Roomie Remote
HTTP Activity & Device Command Interface

Version 9 – Updated September 2024

Roomie Remote supports starting activities and executing device commands over a standard HTTP interface made available on port 47147. The HTTP interface sends and receives JSON payloads for each API. JSON responses are of the form:

```
{
   “st” :     <string: status of “success” | “fail” | “error”>,
   “da” :     <JSON-type: response data>,
   “co” :     <integer: HTTP status code>
}
```

If the “status (st)” field returns “success”, the request was handled successfully. A value of “fail” indicates that there was a problem with the request data, and “error” indicates that a problem occurred while handling the request.

The “data (da)” response field will contain a JSON data type that will vary depending on which API was called. Details are described in the response section of each documented API below.

List Activities

Returns a list of all activities present in the active Roomie configuration. Each activity is listed along with some informative properties as well as the activity UUID. The activity UUID is necessary for starting an activity with the runactivity API.

Request: **GET /api/v1/activities**

Response data: array of activity objects as shown below.

```
[
   {
      “icon” :      <string: Activity icon name>,
      “roomuuid” :  <string: UUID of the room containing the activity>,
      “name” :      <string: Activity name>,
      “toggle” :    <boolean: The output toggle state after starting activity>,
      “type” :      <string: “off” if the activity is a PowerOff activity>
      “uuid” :      <string: Activity UUID>
   },
   { … }
]
```

NOTE: Toggle activities are listed twice. The toggle “on” activity UUID has a suffix of “+”, and the “off” activity UUID has a suffix of “-“. In addition to the activity UUID suffix, the activity name will contain either “(On)” or “(Off)” to indicate the toggle state that will be set when the activity is started.

Example:

```
$ curl -X GET 'http://localhost:47147/api/v1/activities'
{
  "st" : "success",
  "da" : [
   {
      "icon" : "logo-tivo",
      "roomuuid" : "D3579B0E-6E36-4215-9CED-06AE394D1A77",
      "name" : "Living Room: Watch TiVo",
      "uuid" : "4A4A6B37-BE64-4E03-9A8E-81ED0C7727DA"
   },
   {
      "icon" : "curtains",
      "roomuuid" : "D3579B0E-6E36-4215-9CED-06AE394D1A77",
      "name" : "Living Room: Shades (On)",
      "toggle" : true,
      "uuid" : "2FC827F4-FF07-42A7-9289-1B767E8E90E0+"
   },
   {
      "icon" : "curtains",
      "roomuuid" : "D3579B0E-6E36-4215-9CED-06AE394D1A77",
      "name" : "Living Room: Shades (Off)",
      "toggle" : false,
      "uuid" : "2FC827F4-FF07-42A7-9289-1B767E8E90E0-"
   }],
   "co" : 200
}
```

Run Activity

Returns a list of all devices present in the active Roomie configuration. Each device is listed along with several informative properties as well as the device UUID. The device UUID is necessary for sending a command to specific device.

Request: **POST /api/v1/runactivity**

```
{
   “au” : <(required) string: Activity UUID to start>,
   “ts” : <(optional) string: “on” | “off”. Sets activity toggle state>,
   “de” : <(optional) integer: delay in seconds before starting activity>
}
```

NOTE: If present, the “delay” value must be > 1.0

Response data: None

Example:

```
$ curl -X POST 'http://localhost:47147/api/v1/runactivity' -d ‘{ "au" :
"53B7B23B-F70F-498D-A128-477FCBD05A58" }'

{
   "st" : "success",
   "da" : {
   },
   "co" : 200
}
```

List Devices

Returns a list of all devices present in the active Roomie configuration. Each device is listed along with several informative properties as well as the device UUID. The device UUID is necessary for sending a command to specific device.

Request: **GET /api/v1/devices**

Response data: array of device objects as shown below.

```
[
   {
      “address” :   <string: Device IP address>,
      “port” :      <integer: Device port>,
      “model” :     <string: Device model>,
      “uuid” :      <string: Device UUID>,
      “roomname” :  <string: Name of the room containing device>,
      “brand” :     <string: Device brand>,
      “type” :      <string: Device type>,
      “name” :      <string: Device name>,
      “roomuuid” :  <string: UUID of the room containing device>
   }
]
```

Example:

```
$ curl -X GET 'http://localhost:47147/api/v1/devices'
{
   "st" : "success",
   "da" : [
   {
      "address" : "10.0.0.49",
      "port" : 8060,
      "model" : "All Models",
      "uuid" : "A94CF6C9-D458-4A96-92D2-7C00F795F023",
      "roomname" : "Living Room",
      "brand" : "Roku",
      "type" : "Player",
      "name" : "Roku Media Player",
      "roomuuid" : "D3579B0E-6E36-4215-9CED-06AE394D1A77"
   },
   {
      "address" : "10.0.0.122",
      "port" : 4998,
      "model" : "All Models",
      "uuid" : "4DA23B69-DFF5-4D24-873E-6418A2D3777E",
      "roomname" : "Living Room",
      "brand" : "Samsung",
      "type" : "TV",
      "name" : "Samsung TV",
      "roomuuid" : "D3579B0E-6E36-4215-9CED-06AE394D1A77"
   },
   {
      "address" : "10.0.0.18",
      "port" : 3005,
      "model" : "Home Theater",
      "uuid" : "2DCCA135-4264-41B1-842B-8118337C71DE",
      "roomname" : "Office",
      "brand" : "Plex",
      "type" : "Player",
      "name" : "Plex Media Player",
      "roomuuid" : "876926CA-C99A-4A6C-BF46-15AFC1932B2C"
   }],
   "co" : 200
}
```

Send Device Command

Send a command to a specific device, with an optional repeat parameter. The response will include a session identifier that can be used to refresh a repeating command. If a command repeat option is specified, the refresh API will need to be called frequently to continue the repeating command. If the refresh API is not called, the repeating command will time out after a short duration.

