registration cid  **code** error code；200 means success, error for otherwise  **errInfo** return error info in case of error |
| onDisconnected | Callback for disconnection |
| onError | Callback for error |

**Example：**

String DATA\_HANDLER\_CALLBACK\_ID = "10000";

mRobotClientMgr.addDataClientListener(DATA\_HANDLER\_CALLBACK\_ID, new BaseDataClientListener() {

@Override

public void onConnected() {

LogUtils.d(TAG, "onConnected");

}

@Override

public void onRegisterResult(String srcCId, int code, String errInfo) {

LogUtils.d(TAG, "onRegisterResult: " + srcCId + ", " + code + ", " + errInfo);

handleRegisterResult(srcCId, code, errInfo);

}

@Override

public void onDisconnected() {

LogUtils.d(TAG, "onDisconnected");

}

@Override

public void onError(Throwable e) {

LogUtils.e(TAG, "onError", e);

}

});

### Log setup

**Name：**setDebug (boolean enable)

**Description：enable / disable debug printing to console**

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| enable | boolean | Y | Enable debug logging |

**Name：**setLogLevel (int level)

**Description：set logger level**

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| level | int | Y | LOG\_LEVEL\_VERBOSE everything  LOG\_LEVEL\_DEBUG debug  LOG\_LEVEL\_INFO info  LOG\_LEVEL\_WARN warning  LOG\_LEVEL\_ERROR error |

### Send action

**Name：**

boolean sendAction(String action, Map<String, String> params, ActionEventCallback requestCallback);

boolean sendAction(String dstId, String action, Map<String, String> params, ActionEventCallback requestCallback);

boolean sendAction(String srcId, String dstId, String action, Map<String, String> params, ActionEventCallback requestCallback);

**Action description：send action**

**Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| srcId | String | Y | Sender ID |
| dstCid | String | Y | Receiver ID |
| action | String | Y | different actions entitle different names |
| params | Map<String, String> | N | Send parameters according to correspondent action, use **null** if not needed |
| requestCallback | ActionEventCallback | N | Callback success: onSuccess(RobotEvent robotEvent);  Use **resultCode** for correspondent action, see error code description;  robotEvent.resultCode == CODE\_OK  Callback on failure: onFailed(int code, String msg, Throwable e); |

**Return：return results according to correspondent actions**

**Example：**

String srcId = "200100";

String dstCid = "200101";

String action = "robot.startNavigation"; //bring up nav

Map<String, String> params = new HashMap<>();

params.put("mapId", "069a6c1d-e0fb-4d5b-a8fa-5b4c9a0b2cf5");

mRobotClientMgr.sendAction(srcId, dstCid, action, params, new ActionEventCallback() {

@Override

public void onSuccess(RobotEvent robotEvent) {

if(robotEvent.resultCode == RobotNotifyCode.CODE\_OK) {

LogUtils.e(TAG, "ok " + robotEvent.resultCode + ", " + robotEvent.notifyInfo);

} else {

LogUtils.e(TAG, "error " + robotEvent.resultCode + ", " + robotEvent.notifyInfo);

}

}

@Override

public void onFailed(int code, String msg, Throwable e) {

LogUtils.e(TAG, "error " + code + ", " + msg + ", " + e);

}

});

### Listen to robot event

**Name：**setRobotEventListener(RobotEventListener listener)

**Description: setup robot event listener**

**Description:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| listener | RobotEventListener | Y | RobotEvent see following |

RobotEvent Description:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| cid | String | Y | Robot id |
| notifyAction | String | Y | Action name |
| resultCode | int | Y | Return error code, 200 means success, otherwise means failure； See error code description for detail. |
| notifyInfo | String | N | Correspondent error info for actions. |
| notifyParams | String | N | Return |
| notifyPktId | String | Y | Event notification ID |
| data | Object | N | Extra data for extension. |

**Example：**

mRobotClientMgr.setRobotEventListener(new RobotEventListener() {

@Override

public void onRobotEvent(RobotEvent robotEvent) {

handleRobotEvent(robotEvent);

}

});

