Samsung Smart TV App 개발

삼성전자 VD 사업부
유영욱 선임

목차

- Samsung Smart TV
- Smart TV App
- Samsung Smart TV SDK
- Hello TV App 만들기
- Key Event 처리
- Q & A

Samsung Smart TV

Samsung Smart TV History

InfoLive (2007) | Power InfoLink (2008) | Internet@TV (2009) | Internet@TV (2010) | Smart TV (2011) | Smart TV (2012) | Smart TV (2013)

✓ Multimedia support
✓ Possible to add service

✓ TV app store launch
✓ Paid app available
(US, Korea)

✓Smart Interaction
- Voice,Gesture,Face
✓Convergence
- Allshare
- TV-Mobile Framework
✓Signature Services
✓DRM extension

✓ RSS (Text)-based
✓ Static service

✓ Service upgradable
✓ SDK release

✓Content recommendation
✓ Web browsing
✓ Full unified search

✓Recommendation
✓Performance

Smart TV App

어떤 App을 개발해야 될까?

인기 Smart TV App

동영상 관련 App


You
Tube


PLAYY


PLAYY


Dailymotion
Channels/HD Videos
Most recent
Featured
HD Videos
News & Politics
Funny
Film & TV
Music
Auto-Moto
Arts
Gaming
Webcam & Vlogs
Dailymotion
sur grand écran
Move
Select
Return

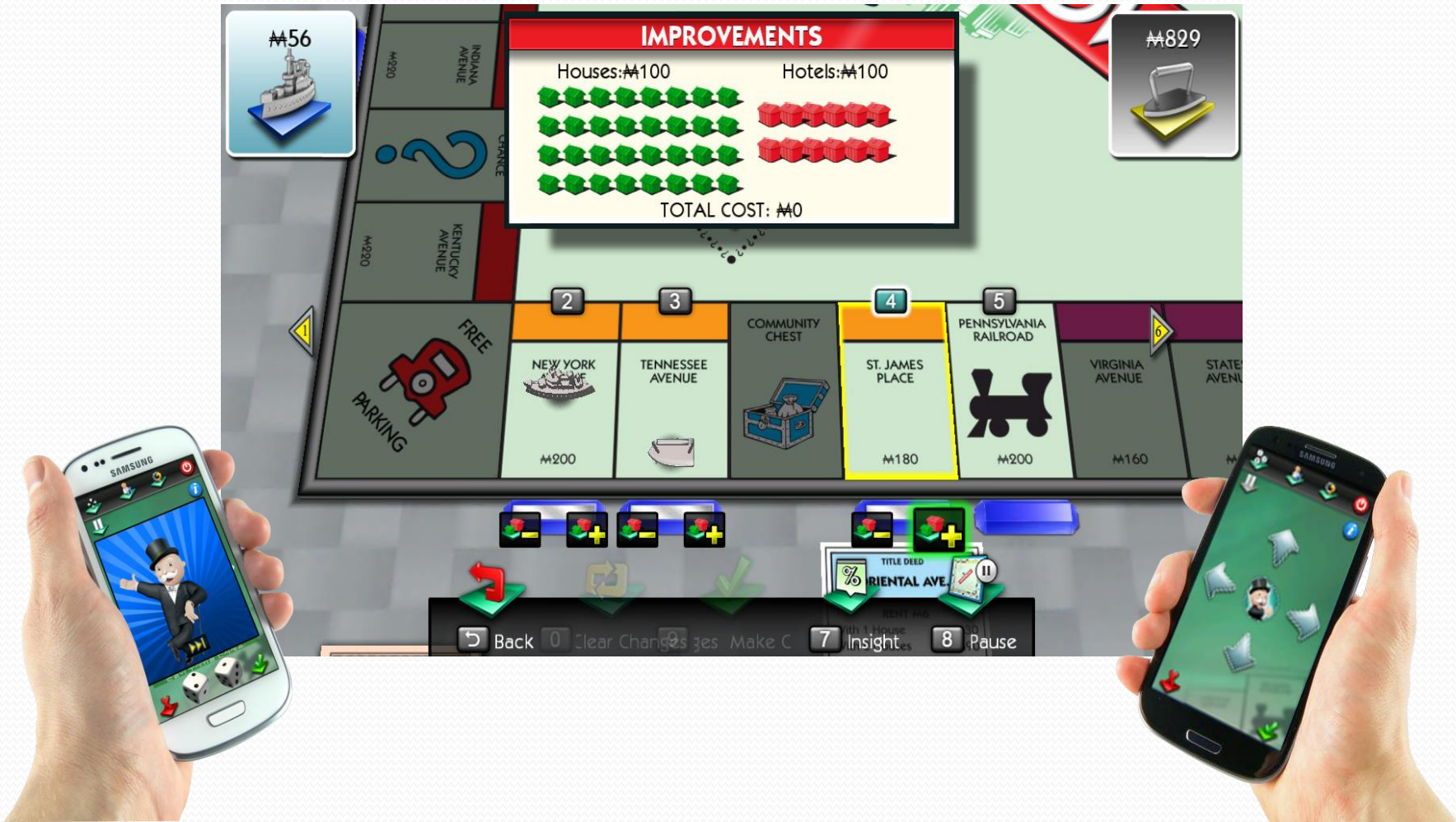

Dailymotion
Top Videos
EPIC
Dailymotion
sur grand écran
Seek
Pause
Stop
Move
Select
Stop

영상통화 & 사진, 그림 감상 App

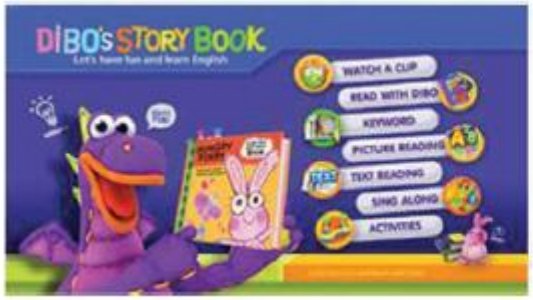
세계 최대 인터넷 전화

스카이프

Mobile Convergence App



유아용 App


DIBO's STORY BOOK
WATCH A CLIP
READ WITH DIBO
KEYWORD
PICTURE READING
TEXT READING
SING ALONG
ACTIVITIES


DIBO's STORY BOOK


Aa
apple
astronaut
ax
ant
ambulance


Story

건강관리 App

인기 Game App


STEVE
DAVID
BAKER
UNO
Alex


Choose Your Player
TEXAS


Level
2/20
Moves
0
Help
EXIT


Checkmate
Richard I
Saladin

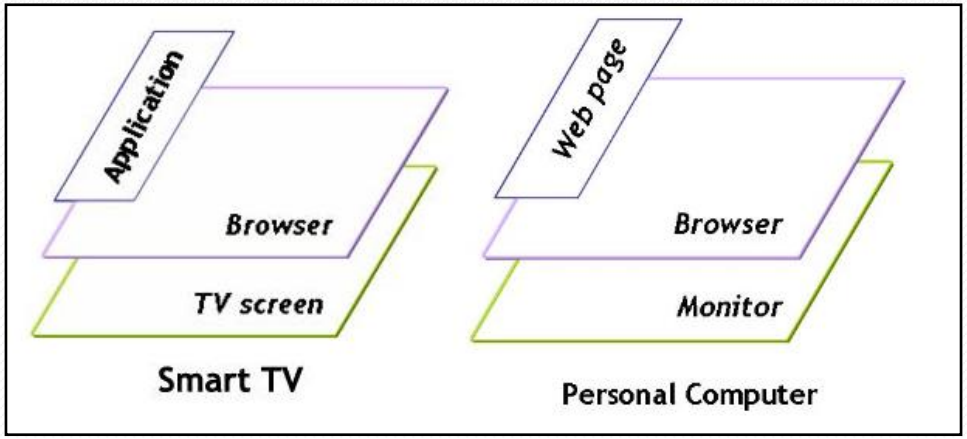
인기 App의 특징?

어떤 App을 개발해야 할까?

- TV의 대형 화면을 이용
- 조작이 간단한 App
- 리모컨으로 조작이 어렵다면 모바일 기기를 이용. (Convergence)
- TV를 주로 많이 보는 시청층을 공략
- 결국 TV는 무엇인가 보려는 사람들이 이용하는 기기이다.

