

---

# **Interactive Mobile Device App API**

---

**Version 1.00**

**Samsung Smart TV**

# Interactive Mobile Device App API

---

|   |           |
|---|-----------|
| <b>1. INTERACTIVE MOBILE DEVICE APP API INTRODUCTION.....</b>                   | <b>5</b>  |
| 1.1. WHAT IS INTERACTIVE MOBILE DEVICE APP API?.....                            | 5         |
| 1.2. ARCHITECTURE .....   | 6         |
| 1.3. VERSION POLICY .....   | 7         |
| <b>2. USE METHOD OF INTERACTIVE MOBILE DEVICE APP API.....</b>                  | <b>7</b>  |
| 2.1. SUBSCRIBING EVENTS .....   | 7         |
| 2.2. VARIABLE DECLARATION OF INTERACTIVEAPP API TYPE (JAVASCRIPT) .....         | 7         |
| 2.3. INTERACTIVE MOBILE DEVICE APP API OPERATION .....                          | 7         |
| <b>3. USE METHOD OF INTERACTIVE MOBILE DEVICE APP API .....</b>                 | <b>8</b>  |
| 3.1. HOW TO HANDLE XML METADATA RECEIVED FROM IPHONE.....                       | 8         |
| 3.2. CREATING XML METADATA FOR IPHONE FROM INTERACTIVE MOBILE DEVICE APPS ..... | 8         |
| <b>4. INTERACTIVE MOBILE DEVICE APP API LIST .....</b>                          | <b>8</b>  |
| 4.1. COMMON FUNCTION.....   | 8         |
| <b>5. INTERACTIVE MOBILE DEVICE APP API FUNCTIONS DETAIL .....</b>              | <b>9</b>  |
| 5.1. COMMON.....  | 9         |
| 5.1.1. <i>Function</i> .....  | 9         |
| 5.1.1.1. <i>_tigger_remote_input</i> .....                                      | 9         |
| 5.1.2. <i>Generation of XML MetaData</i> .....                                  | 9         |
| 5.1.2.1. <i>Function MakeSouceHeader</i> .....                                  | 10        |
| 5.1.2.2. <i>Function MakeSourceItem</i> .....                                   | 10        |
| 5.1.2.3. <i>Function MakeSourceFooter</i> .....                                 | 11        |
| 5.1.2.4. <i>Function MakeMenuHeader</i> .....                                   | 12        |
| 5.1.2.5. <i>Function MakeMenuItem</i> .....                                     | 13        |
| 5.1.2.6. <i>Function MakeMenuFooter</i> .....                                   | 13        |
| 5.1.2.7. <i>Function MakePopupMessageHeader</i> .....                           | 14        |
| 5.1.2.8. <i>Function MakePopupItem</i> .....                                    | 15        |
| 5.1.2.9. <i>Function MakePopupFooter</i> .....                                  | 15        |
| 5.1.2.10. <i>Function MakePopupOffItem</i> .....                                | 16        |
| <b>6. OPEN API .....</b>  | <b>17</b> |
| 6.1. PREDEFINE OPEN API.....  | 17        |
| 6.1.1. <i>connect</i> .....   | 17        |
| 6.1.2. <i>action</i> .....  | 17        |
| 6.1.3. <i>media</i> .....   | 18        |

## Interactive Mobile Device App API

---

|           |                                    |           |
|-----------|------------------------------------|-----------|
| 6.1.4.    | <i>mousepad</i> .....              | 19        |
| 6.1.4.1.  | Mobile predefined message .....    | 19        |
| 6.1.5.    | <i>Metadata Special char</i> ..... | 20        |
| <b>7.</b> | <b>XML STRUCTURE</b> .....         | <b>21</b> |
| 7.1.      | MENU XML.....                      | 21        |
| 7.1.1.    | <i>Menu Type</i> .....             | 22        |
| 7.1.2.    | <i>Menu Schema</i> .....           | 26        |
| 7.1.2.1.  | Schema Design View .....           | 26        |
| 7.1.2.2.  | Schema Documen .....               | 26        |
| 7.2.      | SOURCE XML.....                    | 28        |
| 7.2.1.    | <i>Source XML</i> .....            | 29        |
| 7.2.2.    | <i>Source Schema</i> .....         | 37        |
| 7.2.2.1.  | Schema Design View .....           | 37        |
| 7.2.2.2.  | Schema Document .....              | 38        |
| 7.3.      | MESSAGE XML .....                  | 40        |
| 7.3.1.    | <i>Message Schema</i> .....        | 41        |
| 7.3.1.1.  | Schema Design View .....           | 41        |
| 7.3.1.2.  | Schema Document .....              | 41        |
| <b>8.</b> | <b>XML EXAMPLE</b> .....           | <b>43</b> |
| 8.1.      | MENU XML.....                      | 43        |
| 8.2.      | SOURCE XML - TABLE.....            | 45        |
| 8.3.      | SOURCE XML – CUSTOM .....          | 47        |

## Preface

---

- **Document Object**

The document describes the introduction, use method and detail APIs about Interactive Mobile Device App provided by Samsung Smart TV products

- **Document Scope**

The document describes only Interactive Mobile Device App API provided by Samsung Smart TV products. The guideline documents like development or UI is cared by other documents

- **Document Subject**

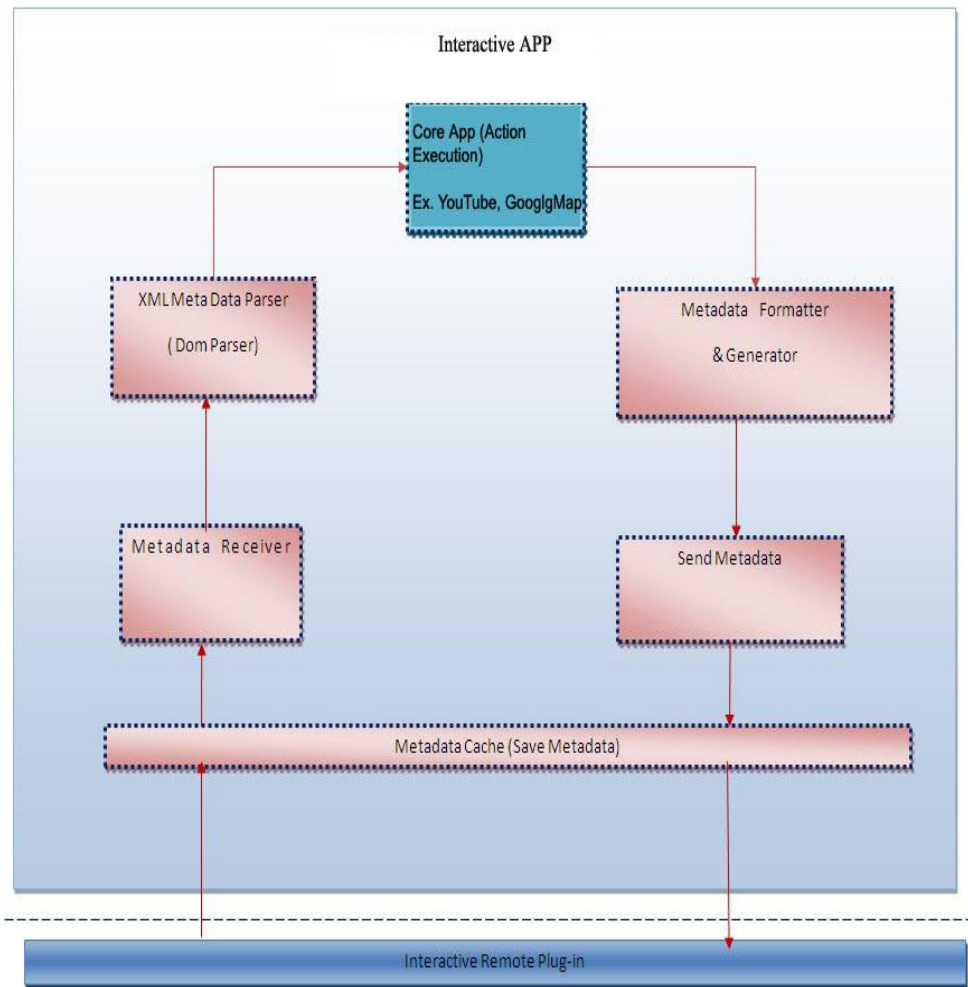
The document is subject to the developers who have the basic understanding about digital TV and javascript language

# 1. Interactive Mobile Device App API Introduction

## 1.1. What is Interactive Mobile Device App API?

- Interactive Mobile Device App API is defined as follows: API which application uses to control there actions from Interactive Remote like iPhone which is similar to remote control. API not only used to control the action on TV but also to draw the UI for iPhone for App Actions.
  - Interactive Mobile Device App API consists of Creation of XML and Extracting information from XMI reveided from iPhone.
  - Interactive Mobile Device App API consists of Controlling App operation through iPhone, similar to Remote Contol.
  