### Service start or stop

**Name：**start();

**Description: start service, automatically connect to server according to server configuration and listen to events**

**Name：**stop();

**Description：stop service**

**Name：**boolean isConnected ();

**Description: check server connection**

## Navigation



### bring up navigation

**Name:** robot.startNavigation

**Action description: bring up navigation. Only after bringing up the navigation can nav task be executed**

**Description：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** | **Value** |
| mapId | String | Y | Map ID |  |
| useDefault | String | N | Use default map when set to 1 |  |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | Robot ID |
| notifyAction | String | robot.startNavigation |
| resultCode | int | Error code, 200 means success, otherwise see error code description for more information.  Common error:  CODE\_ERROR\_STOP\_SWITCH\_IS\_OPEN, bring up nav failed, E-stop has been pressed  CODE\_ERROR, useMap failed, use map failed, correspondent map file doesn’t exist or map file format error. See notifyInfo |
| notifyInfo | String | Error info |
| notifyParams | String | Bring up navigation success |

**Note:**

Make sure E-stop hasn’t been pressed when bringing up navigation and robot should be moved to pre-defined starting location

### Stop navigation

**Name:** robot. stopNavigation

**Action description：stop navigation**

**Description:** N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | Robot id |
| notifyAction | String | robot. stopNavigation |
| resultCode | int | Error code, 200 means success, otherwise see error code description |
| notifyInfo | String | Error info |
| notifyParams | String |  |

**Note:**

Robot should be put to starting location after navigation restarts

### Get navigation status

**Name:** robot. getNavState

**Action description:** get navigation status

**Description:** N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | Robot ID |
| notifyAction | String | robot. getNavState |
| resultCode | int | Error code, 200 means success, otherwise see error code description |
| notifyInfo | String | Error info |
| notifyParams | String | See navigation status |

Navigation status:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | | **Value** | | **Type** | | **Comment** |
| NAVI\_START | #NAV02 | | String | | Nav bring up | |
| NAVI\_STOP | #NAV07 | | String | | Nav stop | |

**Note:**

Robot should be put to starting location after navigation restarts

### Navigate to point

**Name:** robot.navToPoint

**Action description:** navigate to marked point

**Description:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| x | String | Y | X coordinate  Float converted to String; |
| y | String | Y | y coordinate  Float converted to String; |
| z | String | Y | Z coordinate  Float converted to String; |
| name | String | Y | Point name for logging |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | Robot ID |
| notifyAction | String | robot.navToPoint |
| resultCode | int | Error code, 200 means success, otherwise see error code description |
| notifyInfo | String | Error info |
| notifyParams | String | See navigation status |

### Navigation status

Navigation status is defined in IPCResponse class:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Value** | **Type** | **Comment** |
| NAVI\_ARRIVE | #NAV01 | String | Arrival |
| NAVI\_START | #NAV02 | String | Bring up |
| NAVI\_LOST | #NAV03 | String | Lost |
| NAVI\_GIVEUP | #NAV04 | String | Aborted |
| NAVI\_TIMEOUT | #NAV05 | String | Timeout |
| NAVI\_CANCLE\_SUCCESS | #NAV06 | String | Cancel |
| NAVI\_STOP | #NAV07 | String | Stop |

### Cancel navigation

**Name:** robot.cancelGoal

**Action description：cancel navigation to point**

**Description: N/A**

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot.cancelGoal |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Set default navigation map ID

**Name：**robot.setDefaultNavMap

**Action description：set default navigation map ID**

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| mapId | String | Y | map ID |

**Return:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot.setDefaultNavMap |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String | N/A |

## Map management



### Acquire map list

**Name：**robot. getMapList

**Action description：acquire map list**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. getMapList |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String | Map info list, json string, List<MarkPointEntity> |

MapListEntity：

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| mapFileInfoList | List<MapFileInfo> | Map info list |
| defaultNavMapId | String | Default map, null means no default map exists |

MapFileInfo：

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| id | String | Map id |
| name | String | Name |
| desc | String | Description |
| createTime | String | Date |
| width | Int | Width |
| height | Int | Height |
| ratio | Double | Resolution, pixel to meter ratio |
| x | Double | Initial location X |
| y | Double | Initial location Y |
| z | Double | Initial location Z |

### Delete map

**Name：**robot.deleteMap

**Action description：delete map**

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| mapId | String | Y | map ID |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot.deleteMap |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String | N/A |

### Edit map info

**Name：**robot.modifyMap

**Action description：edit map info, currently support “name”**

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| mapId | String | Y | map ID |
| newMapName | String | Y | Name after edit |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot.modifyMap |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Acquire map data

**Name：**robot.getMapData

**Action description：acquire map data, will save map to path after acquisition command sent** setUseMapDir() set map saving path, e.g.

