Creating a Smart Home Application

Published 2014-10-28 | (Compatible with SDK 5.0)

> This document shows how to use Smart Home API's for fetching device status and changing their current status.

Contents

**** This class will not be supported in 2015.**

*All functionalities of AppsFramework are more improved, integrating with CAPH. Therefore Appsframework is not supported since 2015 Smart TV. To use functionalities of Appsframework, refer to **here**.*

This tutorial provides information on how to create a Smart Home Application that uses Smart Home API using webapis.js. The application can be tested with Samsung Smart TV Emulator 5.0.

Prerequisites

To create a Smart Home application you'll need SDK version 5.0 and 2014_Smart_TV_Emulator_5_0 which are available on SamsungDForum.

Source Files

> Note
>
> The files needed for the sample application are **here**.

Introduction

The SmartHome Application is used to get the details of connected devices and display them over the main screen. Thus, allowing user to select the desired device to see its current status and change it.

> Note

> Currently devices supported are Air Conditioner, Refrigerator and Washer. Currently API supports change of current status for AC only.

Interfaces description

Webapis.js is used for fetching devices and device status. Currently, this JavaScript file uses default test data. This default data can be seen under CSHTestData in Webapis.js

Webapis.js

This provides set of APIs to get device list and their status. Currently it uses test data. It's related APIs that are used in this application are discussed in further sections of this document.

Using tutorial application

1. When Application is initiated, Webapis.js is loaded.
   <script type="text/javascript" language="javascript" src="$MANAGER_WIDGET/Common/webapi/1.0/webapis.js"></script>
2. Main screen is loaded with devices: Air Conditioner, Refrigerator and Washer.

Main Screen View

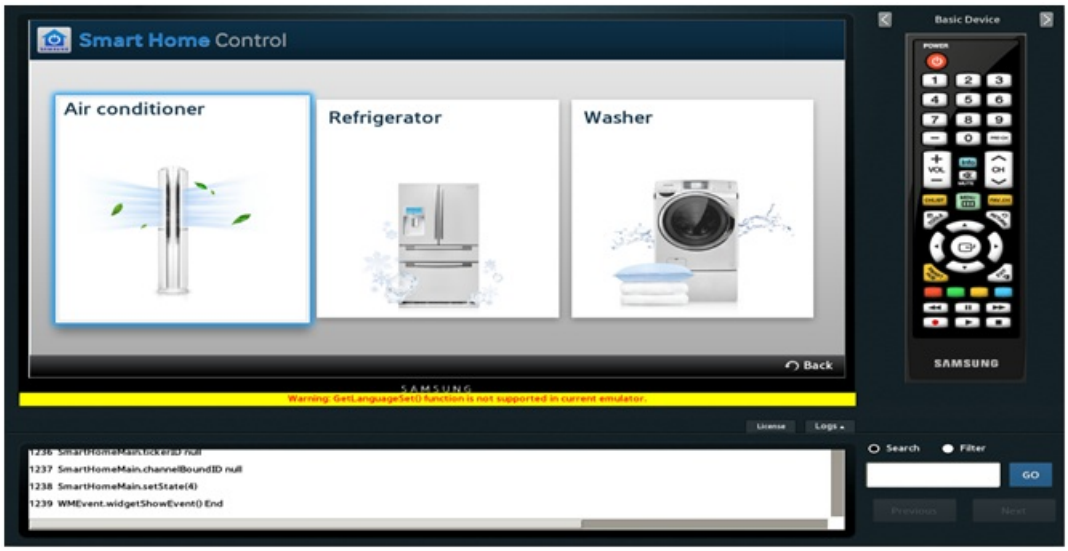

Figure 1: Main View

3. Clicking one of device icon will open detailed view of selected device.
4. Below is the detailed view Of Air Conditioner.