Samsung
Smart TV App

Smart TV App



Smart TV App are web-based software programs that run on digital TVs connected to the Internet

Digital TV
Big screen
Video + Audio
Channel + Volume
Remote Control
Device Discovery
More…

Internet
Web Browsing
Web Services
Streaming Video
Social Networking
HTML
More…

Smart TV Apps

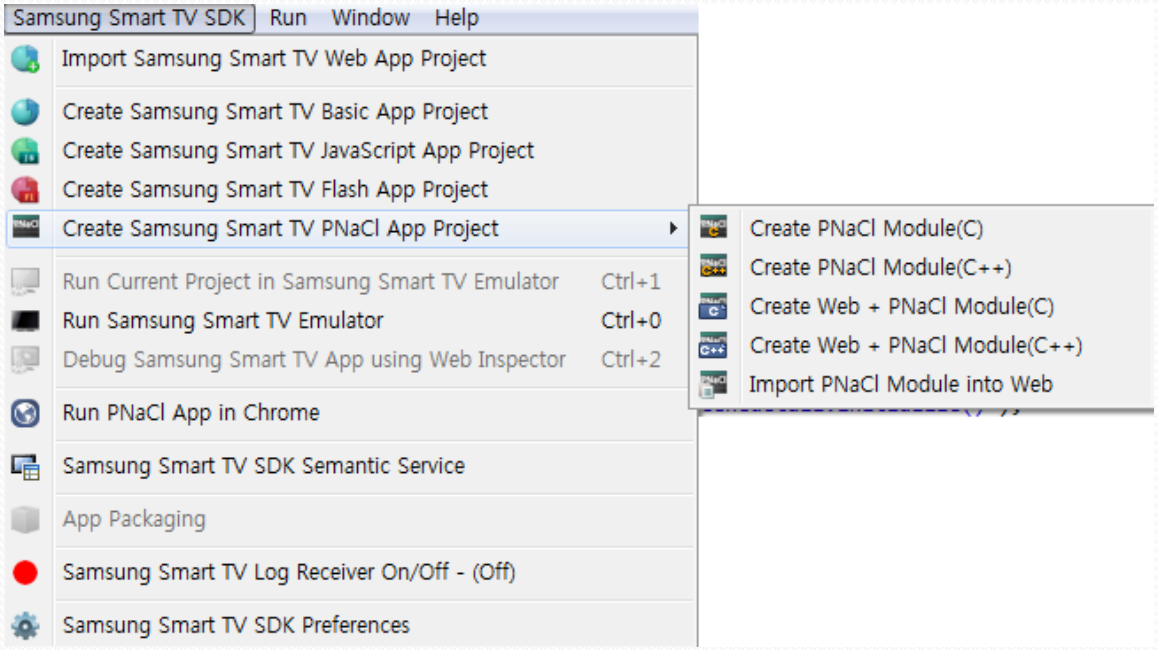
App 종류 – Area

Full Screen Application

Single-wide Application

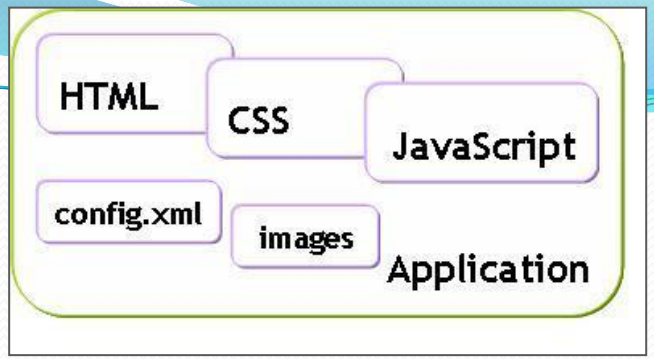
Ticker

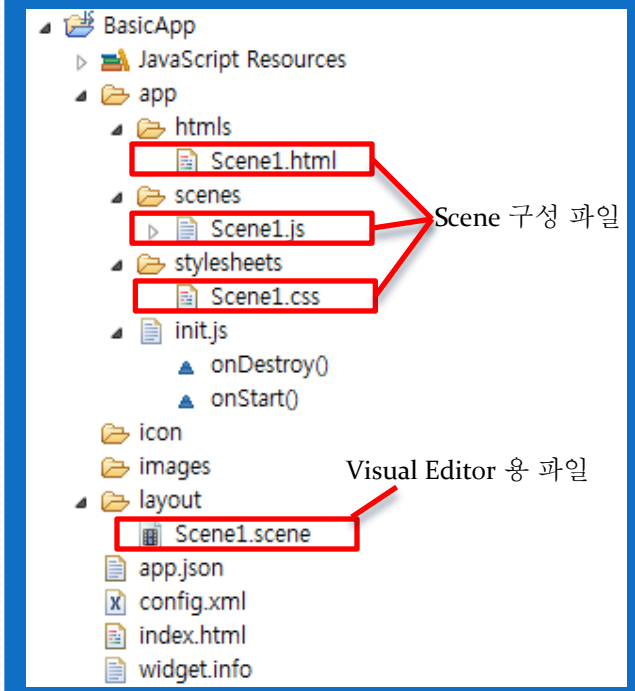
App 종류 – Project

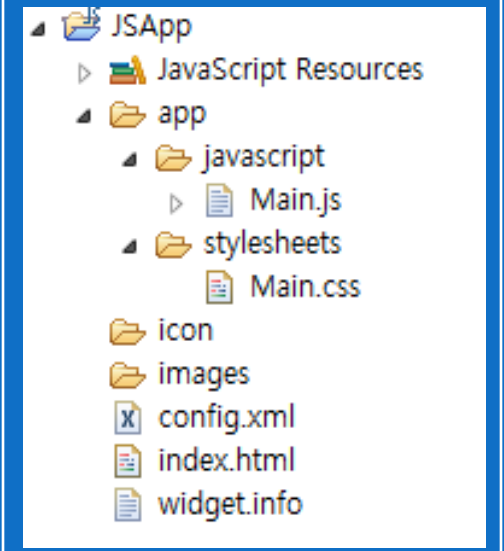


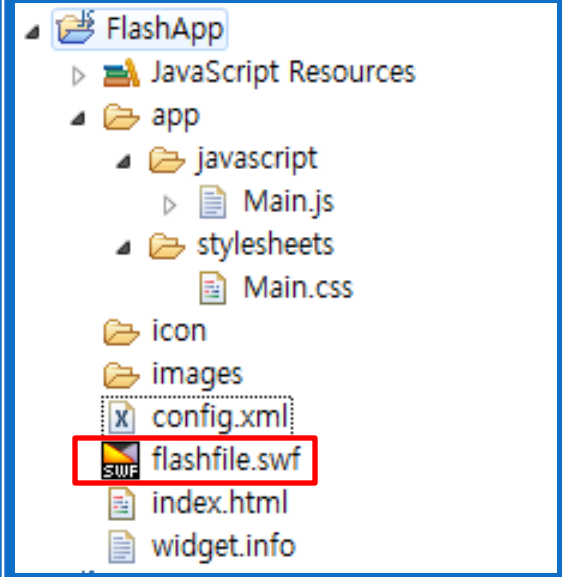
1. Basic (JS)
   - Apps Framework
   - Visual Editor
2. JavaScript (JS)
3. Flash (Flash)
4. PNaCl (Native App)

