

Adding a Scene

Published 2014-10-28 | (Compatible with SDK 2.5,3.5,4.5,5.0,5.1 and 2011,2012,2013,2014 models)

Adding a scene to application.

Contents

Setting the Show/Hide, and Focus/Blur States

Show/Hide/Focus the Scene with Parameter

Initializing Application with Scene

Getting Scene Instance

Supporting Multi-Resolution Scenes

**** 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](#).

In order to add use scene-based architecture in your application, you must add the scenes to your application's [project files](#).

To add a scene to your application:

1. Add the scenes (Array) item in app.json.
2. Add the Scene ID to scenes array item.
3. Add the HTML, CSS, and JS files to the app/html, app/stylesheets, and app/scenes directories respectively.
4. [Set the show/hide, and focus/blur states.](#)
5. [Initialize the scene-based application.](#)
6. [Get the scene instance.](#)
7. If needed, [add support for multiple resolutions](#).

Setting the Show/Hide, and Focus/Blur States

You can call the [Scene Manager API \(sf.scene\)](#) methods to:

Show each scene. The figure below illustrates the hiding sequence for a sample scene named 'Menu'.

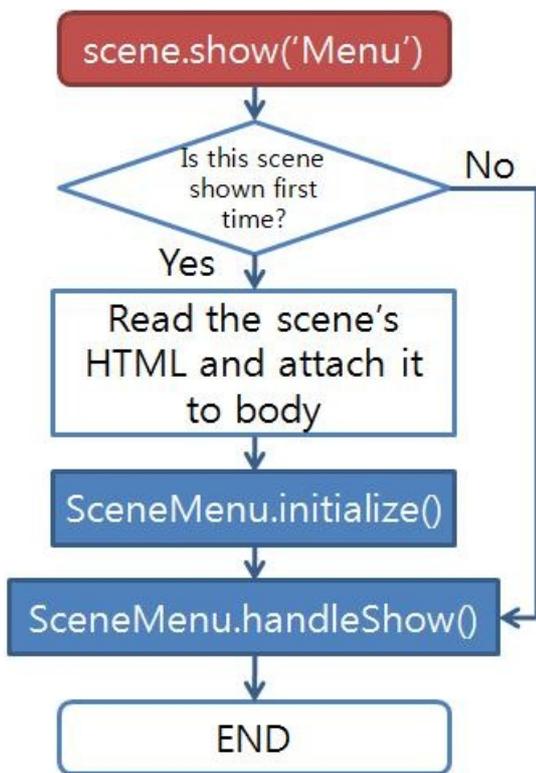


Figure. Showing the Menu scene sequence

Focus or blur each scene. The figure below illustrates the focus sequence for a sample scene named 'Menu'. Note that you do not need to call a blur method. When a scene is focused, the scene that was previously focused is blurred by default.