String mapRootDir= "amyrobot/map/";

mRobotClientMgr.setUseMapDir(mapRootDir);

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| mapId | String | Y | map ID |
| mapLastModifiedTime | long | Y | Last modification time |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot.getMapData.Complete Map received |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String | map ID |

### Acquire mark list

**Name：**robot.getMarkPointList

**Action description：acquire mark list**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot.getMarkPointList |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String | json, List<MarkPointEntity> |

MarkPointEntity：

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| text | String | Mark name for overlap check |
| desc | String | Mark description |
| name2 | String | Alias |
| x | float | Map coordinate X |
| y | float | Map coordinate Y |
| realX | float | Actual coordinatex |
| realY | float | Actual coordinateY |
| realAngle | float | Actual posture |
| isStartpoint | boolean | Check if it is starting point; starting point can’t be deleted |
| angle | float | Posture in degree |
| radian | float | Posture in rad |

### Save mark list

**Name：**robot.saveMarkPoint

**Action description：save mark list**

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| mapId | String | Y | map ID |
| markPointList | String | Y | json，List<MarkPointEntity> see MarkPointEntity |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. saveMarkPoint |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

## Task management



### Acquirement task list

**Name：**robot.getTaskList

**Action description：acquire task list**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot.roamTaskList |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String | Task entity list see RoamTaskListEntity |

RoamTaskListEntity：

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| taskEntityList | List<RoamTaskEntity> | Task entity object list |
| isTaskRun | boolean | If task is running |
| runTaskId | String | Running task ID |
| defaultTaskId | String | Default task ID |

RoamTaskEntity：

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| taskId | String | Task id |
| name | String | Task name |
| mapId | String | Task correspondent map ID |
| mapName | String | Map name |
| roamPointEntityList | List<RoamPointEntity> | Task point list |
| cycleTime | Int | Repetition time, -1 means infinite loop |
| arriveDo | boolean | Whether action needs to be taken on arrival |
| params | Map<String, String> | Add extension params |

RoamPointEntity

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| pointName | String | Task point name |
| action | String | Action on task point arrival; Null or empty indicates default (broadcast); For others see TaskActions |
| strVal | String | String param |
| Params | Map<String, String> | Extended param |
| paramList | Int | List parma |
| actionPreDelayMs | Int | Pre-Action delay in ms |
| actionDelayMs | boolean | Post-Action delay in ms |
| childAction | List<TaskActionEntity> | Child action: parent action execution will start child action; when parent action finishes, stop child action |

Action TaskActionEntity：

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| action | String | Action name |
| strVal | String | Action param |
| params | Map<String, String> | Extended param |
| paramList | List<String> | List param |

TaskActions

public class TaskActions {

//No action

public static final String TASK\_ACTION\_NONE = "none";

//Broadcast

public static final String TASK\_ACTION\_VOICE = "voice";

//Play audio

public static final String TASK\_ACTION\_AUDIO = "audio";

//Play video

public static final String TASK\_ACTION\_VIDEO = "video";

//Voice recognition

public static final String TASK\_ACTION\_SPEECH\_RECOGNITION = "speech.recognition";

//Recharge

public static final String TASK\_ACTION\_BACK\_CHARGE = "robot.backCharge";

//Slide show

public static final String TASK\_ACTION\_SHOW\_PICS = "showPics";

//Task confirmation

public static final String TASK\_ACTION\_TASK\_CONFIRM = "taskConfirm";

//Extended command

public static final String TASK\_ACTION\_EXT\_CMD = "ext.cmd";

//Stop all current task and start a new one

public static final String TASK\_ACTION\_START\_TASK = "robot.startTask";

}

### Create or edit task

**Name：**robot.taskSave

**Action description：save task**

**Description：N/A**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| mode | String | Y | Create; or modify |
| taskEntity | String | Y | TaskEntity |