App 구성파일



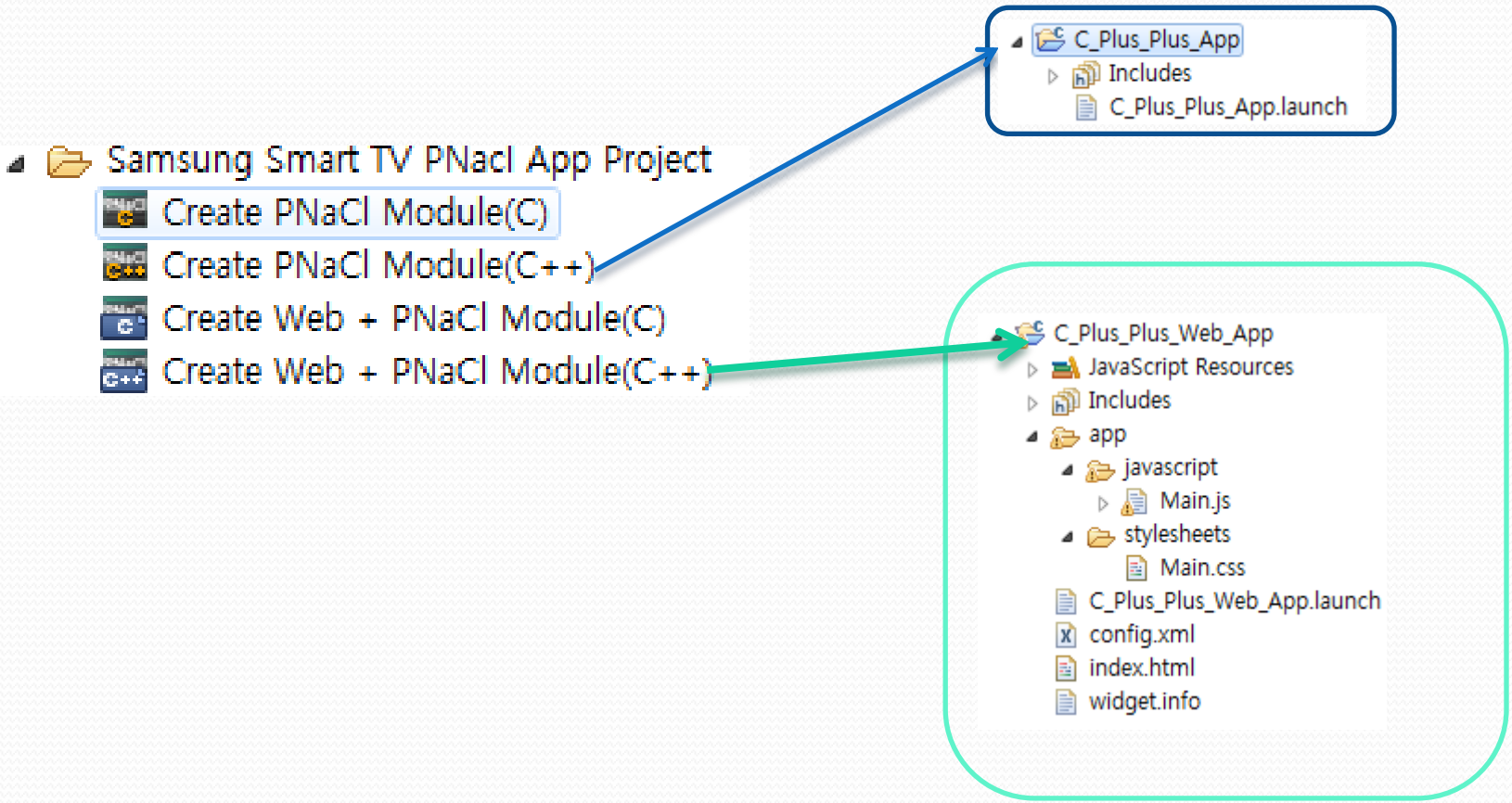




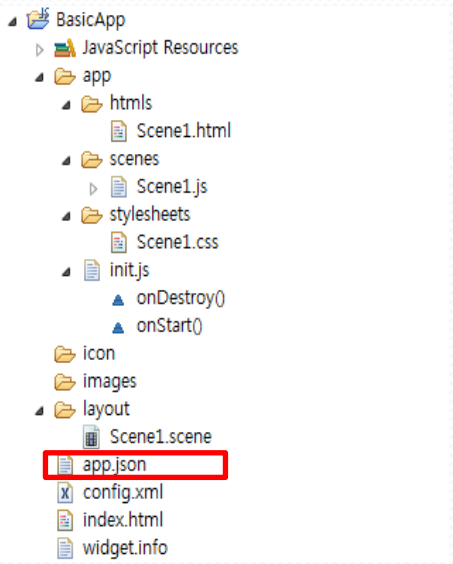


Basic App (Apps Framework)	JavaScript App	Flash App

PNaCl App 구성파일



Basic App – app.json

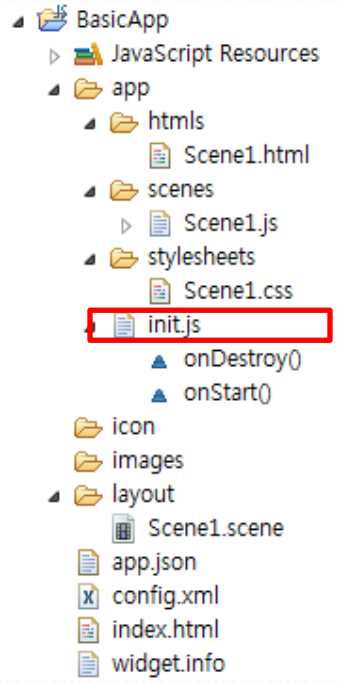



```
{
    "scenes" : [ "Scene1"],
    "files" : [],
    "theme" : "base",
     "modules" : [ "ime" ]
}
```

```
{
    "scenes" : ["Main", "Sub"],
    "files" : [],
    "theme" : "base",
    "languages" : ["en", "ko", "fr-US", "es-US", "pt-US"],
    "resolutions" : ["540p"],
    "modules" : ["ui.*"]
}
```

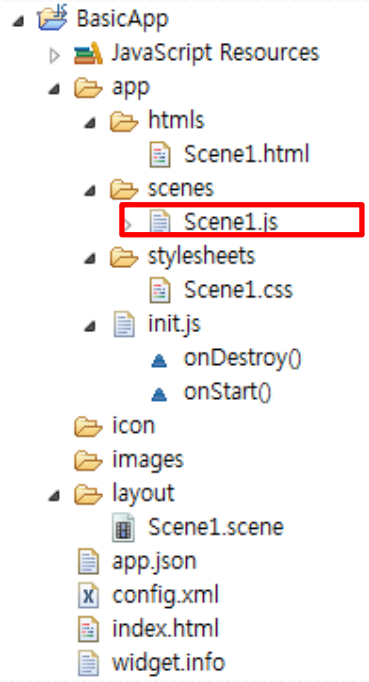
- **scenes** : App을 구성하는 Scene의 목록을 나타냄
- **files** : App 로드 시 include 할 파일들의 목록
  (<script> 태그를 사용하는 것과 동일한 효과)
- **theme** : UI 컴포넌트 테마 (현재는 base 만 존재함)
- **languages** : 지원언어
- **resolutions** : App 해상도 (**54op** – 960 x 540, **72op** – 1280 x 720)
- **modules** : App 함께 로드 할 모듈 이름 (현재 ime, sso만 지원)

Basic App – init.js



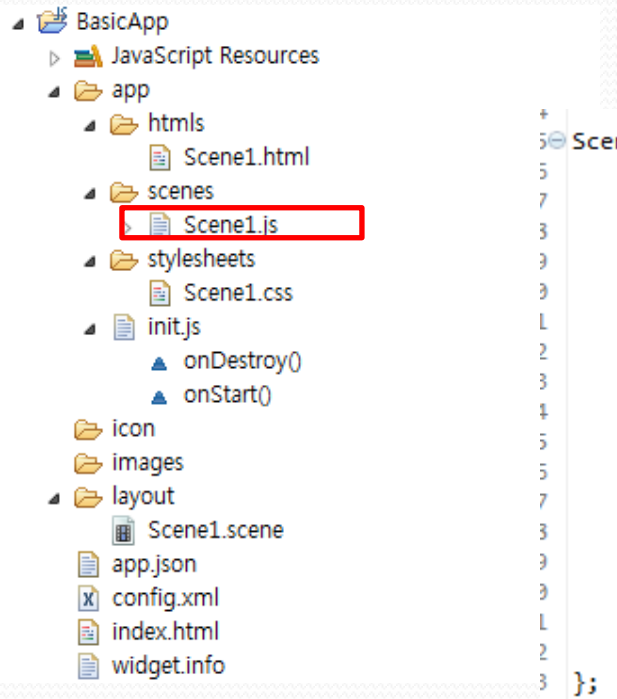

```
function onStart () {
    // TODO : Add your Initilize code here
    // NOTE : In order to start your app, call "sf.start()" at the end of this function!!

    sf.scene.show('Scene1');
    sf.scene.focus('Scene1');
}
function onDestroy () {
    //stop your XHR or Ajax operation and put codes to distroy your application here

}

alert("init.js loaded.");
```

Basic App – Scene1.js (1)



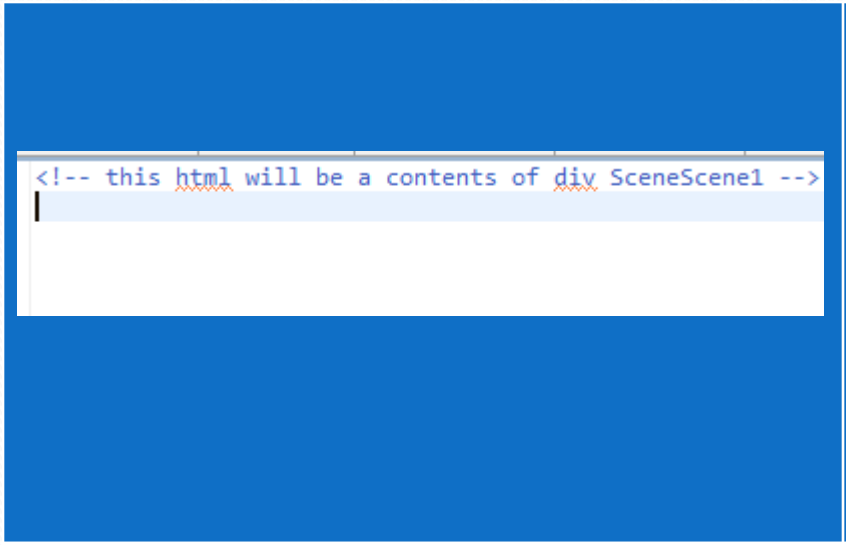

```
1  alert('SceneScene1.js loaded');
2
3  function SceneScene1() {
4
5  };
6
7  SceneScene1.prototype.initialize = function () {
8      alert("SceneScene1.initialize()");
9      // this function will be called only once when the scene manager show this scene first time
10     // initialize the scene controls and styles, and initialize your variables here
11     // scene HTML and CSS will be loaded before this function is called
12
13 };
14
15 SceneScene1.prototype.handleShow = function (data) {
16     alert("SceneScene1.handleShow()");
17     // this function will be called when the scene manager show this scene
18 };
19
20 SceneScene1.prototype.handleHide = function () {
21     alert("SceneScene1.handleHide()");
22     // this function will be called when the scene manager hide this scene
23 };
24
25 SceneScene1.prototype.handleFocus = function () {
26     alert("SceneScene1.handleFocus()");
27     // this function will be called when the scene manager focus this scene
28 };
29
30 SceneScene1.prototype.handleBlur = function () {
31     alert("SceneScene1.handleBlur()");
32     // this function will be called when the scene manager move focus to another scene from this scene
33 };
```