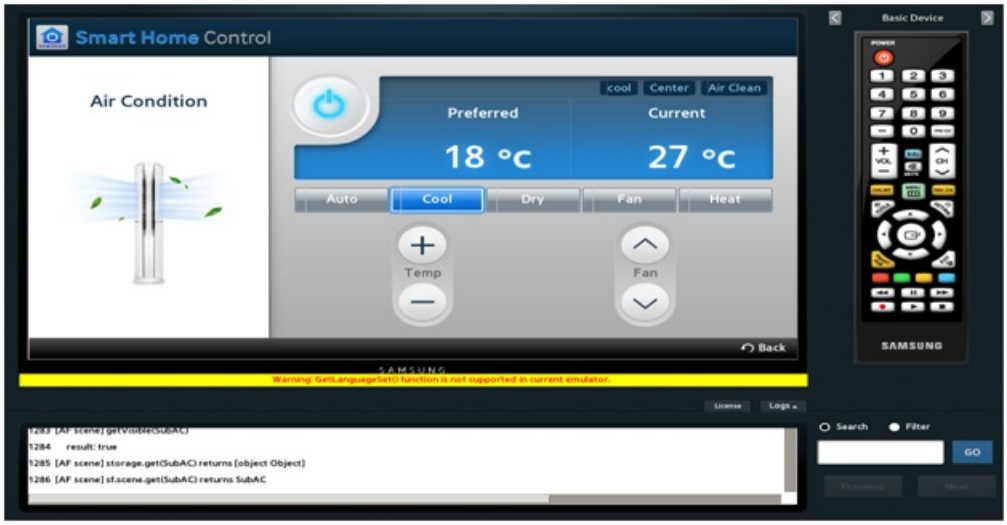

Figure 2: AC Detailed View

5. Detailed view shows current status of Air conditioner. Here Power On-Off, Operation Mode [Auto, cool, Dry, Fan, Heat] can be selected and changed. Wind speed and temperature can be changed using the Fan and Temp buttons respectively.

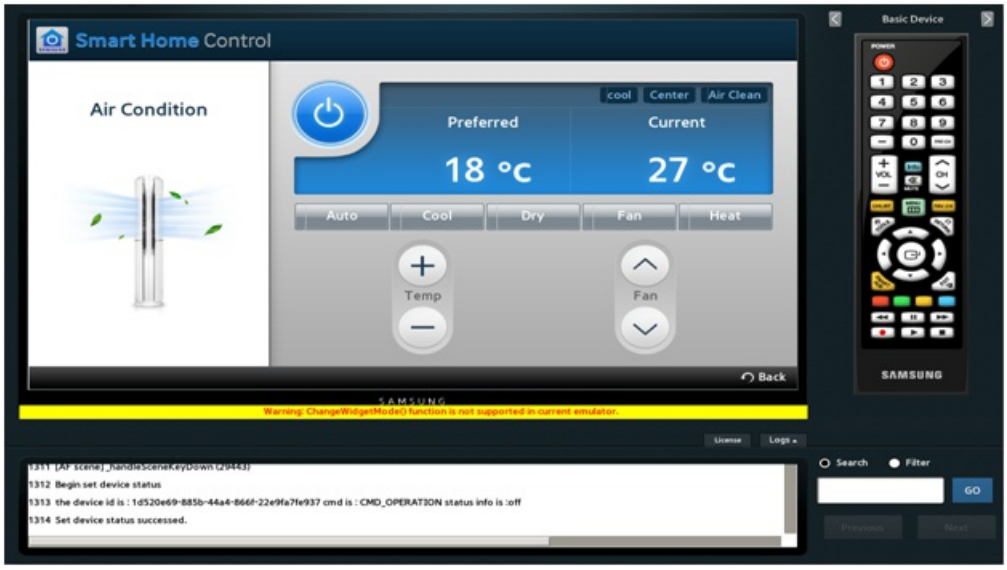

Figure 3: AC Power Status ON

6. When the Air Conditioner is powered off, buttons will be hidden from the view and the user defined values of Current temperature, desired temperature and Wind Speed will not be visible.