- Samsung Interacitve App API can be different for each product
  - Interacitve App API is dependent on Samsung native modules, and products can have different modules

## 1.2. Architecture



- App called Interactive Mobile Device App API to transfer its control on iPhone.
- Interactive Mobile Device App API layer behaves like the language binding between Javascript and XML.
- Communication between the iPhone and the App will happen by subscribing and registering the message events in App.
- These events transfer the information between iPhone and TV application in the form of XML data.
- As per the Diagram the communication between the iPhone and Samsung Smart TV is through XML Metadata. App generates and receives the Metadata from the iPhone.
- The Samsung Smart TV is composed of two main layers : App and Samsung Engine
  - App is made by web script languages including HTML, XML, CSS, and Javascript
  - Samsung Engine has many TV modules, and DeviceAPI is entering point to them by application.

### 1.3. Version Policy

- Each API in DeviceAPI can have the version as additional function, and can be managed with each version number
- Version description format : [InteractiveAppAPIName]-[xxxx] ex) InteractiveApp-V1.0

## 2. Use Method of Interactive Mobile Device App API

### 2.1. Subscribing Events

This event is used for receiving information from iPhone to TV App.

```
_g_tigger.pluginAppCommon.SubscribeEvent(_g_tigger.PL_APPCOMMON_MESSAGE_XML_INPUT);
```

This event is used for sending information from TV App to iPhone.

```
_g_tiger.pluginAppCommon.SendEvent_XML_Sync("xyz.meta");
```

xyz.meta file is used to send the xml metadata to Samsung smart phone.

### 2.2. Variable Declaration of InteractiveApp API Type (Javascript)

- Defined as Interactive Mobile Device App API variable

```
var _g_tigger = {  
    /* Define values for RemoteApp */  
    PL_APPCOMMON_MESSAGE_XML_INPUT : 46,  
    pluginAppCommon : null  
};
```

### 2.3. Interactive Mobile Device App API Operation

- Function call

```
_tigger_remote_input(String) :
```

### 3. Use method of Interactive Mobile Device App API

#### 3.1. How to Handle XML Metadata received from iPhone

Method Call:

`_tigger_remote_input (String STR) :`

#### 3.2. Creating XML MetaData for iPhone from Interactive Mobile Device Apps

Depending upon the action various metadata XML strings are to be created. These Meta data created are smart enough to create the UI on iPhone For Example:

### 4. Interactive Mobile Device App API List

#### 4.1. Common Function

| Common   |   |
|--|---|
| Common Function is the function which is used by the Most of the Apps. |   |
| <b>Latest Version</b>  | none  |
| <b>CLSID</b>   | none  |
| <b>Function</b>  | <ul style="list-style-type: none"><li>• <code>_tigger_remote_input</code></li></ul> |
| <b>Event</b>   | none  |
| <b>Enumeration</b>   | <ul style="list-style-type: none"><li>• String</li></ul>                            |

### 5. Interactive Mobile Device App API Functions Detail

#### 5.1. COMMON

##### 5.1.1. Function

###### 5.1.1.1. `_tigger_remote_input`

| <i>Function</i>  |   |
|--|---|
| <b><code>_tigger_remote_input</code></b>   |   |
| <code>_tigger_remote_input</code> used to handle the XML String received from iPhone |   |
| <b>Syntax</b>  | <code>_tigger_remote_input</code>   |
| <b>Parameter</b>   | String  |
| <b>Return Value</b>  | Null  |
| <b>Remarks</b>   | Depending upon the parameter received in xml string, perform the action on the application on tv. |
| <b>Emulator remarks</b>  | none  |
| <b>Example</b>   | <code>_tigger_remote_input("/0046/&lt;search&gt;Banglore&lt;/search&gt;")</code>                  |
| <b>Enumeration</b>   |   |
| <b>Version</b>   | none  |
| <b>Security Type</b>   | none  |

##### 5.1.2. Generation of XML MetaData

###### Function

Source Item is used for creation of Button, Text, Edit, Images, Images with Description , radio button, text combo box, password ,two button , checkbox, . For example how to draw the source item separate document (Tigger XML Guide.doc) has been released.

## Interactive Mobile Device App API

---

### 5.1.2.1. Function MakeSouceHeader

| <i>Function</i><br><i>MakeSouceHeader</i>                          |   |
|--|---|
| <i>MakeSouceHeader</i> used to create the header for source Header |   |
| <b>Syntax</b>  | <i>MakeSouceHeader</i>  |
| <b>Parameter</b>   | XMLBuffer, menuID<br>XMLBuuffer – any xml string<br>menuID - Current Selected Menu Id |
| <b>Return Value</b>  | XMLBuffer(XML String)   |
| <b>Remarks</b>   |   |
| <b>Emulator remarks</b>  | none  |
| <b>Example</b>   | none  |
| <b>Enumeration</b>   |   |
| <b>Version</b>   | none  |
| <b>Security Type</b>   | none  |
|  |   |

### 5.1.2.2. Function MakeSourceItem

| <i>Function</i><br><i>MakeSourceItem</i>   |   |
|--|---|
| MakeSourceItem used to create the Body for the responseXML of the Category for the Source Item to be displayed on Samsung Smart TV |   |
| <b>Syntax</b>  | <i>MakeSoureItem</i>  |
| <b>Parameter</b>   | XMLBuffer, type, typeDefault, identifier, title, description, image, onClick, onTrue, onFalse, multiTitles, multiURLs<br>type- desc , image,imagedesc,group,button,text,edit,password,picture<br>,checkbox,checkbox,combobox,radiobutton,twobutton,4digit |
| <b>Return Value</b>  | XMLBuffer(XML String)   |

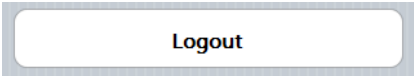
## Interactive Mobile Device App API

|                         |   |
|-------------------------|---|
| <b>Remarks</b>          | MakeSourceItem used to create the Body for the responseXML of the Category for the Source Item to be displayed on Samsung Smart TV. |
| <b>Emulator remarks</b> | none  |
| <b>Example</b>          | none  |
| <b>Enumeration</b>      |   |
| <b>Version</b>          | none  |
| <b>Security Type</b>    | none  |
|                         |   |

### 5.1.2.3. Function MakeSourceFooter

|   |  |
|---|--|
| <i>Function</i>   |  |
| <i>MakeSourceItem</i>   |  |
| MakeSourceFooter used to create the Footer for the responseXML of source Item |  |
| <b>Syntax</b>   | <i>MakeSoureItem</i>   |
| <b>Parameter</b>  | XMLBuffer  |
| <b>Return Value</b>   | XMLBuffer(XML String)  |
| <b>Remarks</b>  | To create the Footer for the response XML of the source Item |
| <b>Emulator remarks</b>   | none   |
| <b>Example</b>  | none   |
| <b>Enumeration</b>  |  |
| <b>Version</b>  | none   |
| <b>Security Type</b>  | none   |

For Example: The XML Input for creation of Button looks like :

|   |  |
|---|--|
| <pre>&lt;source &gt;   &lt;menu&gt;menuId&lt;/menu&gt;   &lt;sourceItem&gt;     &lt;type&gt;button&lt;/type&gt;</pre> |  |
|---|--|

## Interactive Mobile Device App API

---

|   |  |
|---|--|
| <pre>                 &lt;identifier&gt;button1&lt;/identifier&gt;                 &lt;title&gt;Logout&lt;/title&gt;                 &lt;onClick&gt;action?select=logout&lt;/onClick&gt;             &lt;/sourceItem&gt; &lt;/source&gt; </pre> |  |
|---|--|

### Function

Menu Item is used for creation of Menu on iPhone like table (creation of video list) , qwerty keyboard , custom menu item , search , media and mouse pad . For example how to draw the menu item separate document (Tigger XML Guide.doc) has been released.API Method created are described below.

#### 5.1.2.4. Function MakeMenuHeader

| <i>Function</i>   |   |
|---|---|
| <i>MakeMenuHeader</i>   |   |
| <i>MakeMenuHeader</i> used to create the header for Menu Header |   |
| <b>Syntax</b>   | <i>MakeMenuHeader</i>   |
| <b>Parameter</b>  | XMLBuffer<br>XMLBuffer – any xml string                         |
| <b>Return Value</b>   | XMLBuffer(XML String)   |
| <b>Remarks</b>  | <i>MakeMenuHeader</i> used to create the header for Menu Header |
| <b>Emulator remarks</b>   | none  |
| <b>Example</b>  | none  |
| <b>Enumeration</b>  |   |
| <b>Version</b>  | none  |
| <b>Security Type</b>  | none  |