Basic App – Scene1.js (2)

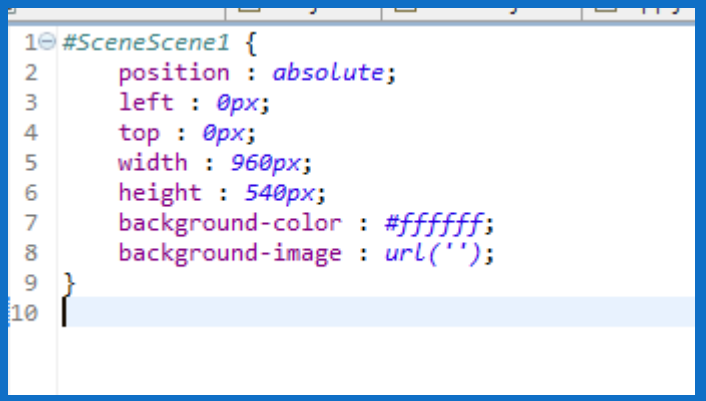



```
SceneScene1.prototype.handleKeyDown = function (keyCode) {
    alert("SceneScene1.handleKeyDown(" + keyCode + ")");
    // TODO : write an key event handler when this scene get focued
    switch (keyCode) {
        case sf.key.LEFT:
            break;
        case sf.key.RIGHT:
            break;
        case sf.key.UP:
            break;
        case sf.key.DOWN:
            break;
        case sf.key.ENTER:
            break;
        default:
            alert("handle default key event, key code(" + keyCode + ")");
            break;
    }
};
```

Basic App – Scene1.html, Scene1.css



Scene1.html
(비어있음)

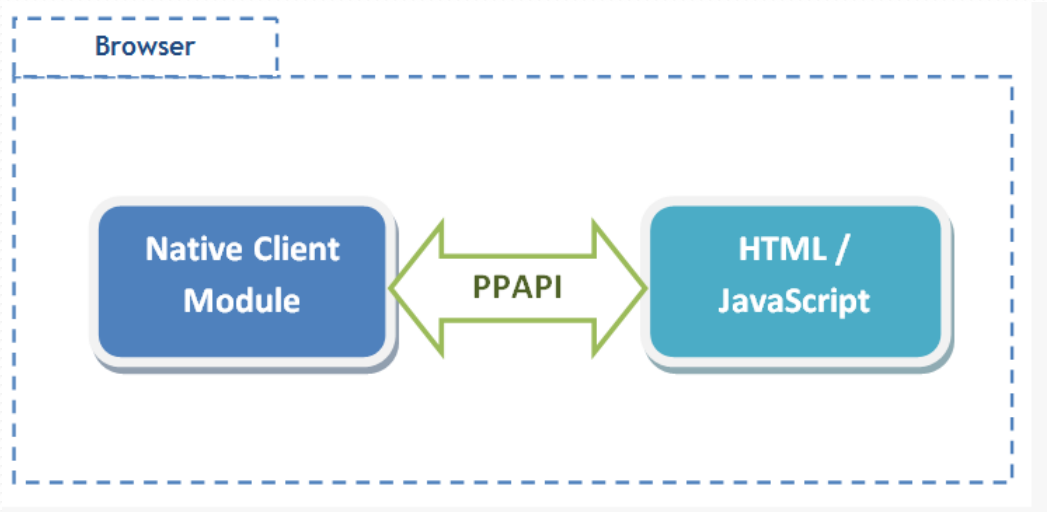


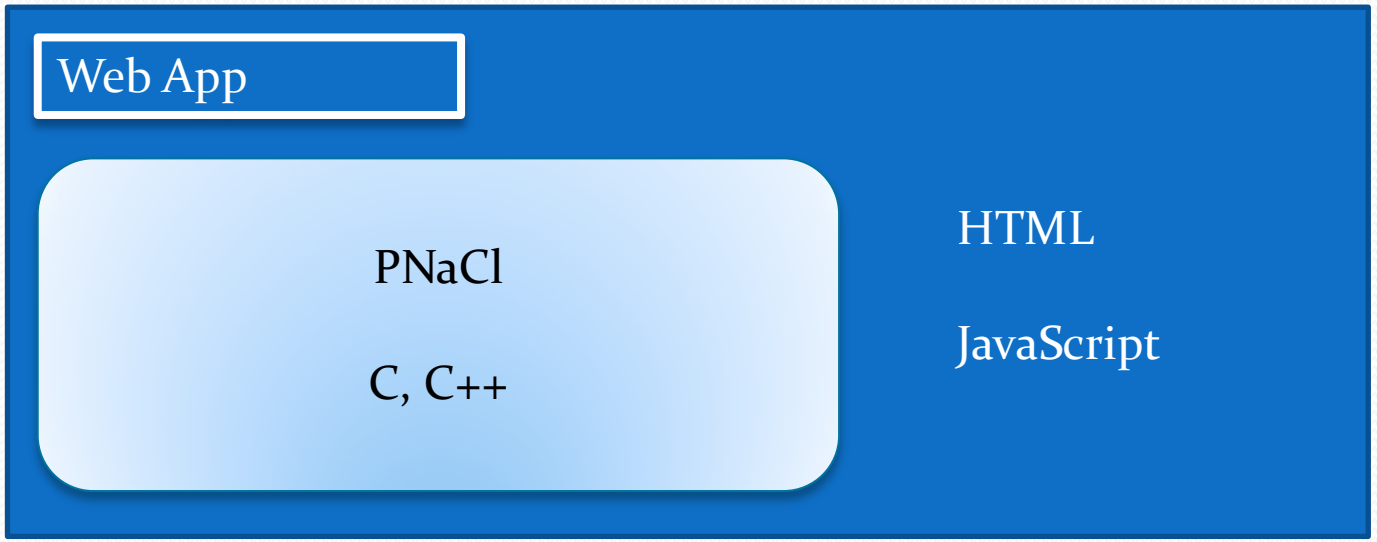
Scene1.css

Basic App – index.html

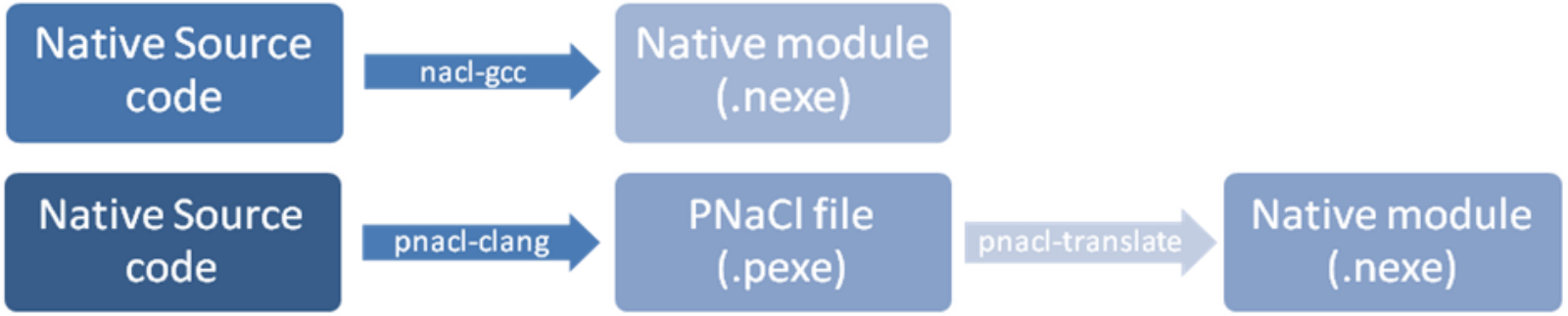
```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
5          <title>BasicApp</title>
6          <script type="text/javascript" language="javascript" src="$MANAGER_WIDGET/Common/af/2.0.0/loader.js"></script>
7      </head>
8      <body>
9      </body>
10 </html>
11
```

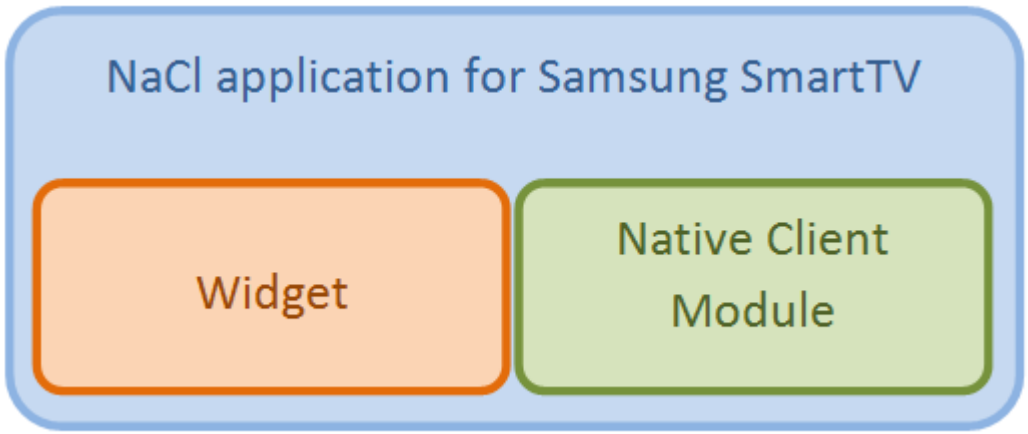
Native App (PNaCl)