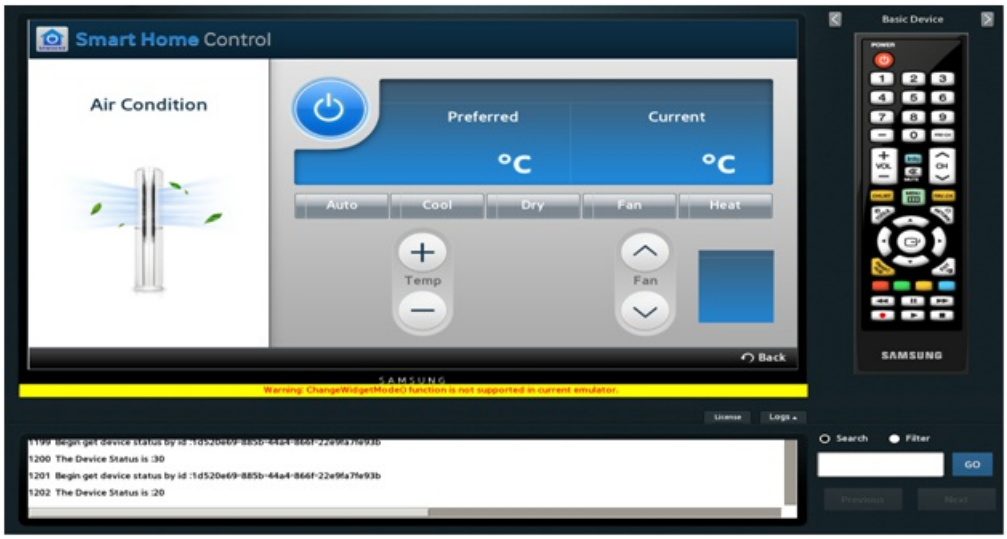

Figure 4: AC Power Status OFF

7. Temperature/Wind Speed can be changed as shown below:

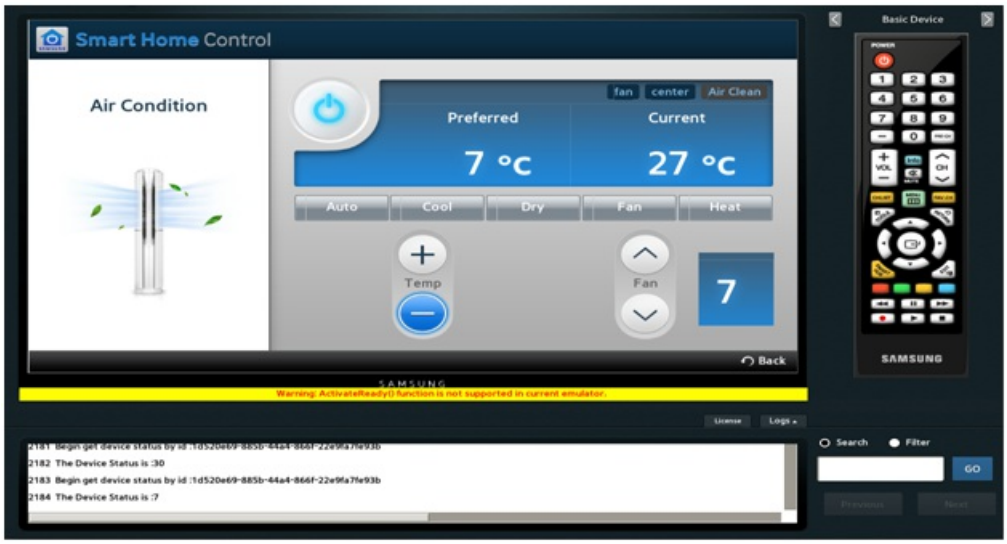

Figure 5: Temperature Change from 18 to 7

Figure 6: AC Operation Mode Dry

Detailed View of the Refrigerator