## Interactive Mobile Device App API

---

### 5.1.2.5. Function MakeMenuItem

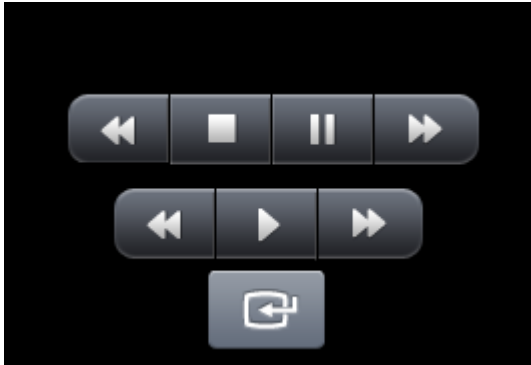
| <i>Function</i><br><i>MakeMenuItem</i>  |   |
|---|---|
| MakeSourceItem used to create the Body for the responseXML of the menu displayed on Samsung Smart TV. |   |
| <b>Syntax</b>   | <i>MakeMenuItem</i>   |
| <b>Parameter</b>  | XMLBuffer, identifier, type, title, name, icon, onClick, onEvent, onHistory<br>Type : table, qwerty keyboard, custom,search, media , mousepad |
| <b>Return Value</b>   | XMLBuffer(XML String)   |
| <b>Remarks</b>  |   |
| <b>Emulator remarks</b>   | none  |
| <b>Example</b>  | none  |
| <b>Enumeration</b>  |   |
| <b>Version</b>  | none  |
| <b>Security Type</b>  | none  |

### 5.1.2.6. Function MakeMenuFooter

| <i>Function</i><br><i>MakeMenuFooter</i>                                  |                       |
|---|-----------------------|
| MakeMenuFooter used to create the footer for the menu displayed on iPhone |                       |
| <b>Syntax</b>   | <i>MakeMenuFooter</i> |
| <b>Parameter</b>  | XMLBuffer(xml string) |
| <b>Return Value</b>   | XMLBuffer(XML String) |
| <b>Remarks</b>  |                       |
| <b>Emulator remarks</b>   | none                  |
| <b>Example</b>  | none                  |
| <b>Enumeration</b>  |                       |
| <b>Version</b>  | none                  |
| <b>Security Type</b>  | none                  |

## Interactive Mobile Device App API

For Example: The XML Input for creation of Media Menu item looks like :

|  |  |
|--|--|
| <pre> &lt;menu&gt; &lt;menuItem&gt;     &lt;identifier&gt;menuID6&lt;/identifier&gt;     &lt;type&gt;media&lt;/type&gt;     &lt;title&gt;Title F&lt;/title&gt;     &lt;name&gt;F&lt;/name&gt;     &lt;icon&gt;example/f.png&lt;/icon&gt;     &lt;onClick&gt;action?menu=f&lt;/onClick&gt; &lt;/menuItem&gt; &lt;/menu&gt; </pre> |  |
|--|--|

### Function

Message Item is used for creation of PopUp messages on the on the iPhone. These popup messages are displayed on TV and will be displayed on iPhone. For example how to draw the messages item separate document (Tigger XML Guide.doc) has been released.API Method created are described below.

#### 5.1.2.7. Function MakePopupMessageHeader

| <i>Function</i>   |  |
|---|--|
| <i>MakePopupMessageHeader</i>   |  |
| MakeSourceItem used to create the Body for the responseXML of the popup message displayed on iPhone |  |
| <b>Syntax</b>   | <i>MakePopupMessageHeader</i>                                  |
| <b>Parameter</b>  | XMLBuffer(XML String)  |
| <b>Return Value</b>   | XMLBuffer(XML String)  |
| <b>Remarks</b>  | MakePopupMessageHeader used to create the header for PopUP XML |
| <b>Emulator remarks</b>   | none   |
| <b>Example</b>  | none   |
| <b>Enumeration</b>  |  |
| <b>Version</b>  | none   |
| <b>Security Type</b>  | none   |

## Interactive Mobile Device App API

---

### 5.1.2.8. Function MakePopupItem

| <i>Function</i>         |   |
|-------------------------|---|
| <i>MakePopupItem</i>    |   |
| <b>MakePopupItem</b>    | used to create the body for PopUP XML               |
| <b>Syntax</b>           | <i>MakePopupMessageHeader</i>                       |
| <b>Parameter</b>        | XMLBuffer, title, descr, buttonNames, buttonURLs    |
| <b>Return Value</b>     | XMLBuffer(XML String)                               |
| <b>Remarks</b>          | MakePopupItem used to create the body for PopUP XML |
| <b>Emulator remarks</b> | none  |
| <b>Example</b>          | none  |
| <b>Enumeration</b>      |   |
| <b>Version</b>          | none  |
| <b>Security Type</b>    | none  |


### 5.1.2.9. Function MakePopupFooter

| <i>Function</i>         |   |
|-------------------------|---|
| <i>MakePopupFooter</i>  |   |
| <b>MakePopupFooter</b>  | used to create the Footer for the Popup message XML     |
| <b>Syntax</b>           | MakePopupFooter   |
| <b>Parameter</b>        | XMLBuffer,  |
| <b>Return Value</b>     | XMLBuffer(XML String)                                   |
| <b>Remarks</b>          | MakePopupFooter used to create the Footer for PopUP XML |
| <b>Emulator remarks</b> | none  |
| <b>Example</b>          | none  |
| <b>Enumeration</b>      |   |
| <b>Version</b>          | none  |
| <b>Security Type</b>    | none  |

## Interactive Mobile Device App API

---

For Example XML Creation for the POPUP message is shown below.

|   |  |
|---|--|
| <pre> &lt;message&gt;   &lt;title&gt;Title&lt;/title&gt;   &lt;text&gt;Message Description&lt;/text&gt;   &lt;buttonNames&gt;   &lt;name type="cancel"&gt;Cancel&lt;/name&gt;   &lt;name&gt;Ok&lt;/name&gt;   &lt;/buttonNames&gt;   &lt;buttonURLs&gt;   &lt;onClick&gt;action?message=cancel&lt;/onClick&gt;   &lt;onClick&gt;action?message=ok&lt;/onClick&gt;   &lt;/buttonURLs&gt; &lt;/message&gt; </pre> |  |
|---|--|

### 5.1.2.10. Function MakePopupOffItem

This Function is used to close the popup message displayed on iPhone.

| Function   |  |
|--|--|
| <i>MakePopupFooter</i>   |  |
| This Function is used to close the popup message displayed on iPhone |  |
| <b>Syntax</b>  | MakePopupOffItem   |
| <b>Parameter</b>   | XMLBuffer,   |
| <b>Return Value</b>  | XMLBuffer(XML String)  |
| <b>Remarks</b>   | This Function is used to close the popup message displayed on iPhone |
| <b>Emulator remarks</b>  | none   |
| <b>Example</b>   | none   |
| <b>Enumeration</b>   |  |
| <b>Version</b>   | none   |
| <b>Security Type</b>   | none   |

## 6. Open API

---

### 6.1. Predefine Open API

TV Interactive Server support HTTP protocol and Open API. All Open API use HTTP GET method.

※ All examples server ip pretend “192.168.0.1”.

#### 6.1.1. connect

This is Open API to get latest metadata after mobile connect to TV. “connect” api don’t reach to App.

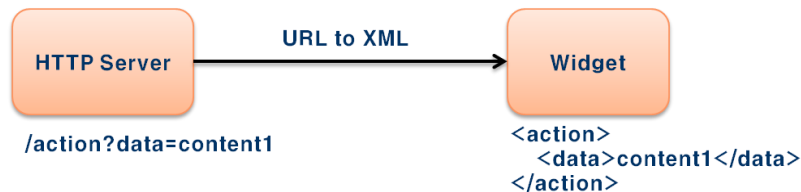
Example>

<http://192.168.0.1/connect>

#### 6.1.2. action

If user control UI, Mobile should send UI feedback to TV. “action” API is used to send UI feedback.

App receive XML data from “action” API. TV change from URL type to XML type.



Example>

<http://192.168.0.1/action?menu=1>

App receive

```
<action>
  <menu>1</menu>
</action>
```

<http://192.168.0.1/action?click=logout>

App receive

```
<action>
  <click>logout</click>
</action>
```

<http://192.168.0.1/action?select=1&range=all>

App receive

## Interactive Mobile Device App API

---

```
<action>
  <select>1</select>
  <range>all</range>
</action>
```

<http://192.168.0.1/action?menu=m1&switch=on&data=10>

App receive

```
<action>
  <menu>m1</menu>
  <switch>on</switch>
  <data>10</data>
</action>
```

### 6.1.3. media

“media” API support basic controllers to control movies. It’s used in “media” menu type. “media” menu type has controllers like Rew, Pause, FF, REC, Play, Stop. These controllers have a predefined url.

Rew: [media?button=rew](#)

Pause: [media?button=pause](#)

Stop: [media?button=stop](#)

FF: [media?button=ff](#)

Prev: [media?button=prev](#)

Play: [media?button=play](#)

Next: [media?button=next](#)

Enter : [media?button=enter](#)

Vol+ : [media?button=volup](#)

Vol- : [media?button=voldown](#)

App receive like

```
<media>
  <button>rew</button>
</media>
```

## Interactive Mobile Device App API

---

### 6.1.4. mousepad

“mousepad” API support to send mouse pointer to TV. App like google map need a mouse pointer. “mousepad” menu type use “mousepad” API.

Example>

<http://192.168.0.1/mousepad?x=30&y=40&width=320&height=400>

App receive like

```
<mousepad>
```

```
<x>30</x>
```

```
<y>40</y>
```

```
<width>320</width>
```

```
<height>400</height>
```

```
</mousepad>
```

x : touch x point

y : touch y point

width : width of touch area.

height : height of touch area.