Native App (PNaCl)





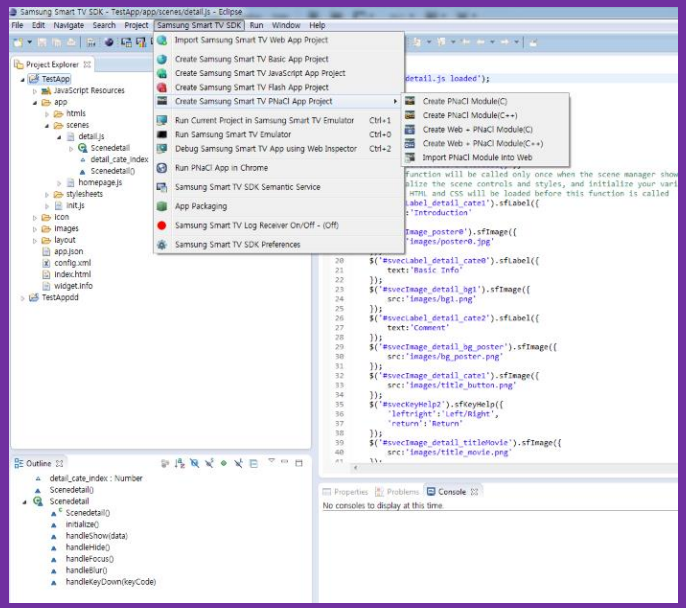
공통 – config.xml

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<widget>
    <cpname></cpname>
    <cplogo></cplogo>
    <cpauthjs></cpauthjs>
    <ThumbIcon>icon/sampleIcon_106_87.png</ThumbIcon>
    <BigThumbIcon>icon/sampleIcon_115_95.png</BigThumbIcon>
    <ListIcon>icon/sampleIcon_85_70.png</ListIcon>
    <BigListIcon>icon/sampleIcon_95_78.png</BigListIcon>
    <category></category>
    <autoUpdate>n</autoUpdate>
    <ver>0.100</ver>
    <mgrver></mgrver>
    <fullwidget>y</fullwidget>
    <type>user</type>
    <srcctl>y</srcctl>
    <ticker>n</ticker>
    <childlock>n</childlock>
    <videomute>n</videomute>
    <dcont>y</dcont>
    <widgetname>BasicApp</widgetname>
    <description> </description>
    <width>960</width>
    <height>540</height>
    <author>
        <name></name>
        <email></email>
        <link></link>
        <organization></organization>
    </author>
</widget>
```

Samsung
Smart TV SDK

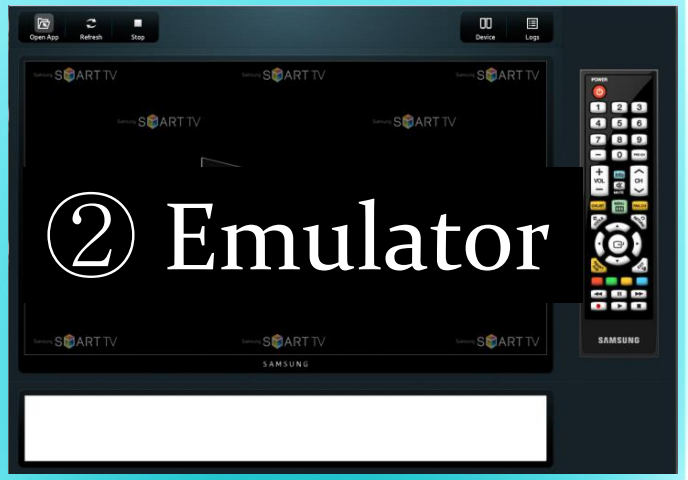
Samsung Smart TV SDK

① IDE
(Eclipse-based)



Virtual Box

Linux (Ubuntu 12.04)



② Emulator

SDK 실행 요구사항

- 지원 OS : Windows, Mac OS, Linux

- 하드웨어 요구 사항
  - Processor : Dual Core 1.5GHz / Single Core 3GHz or higher
  - RAM : 2 GB or higher
  - Screen resolution : 1280 x 1024 or higher
  - HDD : 5GB or higher

- 소프트웨어 요구사항
  - VirtualBox
  - Java Standard Edition(Java SE) 1.6 or higher

Opeartion System	Eclipse Version	Pepper Version	CDT Version	Other
Windows	Juno	Pepper_25	8.1.2	Python 2.6.x/2.7.x
Mac	Juno	Pepper_25	8.1.2	N/A
Linux	Juno	Pepper_25	8.1.2	N/A

SDK Download

2013 Main SDK

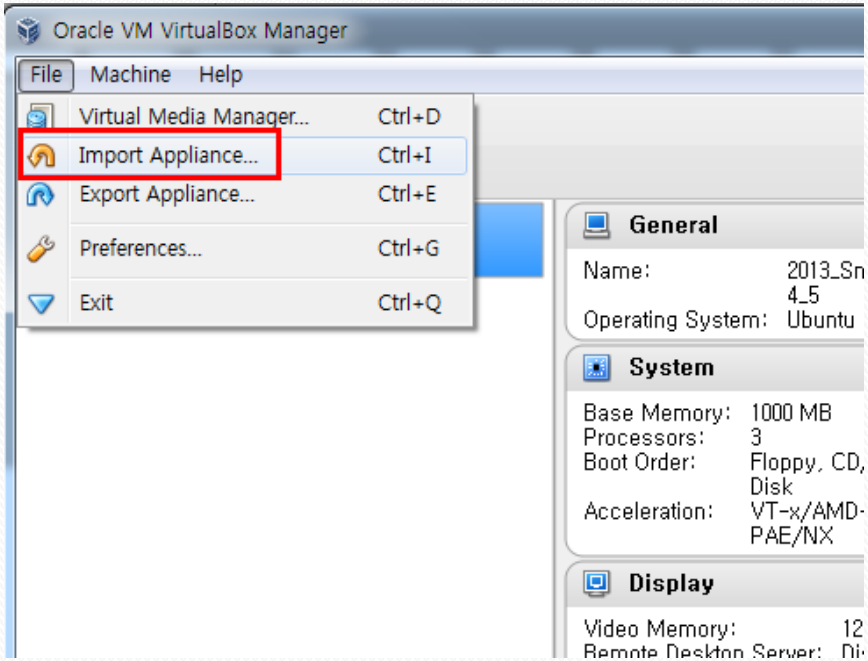
SDK 4.5

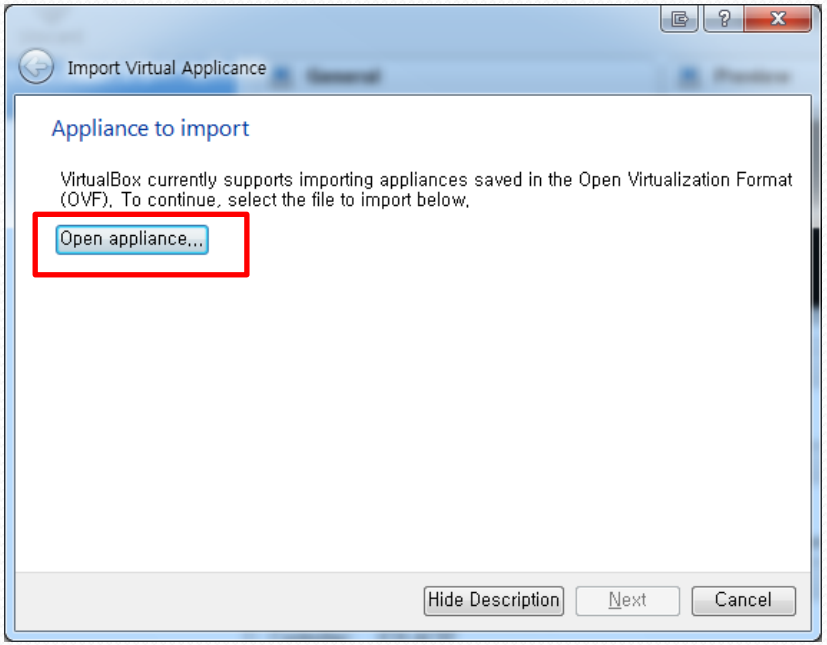
Download SDK	Size	Release Date	Download
SDK IDE Download for Windows 32bit	593 MB	2013-08-01	1332
SDK IDE Download for Windows 64bit	593 MB	2013-08-01	1986
SDK IDE Download for Linux 32bit	581 MB	2013-08-01	184
SDK IDE Download for Linux 64bit	582 MB	2013-08-01	253
SDK IDE Download for Mac OS 64bit	581 MB	2013-08-01	541
SDK Emulator Image for Virtual Box	854 MB	2013-08-01	1705
Mashup Server for Windows 32bit	44 MB	2013-08-01	120
Mashup Server for Windows 64bit	44 MB	2013-08-01	191
Local Cloud Development Environment	1007 MB	2013-08-01	265

URL : http://www.samsungdforum.com/Devtools/SdkDownload

SDK 설치

[설치방법]
IDE – JRE 설치 후 SDF에서 IDE 파일 다운로드 후 압축해제
Emulator – 1) VirtualBox를 먼저 설치
2) SDF에서 Emulator 이미지 파일 (.OVA) 다운로드
3) VirtualBox 에서 Emulator 이미지 Import

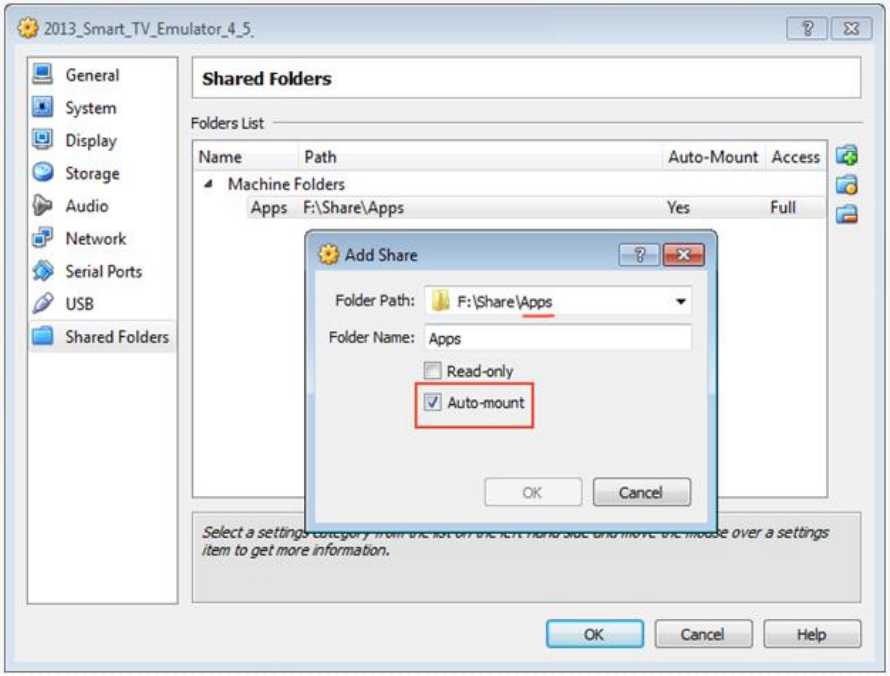




SDK 설정

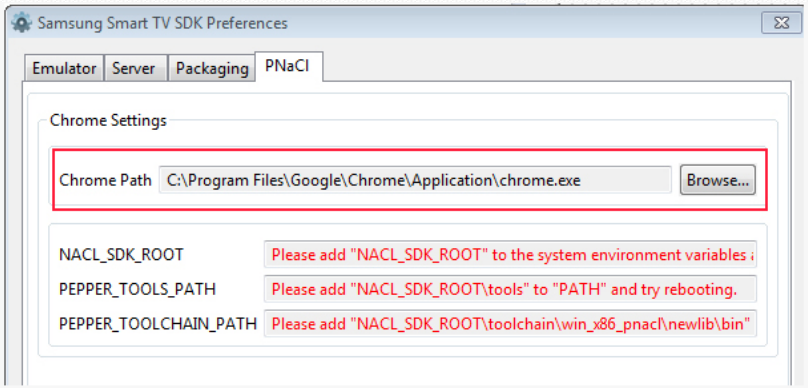
[Emulator 설정]

– Apps 폴더 설정

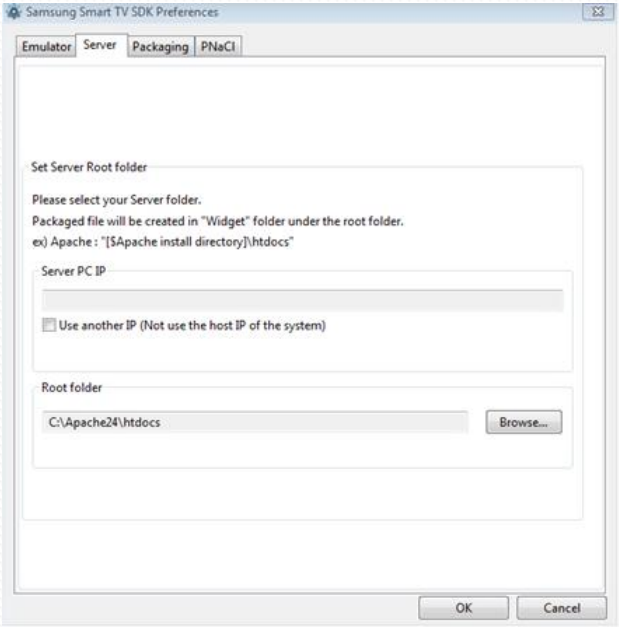


[Editor 설정]

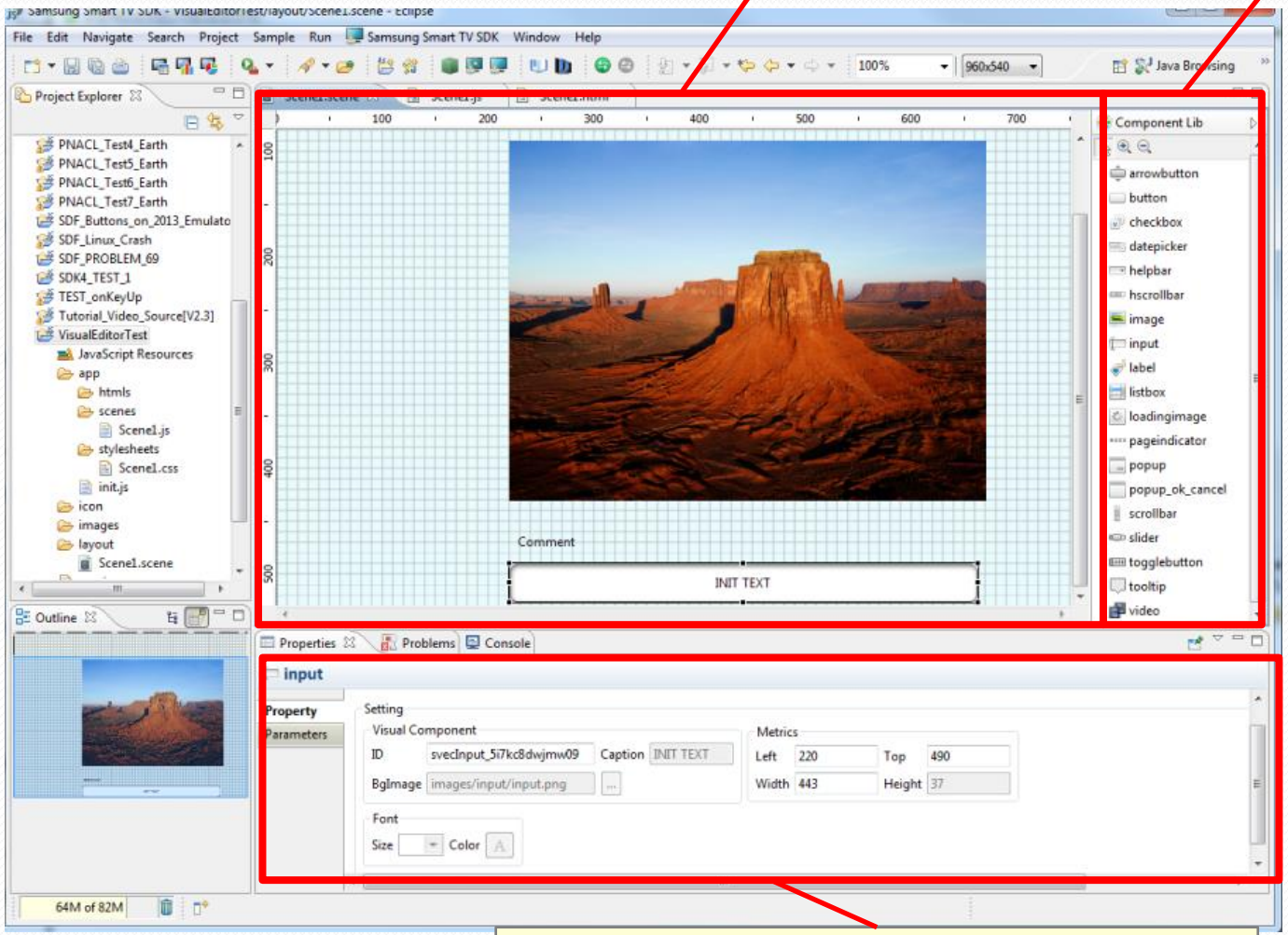
– Chrome Path



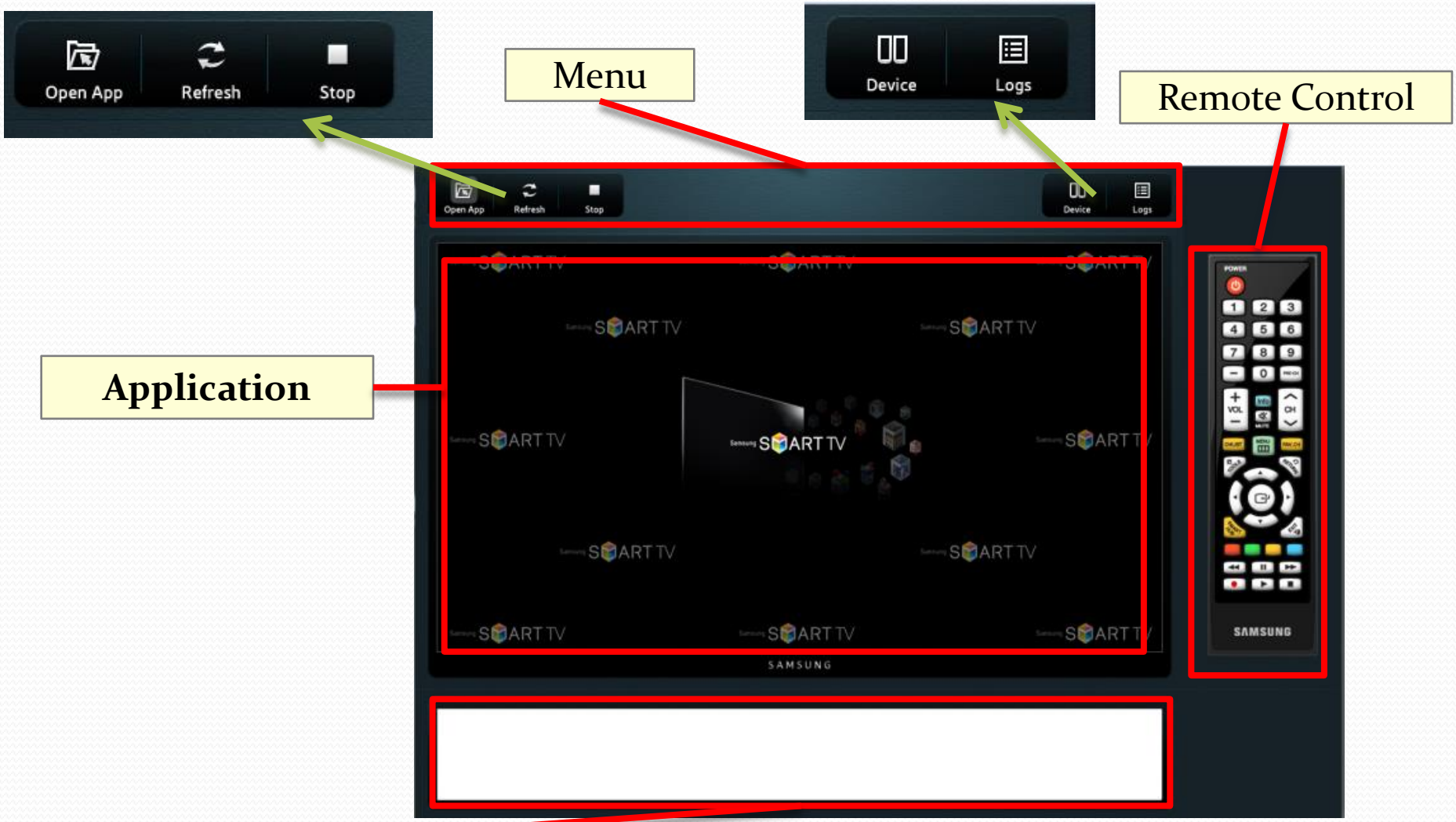
– Web Server