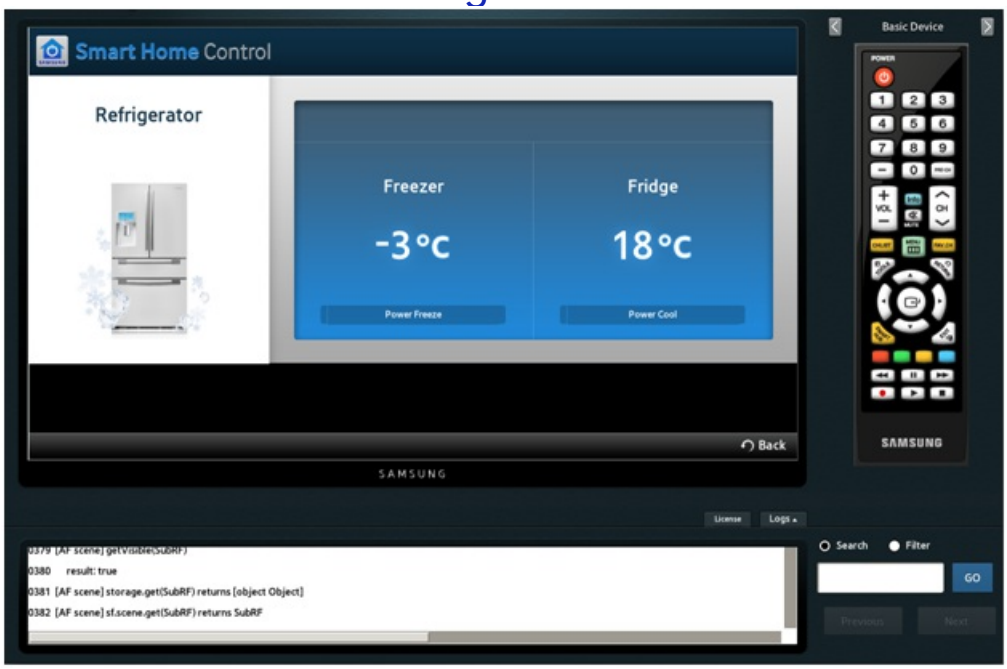

Figure 7: Refrigerator

1. In the detailed Refrigerator view, the values for the freezer and fridge temperature can be fetched using SmartHome APIs for Refrigerator and the same are displayed on the screen.