#### 6.1.4.1. Mobile predefined message

If the user touch two finger on the mobile, two finger distance is 220pixel and touch area is 320x460, work “zoom function start” and mobile send “[mousepad?zoomstart=220&width=320&height=460](#)” message.

```
<mousepad>
```

```
<zoomstart>220</zoomstart>
```

```
<width>320</width>
```

```
<height>460</height>
```

```
</mousepad>
```

If the user move two finger on the mobile and distance is 180, mobile send “[mousepad?zoom=180&width=320&height=460](#)” message. This message will be sent every moving.

```
<mousepad>
```

```
<zoom>180</zoom>
```

```
<width>320</width>
```

```
<height>460</height>
```

```
</mousepad>
```

## Interactive Mobile Device App API

---

If the user untouch two finger on the mobile and two finger distance is 80pixel mobile send “mousepad?zoomend=80&width=320&height=460” message.

```
<mousepad>  
    <zoomend>80</zoomend>  
    <width>320</width>  
    <height>460</height>  
</mousepad>
```

App can be worked zoom in/out function with this 3 message.

App can use width and height for various mobile device resolution.

### 6.1.5. Metadata Special char

App developer must change special character.

If you want to use mousepad to <onEvent>, change ‘&’ character to ‘&amp;,’ and ‘<’ to ‘&lt;,’.

If you want to use mousepad <onEvent> tag

Mobile expect

<http://192.168.0.1/mousepad?x=30&y=40&width=320&height=400>

Metadata onEvent must be

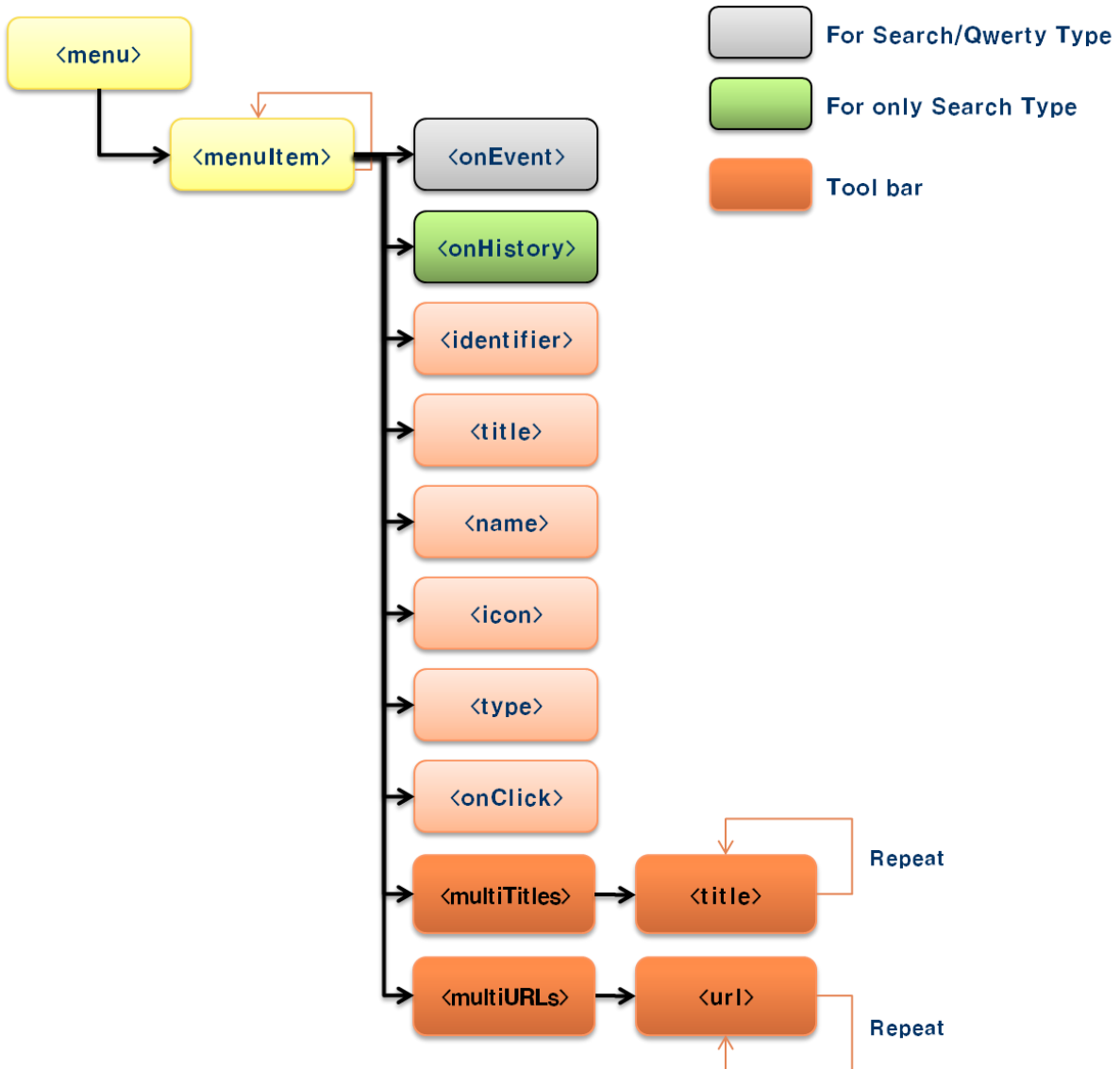
```
<onEvent>mousepad?x=[$mouse.x]&amp;y=[$mouse.y]&amp;width=[$mouse.width]&amp;height=[$mouse.height]  
</onEvent>
```

## 7. XML Structure

---

### 7.1. Menu XML

Menu XML use to transfer menu information of TV to Mobile.





## Interactive Mobile Device App API

| TAG           | Description  |
|---------------|--|
| <identifier>  | Unique value of current menu item  |
| <title>       | Title of top title bar   |
| <name>        | Title of bottom tab bar  |
| <icon>        | Icon Image URL (EX. "image/icon/icon1.png")<br>If a icon is not exist, do not use.   |
| <type>        | Type of current menu.  |
| <onClick>     | When user select a menu in bottom Tab bar, this value send to TV.<br>Ex) <onClick>action?menu=1</onClick>                          |
| <onEvent>     | The type is a search / qwerty, When type a string, this value(URL) is sent to TV<br>The type is a mousepad, send a mouse position. |
| <onHistory>   | <onHistory> tag is used only search type. If user touch bookmark button in search menu, mobile send this value.                    |
| <multiTitles> | For bottom tools bar. This tag have <title> tags.  |
| <multiURLs>   | For bottom tools bar. This tag have <url> tags.  |
| <title>       | For bottom tools bar. Each tools item's title. This title will be  |
| <url>         | For bottom tools bar. Each tools item's URL. If user touch tools bar item, this url will be sent.                                  |



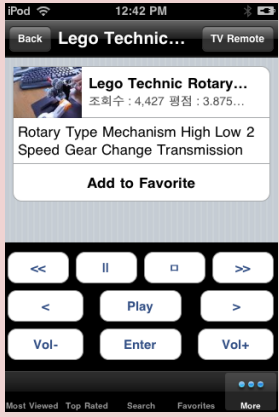
### 7.1.1. Menu Type

| <type> | Description      | Preview   | Available Source type                |
|--------|------------------|---|--------------------------------------|
| table  | Basic Table form |  | normal<br>desc<br>image<br>imagedesc |

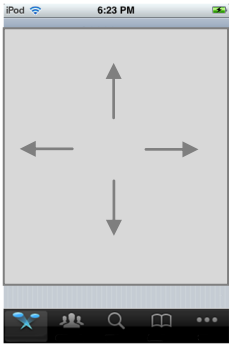
## Interactive Mobile Device App API

|                      |   |   |  |
|----------------------|---|---|--|
| <p><b>qwerty</b></p> | <p>Show mobile keyboard<br/>There is one edit box.</p>                            |    | <p>N/A</p> <p>No need source xml.</p> <p>URL that goes on TV is inputted<br/>&lt;onEvent&gt;.</p> <p>If the &lt;identifier&gt; is “qwerty1”,<br/>predefine script like<br/>[\$qwerty1.value\$] is inputted<br/>user typing keyboard string in<br/>&lt;onEvent&gt; tag.</p> <p>Example&gt;<br/>&lt;onEvent&gt;action?search=[\$qwerty1.value\$]&lt;/onEvent&gt;</p> |
| <p><b>custom</b></p> | <p>This type is made Variable UI<br/>control.</p>                                 |  | <p>normal<br/>desc<br/>image<br/>imagedesc<br/>group<br/>button<br/>text<br/>edit<br/>password<br/>image<br/>checkbox<br/>combobox<br/>radiobutton</p>   |
| <p><b>search</b></p> | <p>Top Title bar have a search<br/>controller.<br/>The result shows in table.</p> |   | <p>URL that goes on TV is inputted<br/>&lt;onEvent&gt;.</p> <p>If the &lt;identifier&gt; is “search1”,<br/>predefine script like<br/>[\$search1.value\$] is inputted<br/>user typing keyboard string in<br/>&lt;onEvent&gt; tag.</p>   |