IDE

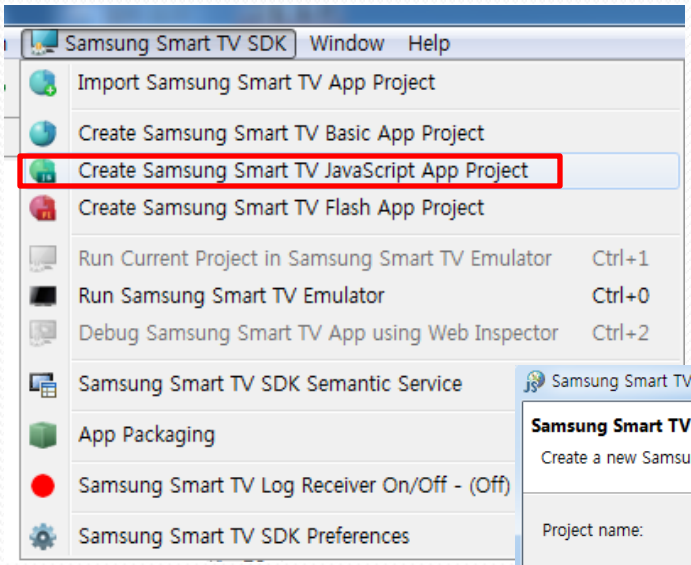

Visual Editor
GUI Components
Properties of Selected Component
File Edit Navigate Search Project Sample Run Samsung Smart TV SDK Window Help
Project Explorer
Component Lib
arrowbutton
button
checkbox
datepicker
helpbar
hscrollbar
image
input
label
listbox
loadingimage
pageindicator
popup
popup_ok_cancel
scrollbar
slider
togglebutton
tooltip
video
Comment
INIT TEXT
Outline
Properties
Problems
Console
input
Property
Parameters
Setting
Visual Component
ID
Caption
INIT TEXT
BgImage
images/input/input.png
Metrics
Left
220
Top
490
Width
443
Height
37
Font
Size
Color
64M of 82M

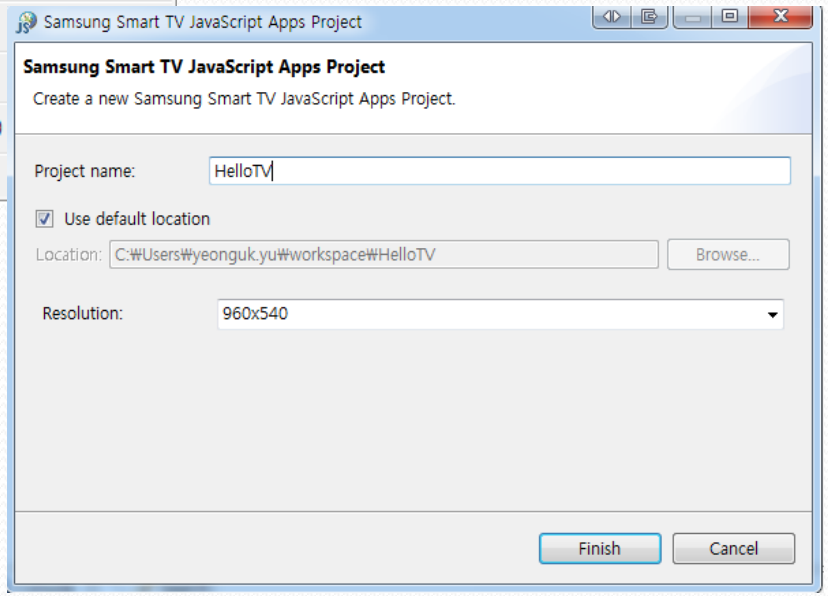
Emulator

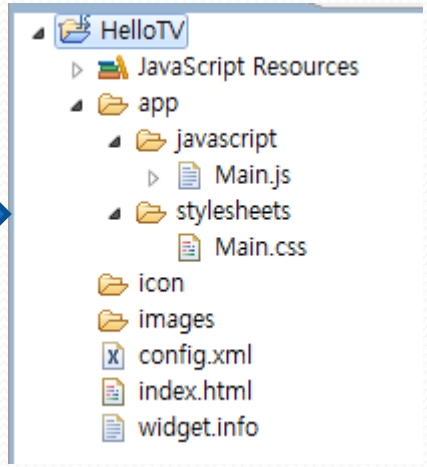

Open App
Refresh
Stop
Device
Logs
Menu
Remote Control
Application
Log Viewer
SAMSUNG

Hello TV App (JavaScript App)

프로젝트 생성


Samsung Smart TV SDK
Window
Help
Import Samsung Smart TV App Project
Create Samsung Smart TV Basic App Project
Create Samsung Smart TV JavaScript App Project
Create Samsung Smart TV Flash App Project
Run Current Project in Samsung Smart TV Emulator Ctrl+1
Run Samsung Smart TV Emulator Ctrl+0
Debug Samsung Smart TV App using Web Inspector Ctrl+2
Samsung Smart TV SDK Semantic Service
App Packaging
Samsung Smart TV Log Receiver On/Off - (Off)
Samsung Smart TV SDK Preferences


Samsung Smart TV JavaScript Apps Project
Samsung Smart TV JavaScript Apps Project
Create a new Samsung Smart TV JavaScript Apps Project.
Project name:
HelloTV
Use default location
Location:
C:₩Users₩yeonguk.yu₩workspace₩HelloTV
Browse...
Resolution:
960x540
Finish
Cancel


HelloTV
JavaScript Resources
app
javascript
Main.js
stylesheets
Main.css
icon
images
config.xml
index.html
widget.info

Hello TV App – index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
5          <title>HelloTV</title>
6
7          <!-- TODO : Common API -->
8          <script type="text/javascript" language="javascript" src="$MANAGER_WIDGET/Common/API/Widget.js"></script>
9          <script type="text/javascript" language="javascript" src="$MANAGER_WIDGET/Common/API/TVKeyValue.js"></script>
10
11         <!-- TODO : Javascript code -->
12         <script language="javascript" type="text/javascript" src="app/javascript/Main.js"></script>
13
14         <!-- TODO : Style sheets code -->
15         <link rel="stylesheet" href="app/stylesheets/Main.css" type="text/css">
16
17         <!-- TODO: Plugins -->
18
19     </head>
20
21     <body onload="Main.onLoad();" onunload="Main.onUnload();">
22
23         <!-- Dummy anchor as focus for key events -->
24         <a href="javascript:void(0);" id="anchor" onkeydown="Main.keyDown();"></a>
25
26         <!-- TODO: your code here -->
27         <label id="hello">Hello TV</label>
28     </body>
29 </html>
```