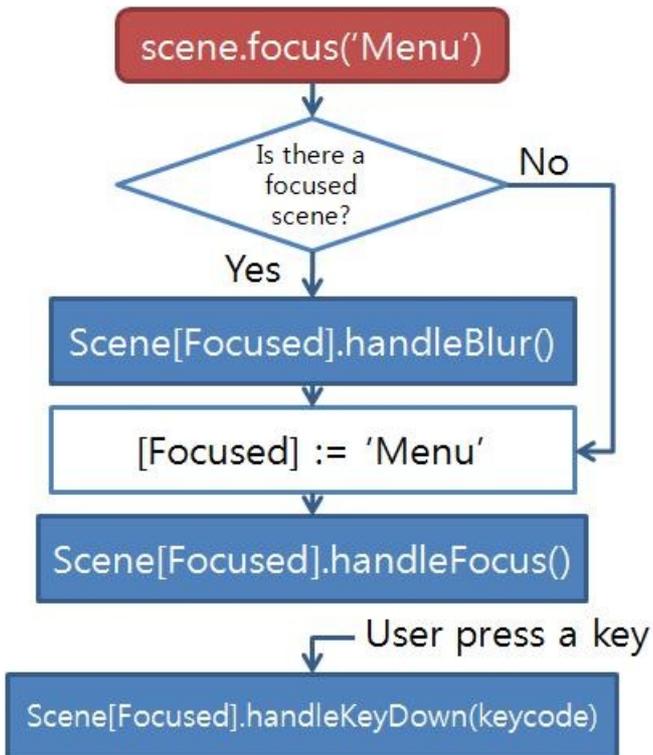


Figure: Focusing the Menu scene sequence

Hide each scene The figure below illustrates the hiding sequence for a sample scene named 'Menu'.

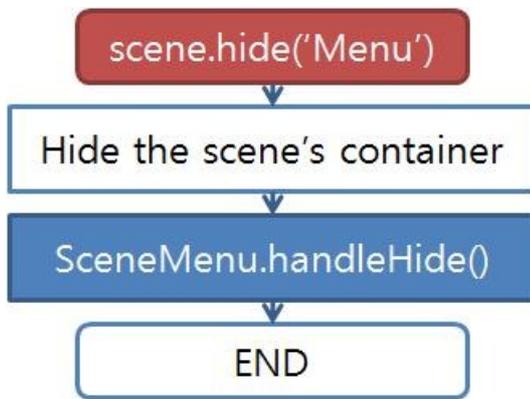


Figure. Hiding the Menu scene sequence

Show/Hide/Focus the Scene with Parameter

The methods `show`, `hide`, and `focus` accept a second parameter. When used, this parameter is passed to the `handleShow`, `handleHide`, and `handleFocus` methods of the scene.

```

SceneMain.prototype.handleKeyDown = function (keyCode) {
  alert("SceneMain.handleKeyDown(" + keyCode + ")");
  switch (keyCode) {
    case sf.key.ENTER:
      sf.scene.show('Sub', {parentSceneId:'Main'}); // show the "Sub" scene and pass object
                                                    // to SceneSub.handleShow
      break;
  }
}

```

Initializing Application with Scene

Initialize the scene based application in the `app/init.js`, which is the entry point of applications based on the `AppsFramework`.

```

function onStart() {
  // TODO : Add your initialize code here

  // show and focus the Main scene
  sf.scene.show('Main');
  sf.scene.focus('Main');
}

function onDestroy () {
  // stop your XHR or Ajax operation and put codes to destroy your application here
}

```

The Main scene can also be shown in a separate Controller object.

Getting Scene Instance

To get a specific scene instance, use `sf.scene.get(SCENE_ID)`. The following example calls the `refresh` method defined in the Sub scene when the **ENTER** key is pressed in the Main scene.

```

SceneMain.prototype.handleKeyDown = function (keyCode) {
    var subScene;

    alert("SceneMain.handleKeyDown(" + keyCode + ")");
    switch (keyCode) {
        case sf.key.ENTER:
            subScene = sf.scene.get('Sub'); // get the "Sub" scene instance
            subScene.refresh();           // Call method of "Sub" scene
            break;
    }
}

```

Supporting Multi-Resolution Scenes

The Smart TV application supports the 960x540 resolution. But as the DTV performance has improved, higher resolutions are supported in the new DTV. You can also add support for lower resolutions for so that your application can run in older devices. The table below lists the resolutions supported by different platform versions.

model	resolution
2010(C model)	960x540(540p)
2011(D model)	960x540(540p), 1280x720(720p)

To support multiple resolutions in your application:

1. Add the resolutions property in the app.json file. The property is an Array object of String indicating the supported resolutions. For example, '540p', '720p'.
2. Add the directories supported as subdirectories to the app/stylesheets directory. For example, app/stylesheets/540p, app/stylesheets/720p.
3. Add stylesheet files corresponding to the respective resolution to the appropriate directories. For example, app/stylesheets/540p/Main.css, app/stylesheets/720p/Main.css.