# Interactive Mobile Device App API

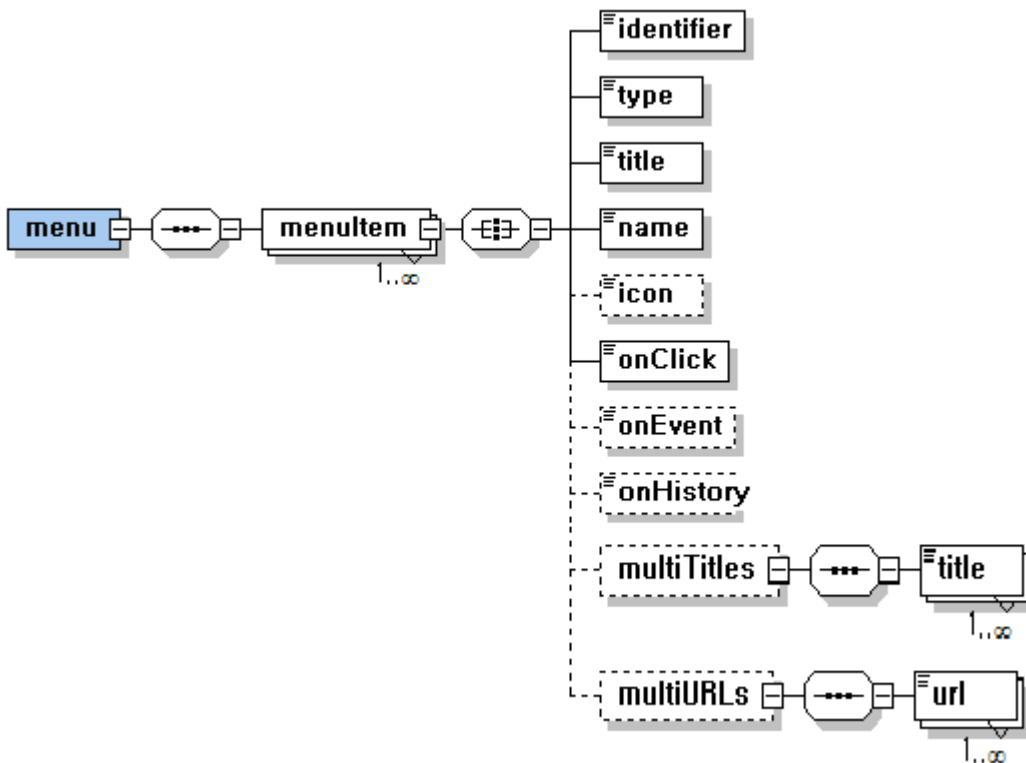
|                        |   |  |  |
|------------------------|---|--|--|
|                        |   |   | <p>Example&gt;</p> <pre>&lt;onEvent&gt;action?search=[\$search1.value\$]&lt;/onEvent&gt;</pre> <p>The search result will show in table.</p> <p>If user touch bookmark button(  ), mobile send <code>&lt;onHistory&gt;</code> tag to TV.</p>   |
| <p><b>media</b></p>    | <p>This type support Media buttons.</p>                   |  | <p>In Custom type area, it's same with Custom type.</p> <p>Each button's URL (Button URL is pre-define)</p> <p>Rew: <code>media?button=rew</code><br/>         Pause: <code>media?button=pause</code><br/>         Stop: <code>media?button=stop</code><br/>         FF: <code>media?button=ff</code><br/>         Prev: <code>media?button=prev</code><br/>         Play: <code>media?button=play</code><br/>         Next: <code>media?button=next</code><br/>         Enter : <code>media?button=enter</code><br/>         Vol+ : <code>media?button=volup</code><br/>         Vol- : <code>media?button=voldown</code></p> |
| <p><b>mousepad</b></p> | <p>When you need mouse pointer, this type is support.</p> |  | <p>No need source xml.</p> <p>In <code>&lt;onEvent&gt;</code> tag, URL is inputed like.</p> <pre>&lt;onEvent&gt;mousepad?x=[\$mouse.x\$]&amp;y=[\$mouse.y\$]&amp;width=[\$mouse.width\$]&amp;height=[\$mouse.height\$]&lt;/onEvent&gt;</pre> <p>Predefine script like <code>[\$mouse.x\$]</code> and <code>[\$mouse.y\$]</code> are Mobile device touch point.<br/> <code>[\$mouse.width\$]</code> and</p>   |

## Interactive Mobile Device App API

|                      |   |   |  |
|----------------------|---|---|--|
|                      |   |    | <p><code>[\$mouse.height\$]</code> are Mobile device touch area size.</p> <p>If the mobile work “zoom in” function, mobile send “<code>mousepad?zoom=in</code>” message.</p> <pre>&lt;mousepad&gt;     &lt;zoom&gt;in&lt;/zoom&gt; &lt;/mousepad&gt;</pre> <p>If the mobile work “zoom out” function, mobile send “<code>mousepad?zoom=out</code>” message.</p> <pre>&lt;mousepad&gt;     &lt;zoom&gt;out&lt;/zoom&gt; &lt;/mousepad&gt;</pre> |
| <p><b>remote</b></p> | <p>If you want normal Remote controller, use this type.</p> |  | <p>N/A</p> <p>This Remote Controller don't send Metadata. Working like general remote controller.</p>  |

## 7.1.2. Menu Schema

### 7.1.2.1. Schema Design View



### 7.1.2.2. Schema Document

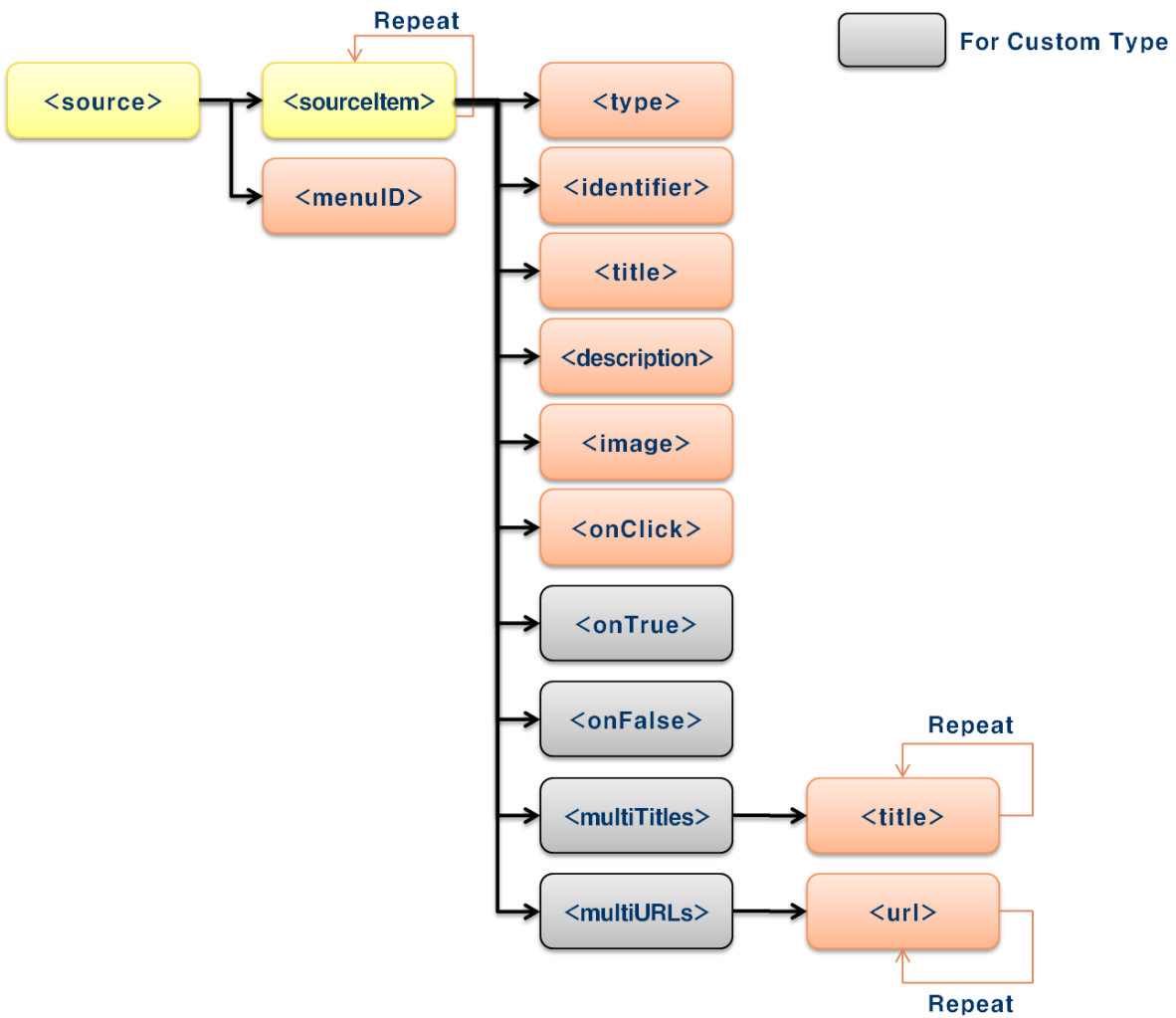
```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="menu">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="menuItem" maxOccurs="unbounded">
          <xs:complexType>
            <xs:all>
              <xs:element name="identifier" type="xs:string"/>
              <xs:element name="type">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration value="table"/>