Hello TV App – main.js

```
1  var widgetAPI = new Common.API.Widget();
2  var tvKey = new Common.API.TVKeyValue();
3
4  var Main =
5  {
6
7  };
8
9  Main.onLoad = function()
10 {
11     // Enable key event processing
12     this.enableKeys();
13     widgetAPI.sendReadyEvent();
14 };
15
16 Main.onUnload = function()
17 {
18
19 };
20
21 Main.enableKeys = function()
22 {
23     document.getElementById("anchor").focus();
24 };
```

```
26 Main.keyDown = function()
27 {
28     var keyCode = event.keyCode;
29     alert("Key pressed: " + keyCode);
30
31     switch(keyCode)
32     {
33         case tvKey.KEY_RETURN:
34         case tvKey.KEY_PANEL_RETURN:
35             alert("RETURN");
36             widgetAPI.sendReturnEvent();
37             break;
38         case tvKey.KEY_LEFT:
39             alert("LEFT");
40             document.getElementById("hello").innerHTML = "LEFT";
41             break;
42         case tvKey.KEY_RIGHT:
43             alert("RIGHT");
44             document.getElementById("hello").innerHTML = "RIGHT";
45             break;
46         case tvKey.KEY_UP:
47             alert("UP");
48             document.getElementById("hello").innerHTML = "UP";
49             break;
50         case tvKey.KEY_DOWN:
51             alert("DOWN");
52             document.getElementById("hello").innerHTML = "DOWN";
53             break;
54         case tvKey.KEY_ENTER:
55         case tvKey.KEY_PANEL_ENTER:
56             alert("ENTER");
57             break;
58         default:
59             alert("Unhandled key");
60             break;
61     }
62 };
```