Detailed View of Washer

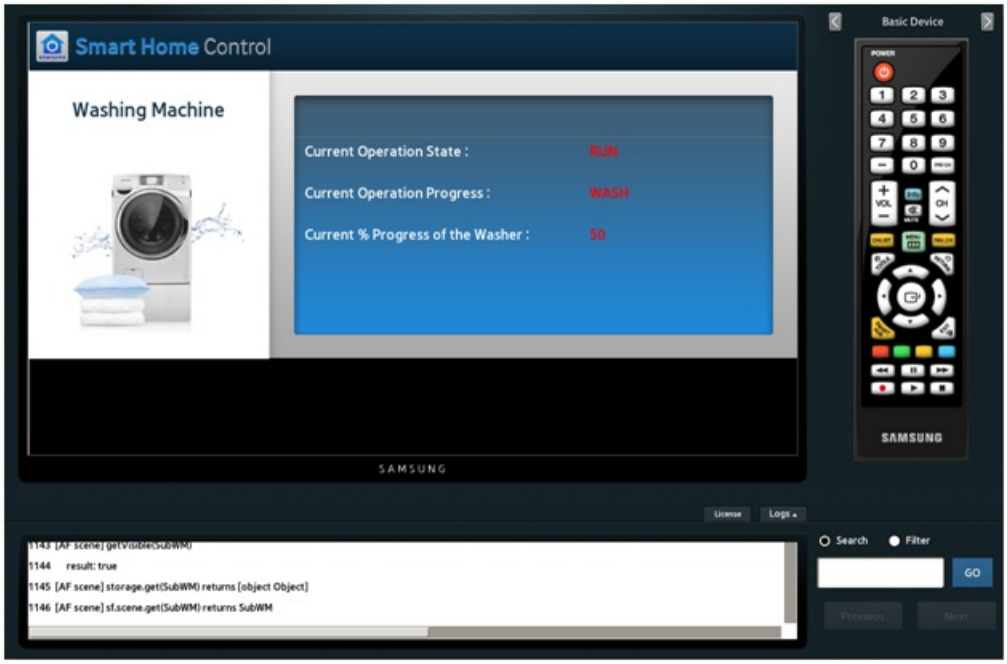

Figure 8: Washing Machine

1. In the detailed Washer view, SmartHome APIs are used to fetch the details of Operation State, Operation Progress and %age progress of the washer.

Creating your own application

Loading Webapis.js

Include below statement in beginning of <head> tag to load Webapis.js.

```
<script type="text/javascript" language="javascript" src="$MANAGER_WIDGET/Common/webapi/1.0/webapis.js"></script>
```

This JavaScript file will provide API's for fetching device list and their status. After loading the JS file, initialize the connection using below code:

```
if (webapis.smarthome.init() == false) {
    alert("Fail to init SmartHome API");
}
```

This returns true if connection is initialized otherwise false.

Fetching the Device List

Device list can be fetched using below API:

```
var deviceIdList = webapis.smarthome.getDevices();
```

It returns array of Device Ids. These Device Ids are used to get device details like device name, device status etc. Device Ids are also used for updating device states. Below are the API's used to fetch Device Name using Device ID:

```
var deviceName = webapis.smarthome.getDeviceName(deviceId);
```

Device Name is kind of device category. That is Ref, AC, Washer etc.. Each Device Name is associated with device types. Device Types can be fetched using below API:

```
var deviceType = webapis.smarthome.getDeviceType(deviceId);
```

Device current status can be fetched using below API:

```
var deviceStatus = webapis.smarthome.getDeviceStatus(deviceId, webapis.smarthome.SMART_HOME_CMD.CMD_OPERATION);
```

This function takes two parameters.

1. Device Id
2. Command for which status is required

Device Current status can be manipulated using below API:

```
var success = webapis.smarthome.setDeviceStatus(deviceId, webapis.smarthome.SMART_HOME_CMD.CMD_OPERATION, webapis.smarthome.OPERATION_VALUE.POWER_ON);
```

This API takes three parameters:

1. Device ID
2. Command for which status change is required.
3. Desired Status value

This returns true if Status updated successfully, else it returns false.

Currently, setDeviceStatus supports change of AC Power, AC Desired Temperature and wind speed and AC Operation Mode.

Below is the code snippet for changing AC Desired Temperature:

```
this.runTempAndWindSet = function () {
    var power = this.getPowerState();
    var cursor = iAC.getDataCursor();
    var macAddress = iAC.getMacAddress();
    var command = '';
    if( power == webapis.smarthome.OPERATION_VALUE.POWER_ON ) {
        switch( cursor ) {
            case 6 : command = this.expectTemp + 1 ; break; // Temp Up
            case 7 : command = this.expectTemp - 1 ; break; // Wind Down
            case 8 : command = this.windLevelSpeed + 1 ;    break; // Wind Up
            case 9 : command = this.windLevelSpeed - 1 ;    break; // Wind Down
        }
        if( cursor == 6 || cursor == 7 ){
            webapis.smarthome.setDeviceStatus(iAC.getMacAddress() , webapis.smarthome.SMART_HOME_CMD.CM
D_AC_TEMPERATURE_DESIRED , command );
            this.setExpectTemp( command );
            this.initTemp(this.getExpectTemp(), this.getCurrentTemp() );
        }
        else if ( cursor == 8 || cursor == 9 ){
            webapis.smarthome.setDeviceStatus(iAC.getMacAddress() , webapis.smarthome.SMART_HOME_CMD.CM
D_AC_WIND_SPEEDLEVEL , command );
            this.setWindLevelSpeed( command );
            this.initWindSpeed( this.getWindLevelSpeed() );
        }
    } else {
        //Nothing
    }
};
```