```

```
        <xs:enumeration value="qwerty"/>
        <xs:enumeration value="custom"/>
        <xs:enumeration value="search"/>
        <xs:enumeration value="media"/>
        <xs:enumeration value="mousepad"/>
        <xs:enumeration value="remote"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="title" type="xs:string"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="icon" type="xs:anyURI" minOccurs="0"/>
<xs:element name="onClick" type="xs:anyURI"/>
<xs:element name="onEvent" type="xs:anyURI" minOccurs="0"/>
<xs:element name="onHistory" type="xs:anyURI" minOccurs="0"/>
<xs:element name="multiTitles" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="title" type="xs:string" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="multiURLs" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="url" type="xs:anyURI" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:all>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

## 7.2. Source XML

Metadata that TV content and information used to deliver mobile

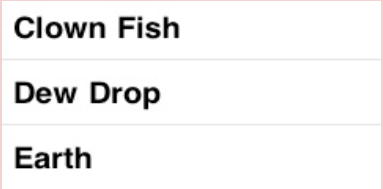


| TAG                       | Description                                    |
|---------------------------|--|
| <b>&lt;menuID&gt;</b>     | Current source item's Menu <identifier> value. |
| <b>&lt;sourceItem&gt;</b> | Show one source item.                          |
| <b>&lt;type&gt;</b>       | Item's Type.                                   |
| <b>&lt;identifier&gt;</b> | Item's Unique ID.                              |

## Interactive Mobile Device App API

|                            |  |
|----------------------------|--|
| <b>&lt;title&gt;</b>       | Item's title.  |
| <b>&lt;description&gt;</b> | Item's description.  |
| <b>&lt;image&gt;</b>       | Items' image URL.  |
| <b>&lt;onClick&gt;</b>     | When user touch item, this value is sent to TV.  |
| <b>&lt;onTrue&gt;</b>      | Used menu type is "custom" and source type is "checkbox".<br>URL is sent to TV, when chekcbox is "true".       |
| <b>&lt;onFalse&gt;</b>     | Used menu type is "custom" and source type is "checkbox".<br>URL is sent to TV, when chekcbox is "false".      |
| <b>&lt;multiTitles&gt;</b> | Used menu type is "custom" and source type are "checkbox", "radiobutton".<br>Title of option.                  |
| <b>&lt;multiURLs&gt;</b>   | Used menu type is "custom" and source type are "checkbox", "radiobutton".<br>URL of option that is sent to TV. |
| <b>&lt;title&gt;</b>       | Child Element of multiTitles, Input each option's title  |
| <b>&lt;url&gt;</b>         | Child Element of multiURLs, Input URL to send when user select each option.                                    |

### 7.2.1. Source XML

| Source Type   | Preview   | Description   |
|---|---|---|
| normal  |  | <p>Simply, just show title.</p> <p>If the item is touched, Send URL in &lt;onClick&gt; Tag.</p> |
| <pre> &lt;sourceItem&gt;   &lt;type&gt;normal&lt;/type&gt;   &lt;identifier&gt;item1&lt;/identifier&gt;   &lt;title&gt;Clown Fish&lt;/title&gt;   &lt;onClick&gt;action?select=1&lt;/onClick&gt; &lt;/sourceItem&gt; &lt;sourceItem&gt;   &lt;type&gt;normal&lt;/type&gt;   &lt;identifier&gt;item2&lt;/identifier&gt;   &lt;title&gt;Dew Drop&lt;/title&gt; </pre> |   |   |

## Interactive Mobile Device App API

```

        <onClick>action?select=2</onClick>
</sourceItem>
<sourceItem>
    <type>normal</type>
    <identifier>item3</identifier>
    <title>Earth</title>
    <onClick>action?select=3</onClick>
</sourceItem>

```

desc

|   |
|---|
| <b>Clown Fish</b><br>Clown fish in coral  |
| <b>Dew Drop</b><br>Dew drop on a leaf     |
| <b>Earth</b><br>Photo of earth from space |


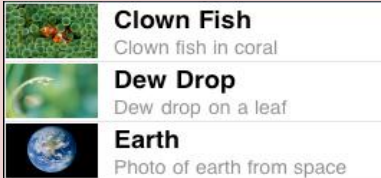
Add description from normal type.  
Input description in <description> tag.

```

<sourceItem>
    <type>desc</type>
    <identifier>item1</identifier>
    <title>Clown Fish</title>
    <description>Clown fish in coral</description>
    <onClick>action?select=1</onClick>
</sourceItem>
<sourceItem>
    <type>desc</type>
    <identifier>item2</identifier>
    <title>Dew Drop</title>
    <description>Dew drop on a leaf</description>
    <onClick>action?select=2</onClick>
</sourceItem>
<sourceItem>
    <type>desc</type>
    <identifier>item3</identifier>
    <title>Earth</title>
    <description>Photo of earth from space</description>
    <onClick>action?select=3</onClick>
</sourceItem>

```

## Interactive Mobile Device App API

|  |   |   |
|--|---|---|
| <p>image</p>   |    | <p>Add image from normal type.<br/>Input image URL in &lt;image&gt; tag.</p> <p>If you want to get image in TV, Input image URL like<br/>&lt;image&gt;Your App img path/img.png&lt;/image&gt;</p> <p>If you want to get image in Internet, Input image URL like<br/>&lt;image&gt;<a href="http://www.google.com/test.png">http://www.google.com/test.png</a>&lt;/image&gt;</p> <p>If you want to get image in TV, Input image URL like<br/>“http://” prefix is to get from internet, no “http://” prefix is to get from TV.</p> |
| <pre> &lt;sourceItem&gt;   &lt;type&gt;image&lt;/type&gt;   &lt;identifier&gt;item1&lt;/identifier&gt;   &lt;title&gt;Clown Fish&lt;/title&gt;   &lt;image&gt;image/icon1.png&lt;/image&gt;   &lt;onClick&gt;action?select=1&lt;/onClick&gt; &lt;/sourceItem&gt; &lt;sourceItem&gt;   &lt;type&gt; image &lt;/type&gt;   &lt;identifier&gt;item2&lt;/identifier&gt;   &lt;title&gt;Dew Drop&lt;/title&gt;   &lt;image&gt;image/icon2.png&lt;/image&gt;   &lt;onClick&gt;action?select=2&lt;/onClick&gt; &lt;/sourceItem&gt; &lt;sourceItem&gt;   &lt;type&gt; image &lt;/type&gt;   &lt;identifier&gt;item3&lt;/identifier&gt;   &lt;title&gt;Earth&lt;/title&gt;   &lt;image&gt;image/icon3.png&lt;/image&gt;   &lt;onClick&gt;action?select=3&lt;/onClick&gt; &lt;/sourceItem&gt; </pre> |   |   |
| <p>imagedesc</p>   |  | <p>Both desc and image type is implemented.<br/>Add &lt;description&gt; and &lt;image&gt; Tag in normal type.</p>   |

## Interactive Mobile Device App API

```

<sourceItem>
    <type>imagedesc</type>
    <identifier>item1</identifier>
    <title>Clown Fish</title>
    <description>Clown fish in coral</description>
    <image>image/icon1.png</image>
    <onClick>action?select=1</onClick>
</sourceItem>
<sourceItem>
    <type> imagedesc</type>
    <identifier>item2</identifier>
    <title>Dew Drop</title>
    <description>Dew drop on a leaf</description>
    <image>image/icon2.png</image>
    <onClick>action?select=2</onClick>
</sourceItem>
<sourceItem>
    <type> imagedesc</type>
    <identifier>item3</identifier>
    <title>Earth</title>
    <description>Photo of earth from space</description>
    <image>image/icon3.png</image>
    <onClick>action?select=3</onClick>
</sourceItem>

```

**group**


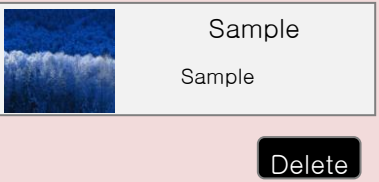

Separate table with group type.  
(Preview use TV Region and Demo group)