Hello TV App – main.css

```
1  *
2  {
3      padding: 0;
4      margin: 0;
5      border: 0;
6  }
7
8  /* Layout */
9  body
10 {
11     width: 960px;
12     height: 540px;
13
14     background-color: #ffffff;
15 }
16
17 #hello {
18     font-size: 150px;
19 }
20
```

Hello TV App – 결과화면

Key Event 처리

Key Code

[URL]
http://samsungdforum.com/Guide/artooo46/index.html

Full- screen Application		Single-wide Application
KEY_VOL_UP	KEY_1	KEY_WHEELDOWN
KEY_VOL_DOWN	KEY_2	KEY_WHEELUP
KEY_MUTE	KEY_3	KEY_RED
KEY_TOOLS	KEY_4	KEY_GREEN
KEY_INFO	KEY_5	KEY_YELLOW
KEY_EMODE	KEY_6	KEY_BLUE
KEY_DMA	KEY_7	KEY_RW
KEY_MENU	KEY_8	KEY_PAUSE
KEY_SOURCE	KEY_9	KEY_FF
KEY_PRECH	KEY_0	KEY_PLAY
KEY_FAVCH	KEY_WHEELDOWN	KEY_STOP
KEY_CHLIST	KEY_WHEELUP	KEY_ENTER
KEY_DMA	KEY_RED	KEY_RETURN
KEY_TTX_MIX	KEY_GREEN	KEY_EXIT
KEY_GUIDE	KEY_YELLOW	
KEY_SUBTITLE	KEY_BLUE	
KEY_ASPECT	KEY_RW	
KEY_DOLBY_SRR	KEY_PAUSE	
KEY_MTS	KEY_FF	
KEY_PANEL_CH_UP	KEY_PLAY	
KEY_PANEL_CH_DOWN	KEY_STOP	
KEY_PANEL_VOL_UP	KEY_ENTER	
KEY_PANEL_VOL_DOWN	KEY_RETURN	
KEY_PANEL_ENTER	KEY_EXIT	
KEY_PANEL_SOURCE		
KEY_PANEL_MENU		

index.html, JS file

```html
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
        <title>HelloTV</title>

        <!-- TODO : Common API -->
        <script type="text/javascript" language="javascript" src="$MANAGER_WIDGET/Common/AP
        <script type="text/javascript" language="javascript" src="$MANAGER_WIDGET/Common/AP

        <!-- TODO : Javascript code -->
        <script language="javascript" type="text/javascript" src="app/javascript/Main.js"><

        <!-- TODO : Style sheets code -->
        <link rel="stylesheet" href="app/stylesheets/Main.css" type="text/css">

        <!-- TODO: Plugins -->

    </head>

    <body onload="Main.onLoad();" onunload="Main.onUnload();">

        <!-- Dummy anchor as focus for key events -->
        <a href="javascript:void(0);" id="anchor" onkeydown="Main.keyDown();"></a>

        <!-- TODO: your code here -->
        <label id="hello">Hello TV</label>
    </body>
</html>
```

index.html

```javascript
Main.keyDown = function()
{
    var keyCode = event.keyCode;
    alert("Key pressed: " + keyCode);

    switch(keyCode)
    {
        case tvKey.KEY_RETURN:
        case tvKey.KEY_PANEL_RETURN:
            alert("RETURN");
            widgetAPI.sendReturnEvent();
            break;
        case tvKey.KEY_LEFT:
            alert("LEFT");
            document.getElementById("hello").innerHTML = "LEFT";
            break;
        case tvKey.KEY_RIGHT:
            alert("RIGHT");
            document.getElementById("hello").innerHTML = "RIGHT";
            break;
        case tvKey.KEY_UP:
            alert("UP");
            document.getElementById("hello").innerHTML = "UP";
            break;
        case tvKey.KEY_DOWN:
            alert("DOWN");
            document.getElementById("hello").innerHTML = "DOWN";
            break;
        case tvKey.KEY_ENTER:
        case tvKey.KEY_PANEL_ENTER:
            alert("ENTER");
            break;
        default:
            alert("Unhandled key");
            break;
    }
};
```

JS file

Key 등록/ 해제

```
var WIDGET = new Common.API.Widget();        // For sendReadyEvent()
var TVKEY  = new Common.API.TVKeyValue();    // Remote controller key value object
var PLUGIN = new Common.API.Plugin();        // Plugin common module

Main.onLoad = function () {
    window.onshow = function () {            // register the onshow event callback
        PLUGIN.registKey(TVKEY.KEY_VOL_UP);
        PLUGIN.registKey(TVKEY.KEY_VOL_DOWN);
    }

    WIDGET.sendReadyEvent();

    /**
     * code
     */
}
```

[등록]
registKey (KEY코드)
-해당 키를 App에서 직접 처리하겠다고 등록 하는 것

(ex : 볼륨 키 – TV의 볼륨 OSD 화면을 이용하지 않고 App에서 별도의 볼륨 UI를 제공하는 등 App에서 볼륨키를 직접 처리할 때)

```
var pluginAPI = new Common.API.Plugin();

Main.onLoad = function() {
    window.onShow = onShowEvent;
    widgetAPI.sendReadyEvent();
};

onShowEvent = function() {


    // 볼륨 OSD를 위한 키 등록 해지
    pluginAPI.unregistKey(tvKey.KEY_VOL_UP);
    pluginAPI.unregistKey(tvKey.KEY_VOL_DOWN);
    pluginAPI.unregistKey(tvKey.KEY_MUTE);
};
```

[해제]
unregistKey(KEY코드)
-해당 키를 App에서 처리하지 않고 TV에서 제공하는 처리를 그래도 따른다는 것

(ex : 볼륨 키 – TV의 볼륨 OSD 화면을 그대로 이용)

마우스 처리

```
<height>540</height>
<mouse>y</mouse>
<author>
```

config.xml : mouse 태그 추가

[주의]
Anchor 를 이용한 이벤트 처리 방식을 사용할 경우 마우스의 클릭 이벤트에 따라 클릭한 요소로 포커스가 이동하는 브라우저의 기본 특성 때문에 리모콘 Key 와 마우스 두 가지 방식은 함께 사용할 수 없음.

[리모콘과 마우스를 동시에 지원하기 위한 Tip!]
Anchor 대신 Body에 onkeydown 핸들러를 등록해서 처리한다.
Body에서 이벤트를 처리하기 때문에 포커스를 잃어버리는 문제가 해결됨.

```
<body onload="Main.onLoad();" onunload="Main.onUnload();"
onkeydown="Main.Keydown();">
```

Q & A

감사합니다.