# Robot State Details – Robot and Plugin interaction

Raja Software

Last updated: 04/07/2014

|  |  |  |
| --- | --- | --- |
| Date | Version | Author/Comments |
| 04/07/2014 | 0.1 | Initial draft |

This is a quick write-up on how the robot and the UI layer need to set and get the robot state details. Recent Linux and Android Rosie simulator versions store the robot state details on the server. If the robot is currently cleaning, the state details contain the cleaning category, cleaning modifier, spot dimensions, etc. A new plugin API has been exposed on the Android plugin interface to get this information from the server. (The iOS plugin will be updated shortly).

## Robot Interaction

* Whenever, the robot state changes, it calls setRobotProfileDetails3 with key “robotCurrentState"” to set the current state – this is an integer value that sets the state, but does not contain any additional information.
* To set additional details, it call call setRobotProfileDetails3 with the key “robotCurrentStateDetails” – it can set both keys in the same setRobotProfileDetails3 call.

The value for “robotCurrentStateDetails” has the following format:

{"robotCurrentState:"<stateId>", "robotStateParams":<JSON Object with state details>}}

* robotCurrentState is the same integer determining the state of the robot.
* robotStateParams contains the extra params associated with the state. This depends on the robotCurrentState value, and can be empty also.
* As an example, when the robot is spot cleaning, the values could be:

{"robotCurrentState: 10002, robotStateParams: { robotCleaningCategory:3, robotCleaningMode: 1, extraStateParams:{robotSpotCleaningArealength:5, robotSpotCleaningAreaheight:3}}}

* For a non-spot cleaning, the values could be:

{"robotCurrentState: 10002, robotStateParams: { robotCleaningCategory:<category>, robotCleaningMode: <mode>, extraStateParams:{}}}

* If the robot has stopped cleaing the robotStateParams would be an empty object.

Basically, whenever the robot state changes, it should set 2 keys in set robot profile details (can be done in a single web service call):

* robotCurrentState like before
* robotCurrentStateDetails - with format {"robotCurrentState:"<stateId>", "robotStateParams":<JSON Object with state details>}}

## Plugin Native Layer

Note: As of 04/07/2014, this has been implemented on Android. The iOS changes will follow shortly.

A new RobotManagerPlugin plugin API getRobotCurrentCleaningDetails has been implemented – this internally calls getRobotProfileDetails2 for key "robotCurrentStateDetails". The plugin does NOT process the response – it simply returns the complete value to the JS layer.

The details of the plugin method are:

Method: RobotMgr.prototype.getRobotCurrentCleaningDetails

parameters: RobotId

Success Callback:

{robotCurrentStateDetails: <state details>}

So in case of spot cleaning the success result would be like:

{robotCurrentStateDetails: {"robotCurrentState: 10002, robotStateParams: { robotCleaningCategory:3, robotCleaningMode: 1, extraStateParams:{robotSpotCleaningArealength:5, robotSpotCleaningAreaheight:3}}}}

UI Layer Responsibility

The UI layer should call the method RobotMgr.prototype.getRobotCurrentCleaningDetails to get the details of the current state of the robot, and update the UI depending on the robot state.

End.