```

<sourceItem>
    <type>group</type>
    <identifier>group1</identifier>
    <title>TV Region</title>
</sourceItem>
...
<sourceItem>

```

# Interactive Mobile Device App API

|  |   |  |
|--|---|--|
| <pre> &lt;type&gt;group&lt;/type&gt; &lt;identifier&gt;group2&lt;/identifier&gt; &lt;title&gt;Demo&lt;/title&gt; &lt;/sourceItem&gt; </pre>  |   |  |
| <b>button</b>  |    | Generally, this type do button function  |
| <pre> &lt;sourceItem&gt; &lt;type&gt;button&lt;/type&gt; &lt;identifier&gt;button1&lt;/identifier&gt; &lt;title&gt;Logout&lt;/title&gt; &lt;onClick&gt;action?select=logout&lt;/onClick&gt; &lt;/sourceItem&gt; </pre>   |   |  |
| <b>morebutton</b>  |   | <p>This is extension version of “imagedesc” type.</p> <p>“morebutton” has a one additional button.</p> <p>&lt;onTrue&gt; tag is a title of additional button and &lt;onFalse&gt; tag is a URL of additional button.</p> <p>If the user touch “Delete” button, mobile will send URL in &lt;onFalse&gt; tag to TV.</p>                                 |
| <pre> &lt;sourceItem&gt; &lt;type&gt;morebutton&lt;/type&gt; &lt;identifier&gt;button2&lt;/identifier&gt; &lt;title&gt;Sample&lt;/title&gt; &lt;description&gt;Sample Description&lt;/description&gt; &lt;image&gt;image/icon.png&lt;/image&gt; &lt;onClick&gt;action?select=1&lt;/onClick&gt; &lt;onTrue&gt;Delete&lt;/onTrue&gt; &lt;onFalse&gt;action?delete=1&lt;/onFalse&gt; &lt;/sourceItem&gt; </pre> |   |  |
| <b>twobutton</b>   |  | <p>“twobutton” has additional button.</p> <p>&lt;onTrue&gt; tag is a title of additional button and &lt;onFalse&gt; tag is a URL of additional button.</p> <p>If the user touch “Next” button, mobile will send URL in &lt;onFalse&gt; tag to TV.</p> <p>If the user touch “Previous” button, mobile will send URL in &lt;onClick&gt; tag to TV.</p> |

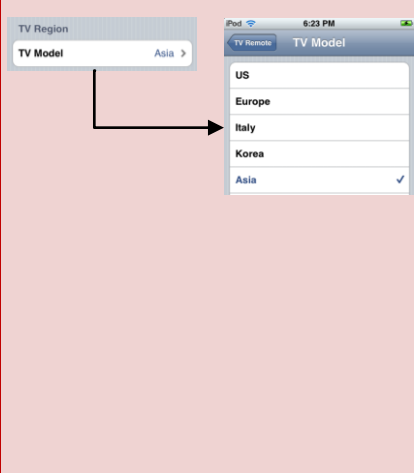
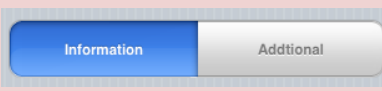
# Interactive Mobile Device App API

|   |   |   |
|---|---|---|
| <pre> &lt;sourceItem&gt;    &lt;type&gt;twobutton&lt;/type&gt;    &lt;identifier&gt;button3&lt;/identifier&gt;    &lt;title&gt;Previous&lt;/title&gt;    &lt;onClick&gt;action?mode=previous&lt;/onClick&gt;    &lt;onTrue&gt;Next&lt;/onTrue&gt;    &lt;onFalse&gt;action?mode=next&lt;/onFalse&gt;  &lt;/sourceItem&gt; </pre>  |   |   |
| <b>text</b>   |    | <p>The text control shows message or description.</p> <p>If you want to show title, use &lt;title&gt;, &lt;description&gt;</p> <p>Only want to show description, use &lt;description&gt;</p>  |
| <pre> &lt;sourceItem&gt;    &lt;type&gt;text&lt;/type&gt;    &lt;identifier&gt;text1&lt;/identifier&gt;    &lt;title&gt;Update&lt;/title&gt;    &lt;description&gt;October 29, 2009&lt;/description&gt;  &lt;/sourceItem&gt;  &lt;sourceItem&gt;    &lt;type&gt;text&lt;/type&gt;    &lt;identifier&gt;text2&lt;/identifier&gt;    &lt;description&gt;Join Save The Bay in standing up to the ...&lt;/description&gt;  &lt;/sourceItem&gt; </pre> |   |   |
| <b>edit<br/>password</b>  |  | <p>Edit control that user can input.</p> <p>If the type is "password", Input character shows like *.</p>  |
| <pre> &lt;sourceItem&gt;    &lt;type&gt;edit&lt;/type&gt;    &lt;identifier&gt;edit1&lt;/identifier&gt;    &lt;title&gt;Username&lt;/title&gt;    &lt;onClick&gt;action?username=[\${edit1.value}]&lt;/onClick&gt;  &lt;/sourceItem&gt;  &lt;sourceItem&gt;    &lt;type&gt;password&lt;/type&gt;    &lt;identifier&gt;edit2&lt;/identifier&gt;    &lt;title&gt;Password&lt;/title&gt; </pre>  |   | <p>※ If user input with keyboard, keyboard string must send to TV. But we use &lt;onClick&gt; tag for sending. So we need to combine &lt;onClick&gt; tag and keyboard string.</p> <p>If &lt;identifier&gt; is "edit1", input &lt;onClick&gt; tag like<br/> <a href="#">&lt;onClick&gt;action?username=[\${edit1.value}]&lt;/onClick&gt;</a></p> <p>[\${&lt;identifier&gt;.value}] is predefine script. Mobile automatically replace from [\${&lt;identifier&gt;.value}] to keyboard string.</p> |

## Interactive Mobile Device App API

|   |   |
|---|---|
| <pre>&lt;onClick&gt;action?username=[\$edit2.value\$]&lt;/onClick&gt; &lt;/sourceItem&gt;</pre>   |   |
| <p><b>4digit</b></p>  | <div data-bbox="323 434 719 510" style="border: 1px solid gray; padding: 5px; display: inline-block;"> <b>Password</b> <input style="width: 20px; height: 20px; border: 1px solid gray;" type="text"/> <input style="width: 20px; height: 20px; border: 1px solid gray;" type="text"/> <input style="width: 20px; height: 20px; border: 1px solid gray;" type="text"/> <input style="width: 20px; height: 20px; border: 1px solid gray;" type="text"/> </div> <p>Password control that user can input 4 numeric password<br/>This will be used in Samsung Smart TV Home like login.</p> |
| <pre>&lt;sourceItem&gt; &lt;type&gt;4digit&lt;/type&gt; &lt;identifier&gt;digitpass&lt;/identifier&gt; &lt;title&gt;Password&lt;/title&gt; &lt;onClick&gt;action?pass=[\$digitpass.value\$]&lt;/onClick&gt; &lt;/sourceItem&gt;</pre>   | <p>If the user send password “1234” to TV, Mobile will send password like “1”, “12”, “123”, “1234”.</p> <p>So if App receive 4 character from mobile, finish input password.</p>  |
| <p><b>picture</b></p>   | <div data-bbox="331 846 711 987" style="border: 1px solid gray; padding: 5px; display: inline-block;"> <p>Internet@TV Icon</p>  </div> <p>Control to show image.<br/>If you need title, use &lt;title&gt; tag.<br/>&lt;image&gt; tag is used as image URL.</p>   |
| <pre>&lt;sourceItem&gt; &lt;type&gt;picture&lt;/type&gt; &lt;identifier&gt;image1&lt;/identifier&gt; &lt;title&gt;Samsung Smart TV Icon&lt;/title&gt; &lt;image&gt;image/icon.png&lt;/image&gt; &lt;onClick&gt;action?select=image_internetTV&lt;/onClick&gt; &lt;/sourceItem&gt;</pre> |   |
| <p><b>checkbox</b></p>  | <div data-bbox="331 1525 711 1585" style="border: 1px solid gray; padding: 5px; display: inline-block;"> <p>JavaScript <input checked="" type="checkbox"/></p> </div> <p>Controller to select True/ False, On/off<br/>If the controller is true, &lt;onTrue&gt; URL is used.<br/>If the controller is false, &lt;onFalse&gt; URL is used.</p> <p>If the “default” attribute of &lt;type&gt; is true or false, the attribute value is default.<br/>Ex&gt;</p> <pre>&lt;type default=”true”&gt;checkbox&lt;/type&gt; &lt;type default=”false”&gt;checkbox&lt;/type&gt;</pre>              |
| <pre>&lt;sourceItem&gt; &lt;type default=”true”&gt;checkbox&lt;/type&gt; &lt;identifier&gt;checkbox1&lt;/identifier&gt; &lt;title&gt;JavaScript&lt;/title&gt; &lt;onTrue&gt;action?javascript=on&lt;/onTrue&gt;</pre>   |   |

# Interactive Mobile Device App API

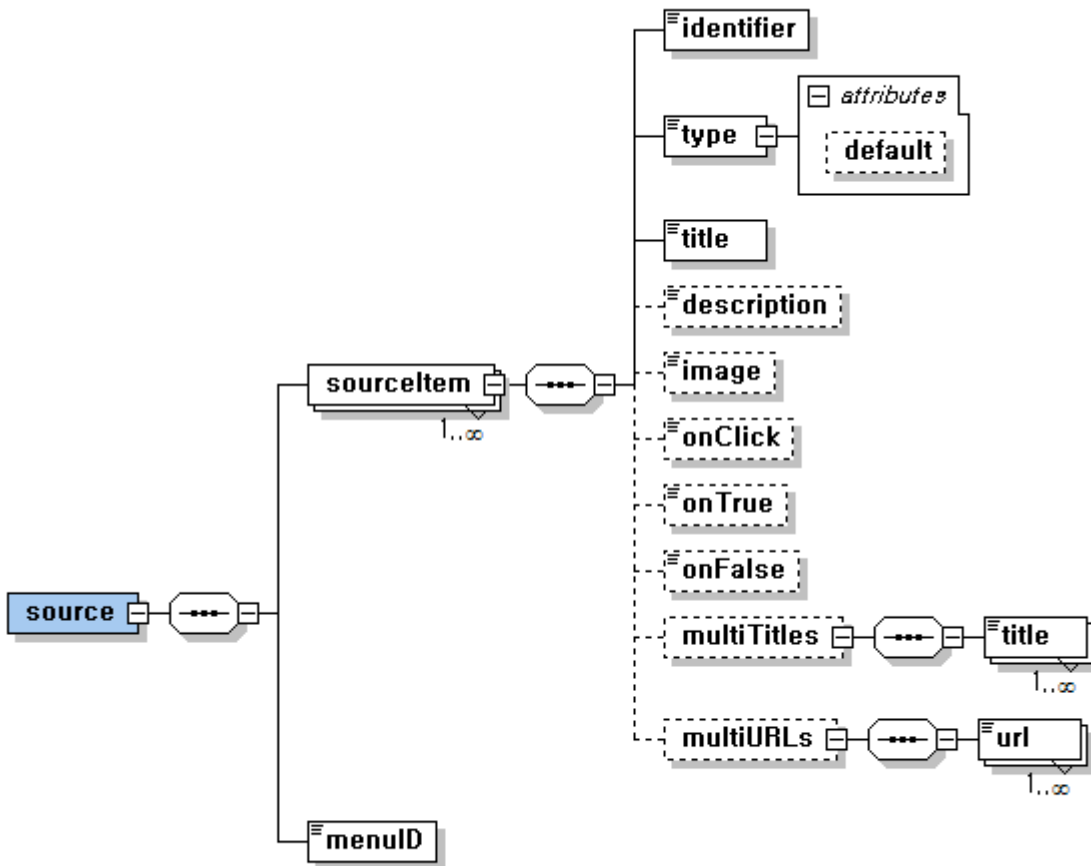
|   |   |   |
|---|---|---|
| <pre>&lt;onFalse&gt;action?javascript=off&lt;/onFalse&gt;</pre> |   |   |
| <pre>&lt;/sourceItem&gt;</pre>                                  |   |   |
| <b>combobox</b>   |    | <p>If many option is needed, use combobox.</p> <p>&lt;multiTitles&gt; have option's titles.</p> <p>&lt;multiURLs&gt; hava option's URLs.</p> <p>If a user select any option, &lt;url&gt; value of &lt;multiURLs&gt; tag is sent to TV.</p> <p>You can setting default value with "default" attribute of &lt;multiTitles&gt; tag. (start index from 0)</p> <p>If you want to set first item is default, set</p> <pre>&lt;multiTitles default="0"&gt;</pre> |
| <pre>&lt;sourceItem&gt;</pre>                                   |   |   |
| <pre>&lt;type default="4"&gt;combobox&lt;/type&gt;</pre>        |   |   |
| <pre>&lt;identifier&gt;combobox1&lt;/identifier&gt;</pre>       |   |   |
| <pre>&lt;title&gt;TV Model&lt;/title&gt;</pre>                  |   |   |
| <pre>&lt;multiTitles&gt;</pre>                                  |   |   |
| <pre>  &lt;title&gt;US&lt;/title&gt;</pre>                      |   |   |
| <pre>  &lt;title&gt;Europe&lt;/title&gt;</pre>                  |   |   |
| <pre>  &lt;title&gt;Italy&lt;/title&gt;</pre>                   |   |   |
| <pre>  &lt;title&gt;Korea&lt;/title&gt;</pre>                   |   |   |
| <pre>  &lt;title&gt;Asia&lt;/title&gt;</pre>                    |   |   |
| <pre>&lt;/multiTitles&gt;</pre>                                 |   |   |
| <pre>&lt;multiURLs&gt;</pre>                                    |   |   |
| <pre>  &lt;url&gt;action?model=us&lt;/url&gt;</pre>             |   |   |
| <pre>  &lt;url&gt;action?model=europe&lt;/url&gt;</pre>         |   |   |
| <pre>  &lt;url&gt;action?model=italy&lt;/url&gt;</pre>          |   |   |
| <pre>  &lt;url&gt;action?model=korea&lt;/url&gt;</pre>          |   |   |
| <pre>  &lt;url&gt;action?model=asia&lt;/url&gt;</pre>           |   |   |
| <pre>&lt;/multiURLs&gt;</pre>                                   |   |   |
| <pre>&lt;/sourceItem&gt;</pre>                                  |   |   |
| <b>radiobutton</b>  |  | <p>Radiobutton is similar to combobox. But radiobutton use less option than combobox.</p> <p>Do not use more than 4 option.</p>   |
| <pre>&lt;sourceItem&gt;</pre>                                   |   |   |
| <pre>&lt;type default="0"&gt;radiobutton&lt;/type&gt;</pre>     |   |   |