RoamTaskEntity class：

public class RoamTaskEntity implements Serializable {

public static final int TASK\_CYCLE\_INFINITE = -1; //infinite loop

public String taskId; //Task id

public String name; //Task name

public String mapId; //Task map Id

public String mapName; //Map name

public List<RoamPointEntity> roamPointEntityList; //Task point list

public int cycleTimes; //Nb of repetition, -1 means infinite loop

public boolean arriveDo; //condition for action on arrival; if false, action will be executed even though navigation is interrupted.

public Map<String, String> params; //add extended command params

}

public class RoamPointEntity implements Serializable {

public String pointName; //task point name

public String action; //action to be executed, null or empty means default action

public String strVal; //string paramter

public Map<String, String> params; //extended param

public List<String> paramList; //list param

public int actionPreDelayMs; //pre-action execution delay in ms

public int actionDelayMs; //post-action execution delay in ms

public List<TaskActionEntity> childActions; //child action, parent action execution will start child action and stops child action when it finishes.

}

public class TaskActionEntity implements Serializable {

public String action; //action name

public String strVal; //action param

public Map<String, String> params; //extended param

public List<String> paramList; //list param

}

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. taskSave |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Delete task

**Name：**robot. taskDelete

**Action description：**delete

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| taskId | String | Y | Task Id |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. taskDelete |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Start / stop task

**Name：**robot. taskStart

**Action description：start task; will recover task if paused; if task is already be executed, it will return directly; current task should be stopped before new task execution.**

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| taskId | String | 是 | Task Id |
| useDefault | String | 否 | Use default task  1 will start default task if task id is empty  0 will not start default task |
| index | int | 是 | Start from which step; default is 0; |
| isJumpStep | Int | 是 | Jump to other step  If isJumpStep=0, whatever the index is, task will be recovered;  If isJumpStep=1, task will start again from index |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. taskStart |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Stop task

**Name：**robot. taskStop

**Action description：stop task**

**Description：N/A**

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. taskStop |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Pause task

**Name：**robot. taskPause

**Action description：pause task**

**Description：N/A**

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. taskPause |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Execute next step

**Name：**robot. taskNextStep

**Action description：execute next step**

**Description：N/A**

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. taskNextStep |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Execute last step

**Name：**robot. taskPrevStep

**Action description：execute last step**

**Description：N/A**

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. taskPrevStep |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Set default task

**Name：**robot. setDefaultTask

**Action description：set default task**

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| taskId | String | Y | Task Id |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. setDefaultTask |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Task test switch

**Name：**robot.taskTest

**Action description：set task test switch**; If task test switch is on, nav will not be brought up by starting task, only actions are to be executed.

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| taskTest | Int | Y | 1 means on, 0 means off； |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. taskTest |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |

## Robot control command



### Go forward

**Name：**robot.goForward

**Action description：**go forward

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot.goForward |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |

### Go backward

**Name：**robot. moveBack

**Action description：**go backward

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. moveBack |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Turn left

**Name：**robot. turnLeft

**Action description：turn left**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. turnLeft |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Turn right

**Name：**robot. turnRight

**Action description：**turn right

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. turnRight |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Stop walking

**Name：**robot. stopWalking

**Action description：**stop walking

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. stopWalking |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Stop all

**Name：**robot. stopAll

**Action description：stop everything**（walking, nav, sing, dance, task）

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. stopAll |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Dancing

**Name：**robot. dance

**Action description：**Dancing

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. dance |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Stop dancing

**Name：**robot. stopDance

**Action description：stop dancing**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. stopDance |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Move

**Name：**robot.move

**Action description：**move

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| linearVelocity | String | Y | Linear velocity；  Pass Float converted String |
| angularVelocity | String | Y | Angular velocity;  Pass Float converted String |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. move |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Turn head right

**Name：**robot. turnHeadRight

**Action description：turn head right**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. turnHeadRight |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Turn head left

**Name：**robot. turnHeadLeft

**Action description：turn head left**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. turnHeadLeft |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Turn head up

**Name：**robot. turnHeadUp

**Action description：turn head up**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. turnHeadUp |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Turn head down