Note: Despite specifying device commands in an array, the API currently is limited to accepting and handling a single command.

Request: **POST /api/v1/sendcommands**

```
{
   “de” :     <(optional) string: Delay in ms before sending commands>,
   “cs”:      <(required) Array of command objects. Details below>
}
```

Command object:

```
{
   “ty” :     <(required) string: type “command” | “activity” | “url”>,
   “de” :     <(optional) string: Delay in milliseconds. Usage varies by type>
   “pa” :     <(required) object: Params fields vary by type. Details below>
}
```

“command” params object:

```
{
   “cm” :        <(required) string: Command name in caps>,
   “device” :    <(required) string: Device UUID of the target device>,
   “cd” :        <(optional) object: Conditions to meet before sending command>
   {
      “au” :     <(required) string: Toggle activity UUID>,
      “ts” :     <(required) string: “on” | “off”>
   },
   “pd” :        <(optional) string: Power delay in milliseconds>,
   “dl” :        <(optional) string: “yes” | “no” Delay devices globally>,
   “re” :        <(optional) string: “constant” | “progressive” Repeat type>,
   “pa” :        <(optional) Params array of strings: Up to 3 command parameters>
}
```

“activity” params object:

```
{
   “au” :     <(required) string: Activity UUID>,
   “ts” :     <(optional) string: “on” | “off” Toggle state to set>,
}
```

“url” params object:

```
{
   “ur” :     <(required) string: URL to load. e.g. http://www.google.com/>
}
```

NOTE: The specified URL must include the scheme (e.g. “http://“)

Response data: Session identifier. The session can be used to refresh or end any repeating commands. NOTE: Sessions are very short-lived to prevent repeating commands from continuing unmonitored.

```
{
   "se" :  "96F4C281-B34D-47D4-9097-EA80874FAFE4"
}
```

Example:

```
$ curl -X POST ‘http://localhost:47147/api/v1/sendcommands' -X POST -d ‘{ "de" :
"1000", "cs" : [ { "ty" : "command", "pa" : { "cm" : "VOLUME UP", "device" :
“2DCCA135-4264-41B1-842B-8118337C71DE”, “re” : “constant” } } ] }’

{
   "st" : "success",
   "da" : {
      "se" : "96F4C281-B34D-47D4-9097-EA80874FAFE4"
   },
   "co" : 200
}
```

Refresh Command Session

Refresh a command session for a repeating command. After sending the repeating command with the sendcommands API, the returned session identifier must be used to refresh the session and keep the repeat timer alive. If the session isn’t refreshed soon enough, it will expire and the repeated command will stop. To avoid excessive repeats that might lead to extremely loud volume levels (as an example), a session will automatically expire after 0.3 seconds of sending the command. If it’s desired that the command should continue repeating, the refresh API must be called more frequently that every 0.3

seconds. It is not necessary to call the refreshcommands API if sendcommands was used without a repeat option.

Request: **POST /api/v1/refreshcommands**

```
{
   “se” :  <(required) string: Command session identifier>
}
```

Response data: Session identifier. The passed-in session identifier is echoed in the response.

```
{
   "se" : "96F4C281-B34D-47D4-9097-EA80874FAFE4"
}
```

Example:

```
$ curl -X POST ‘http://localhost:47147/api/v1/refreshcommands' -X POST -d ‘{ "se"
: “96F4C281-B34D-47D4-9097-EA80874FAFE4" }’

{
   "st" : "success",
   "da" : {
      "se" : "96F4C281-B34D-47D4-9097-EA80874FAFE4"
   },
   "co" : 200
}
```

Stop Command Session

Stop a command session for a repeating command. After sending the repeating command with the sendcommands API, the command will repeat for up to 0.3 seconds before the session expires and stops the command. The stopcommands API can be called to stop a repeating command sooner than the 0.3 session timeout. It is not necessary to call the stopcommands API if sendcommands was used without a repeat option.

Request: **POST /api/v1/stopcommands**

```
{
   “se” :  <(required) string: Command session identifier>
}
```

Response data: Session identifier. The passed-in session identifier is echoed in the response.

```
{
   "se" : "96F4C281-B34D-47D4-9097-EA80874FAFE4"
}
```

Example:

```
$ curl -X POST ‘http://localhost:47147/api/v1/stopcommands' -X POST -d ‘{ "se" :
“96F4C281-B34D-47D4-9097-EA80874FAFE4" }’

{
   "st" : "success",
   "da" : {
      "se" : "96F4C281-B34D-47D4-9097-EA80874FAFE4"
   },
   "co" : 200
}
```