**Name：**robot. turnHeadDown

**Action description：Turn head down**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. turnHeadDown |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Head reset

**Name：**robot. turnHeadReset

**Action description：Head reset**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. turnHeadReset |
| resultCode | Int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Nod

**Name：**robot. headUpDown

**Action description：**nod

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. headUpDown |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Shake

**Name：**robot. headLeftRight

**Action description：shake**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. headLeftRight |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Set LED ring

**Name：**robot. light

**Action description：set LED ring**

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| type | String | Y | LED mode:  lightNormal Normal  lightTalking Talking  lightThinking Thinking  lightSinging Singing |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. light |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

## System command



### Acquire media volume

**Name：**sys.getMusicVolume

**Action description：acquire media volume**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.getMusicVolume |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Set media volume

**Name：**sys.setMusicVolume

**Action description：set media volume**

**Description：**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Comment** |
| volume | Int | Y | Volume |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.setMusicVolume |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Volume up

**Name：**sys.setVolumeAdd

**Action description：volume up**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.setVolumeAdd |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Volume down

**Name：**sys. setVolumeDec

**Action description：volume down**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.setVolumeDec |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Talk / stop talking

**Name：**sys.say

**Action description：send speech text**; send punctuation to stop, e.g. “,”

**Description：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| words | String | Text for speech |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.say |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Sing

**Name：**robot. singsong

**Action description：**sing a song

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. singsong |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Stop singing

**Name：**robot. stopSing

**Action description：stop singing**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | robot. stopSing |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Music-play music

**Name：**sys.player.music.play

**Action description：**play music

**Description：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| url | String | Music file path or url |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.player.music.play |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Music-pause music

**Name：**sys.player.music. pause

**Action description：pause music**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.player.music. pause |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Music-resume

**Name：**sys.player.music. resume

**Action description：**resume play

**Description：N/A**

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.player.music. resume |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Music-stop

**Name：**sys.player.music. stop

**Action description：stop playing**

**Description：**

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.player.music. stop |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Video-play

**Name：**sys.player.video.play

**Action description：play video**

**Description：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| url | String | Video file path or url |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.player.video.play |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Video-pause

**Name：**sys.player.video. pause

**Action description：pause video**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.player.video. pause |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Video-resume

**Name：**sys.player.video. resume

**Action description：resume video**

**Description：N/A**

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.player.video. resume |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Video-stop

**Name：**sys.player.video. stop

**Action description：stop video**

**Description：N/A**

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.player.video. stop |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Show face

**Name：**sys.showFace

**Action description：show robot face**

**Description：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| faceType | String | faceTalk  faceSmile |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.showFace |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Hide face

**Name：**sys.hideFace

**Action description：hide robot face**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys. hideFace |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Slide show-start

**Name：**sys.showpics.start

**Action description：**start slide show

**Description：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| picList | List<String> | List of picture path |

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.showpics.start |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

### Slide show-stop

**Name：**sys.showpics.stop

**Action description：stop slideshow**

**Description：**N/A

**Return：**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Comment** |
| cid | String | robot ID |
| notifyAction | String | sys.showpics.stop |
| resultCode | int | error code, 200 means success, otherwise see error code description； |
| notifyInfo | String | Error info |
| notifyParams | String |  |

## Error code description

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Comment** |
| CODE\_OK | 200 | success |
| CODE\_ERROR | 400 | Common error, see notifyInfo |
| CODE\_ERROR\_TIMEOUT | 401 | Time out |
| CODE\_ERROR\_IO | 402 | File r/w error |
| CODE\_ERR\_NAME\_EMPTY | 403 | Name can’t be empty |
| CODE\_ERR\_NAME\_DUPLICATE | 405 | Overlapped name |
| CODE\_ERROR\_INVALID\_PARAMS | 406 | Invalid param |
| CODE\_ERROR\_STOP\_SWITCH\_IS\_OPEN | 407 | E-stop has been pressed |
| CODE\_ERROR\_INVALID\_CMD | 408 | Invalid command |
| CODE\_ERROR\_RESULT\_EMPTY | 409 | Result is empty |
| CODE\_ERROR\_ALREADY\_START | 410 | Already started, should be stopped first |