```

<identifier>radiobutton1</identifier>
<multiTitles>
  <title>Information</title>
  <title>Additional</title>
</multiTitles>
<multiURLs>
  <url>action?select=information</url>
  <url>action?select=additional_movie</url>
</multiURLs>
</sourceItem>
  
```

## 7.2.2. Source Schema

### 7.2.2.1. Schema Design View



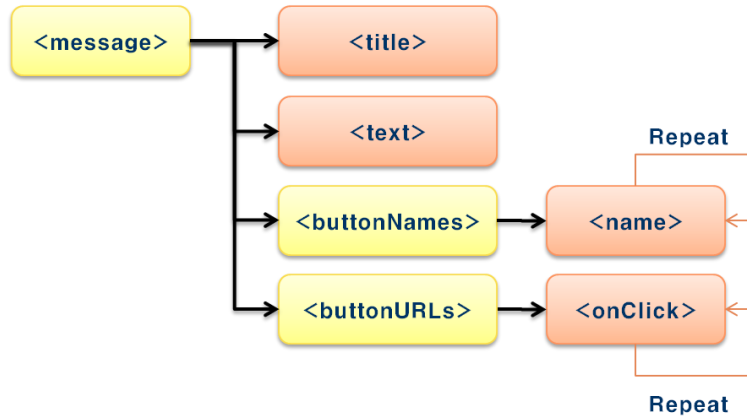
### 7.2.2.2. Schema Document

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="enumerationType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="normal"/>
      <xs:enumeration value="desc"/>
      <xs:enumeration value="image"/>
      <xs:enumeration value="imagedesc"/>
      <xs:enumeration value="group"/>
      <xs:enumeration value="button"/>
      <xs:enumeration value="text"/>
      <xs:enumeration value="edit"/>
      <xs:enumeration value="password"/>
      <xs:enumeration value="picture"/>
      <xs:enumeration value="checkbox"/>
      <xs:enumeration value="combobox"/>
      <xs:enumeration value="radiobutton"/>
      <xs:enumeration value="twobutton"/>
      <xs:enumeration value="4digit"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="source">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="sourceItem" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="identifier" type="xs:string"/>
              <xs:element name="type">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="enumerationType">
                      <xs:attribute name="default" type="xs:string" use="optional"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
        </xs:complexType>
    </xs:element>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="description" type="xs:string" minOccurs="0"/>
    <xs:element name="image" type="xs:anyURI" minOccurs="0"/>
    <xs:element name="onClick" type="xs:anyURI" minOccurs="0"/>
    <xs:element name="onTrue" type="xs:anyURI" minOccurs="0"/>
    <xs:element name="onFalse" type="xs:anyURI" minOccurs="0"/>
    <xs:element name="multiTitles" minOccurs="0">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="title" type="xs:string" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="multiURLs" minOccurs="0">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="url" type="xs:anyURI" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="menuID" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

### 7.3. Message XML

Message XML use to show TV error message or other message box in mobile.



| TAG  | Description   |
|--|---|
| <b>&lt;title&gt;</b>                                     | Title of message box  |
| <b>&lt;text&gt;</b>                                      | Description of message box  |
| <b>&lt;buttonNames&gt;</b>                               | Button names of message box.  |
| <b>&lt;buttonURLs&gt;</b>                                | URL of buttons.   |
| <b>&lt;name&gt;</b><br><b>&lt;name type="cancel"&gt;</b> | The name of each buttons in message box.<br>If attribute is type="cancel", the button is cancel button.<br>The other buttons don't need "type" attribute. |
| <b>&lt;onClick&gt;</b>                                   | If user select each button of message box, <onClick> URL is sent to TV.   |

| XML Data  | Preview |
|---|---------|
| <pre> &lt;message&gt;   &lt;title&gt;Title&lt;/title&gt;   &lt;text&gt;Message Description&lt;/text&gt;   &lt;buttonNames&gt;     &lt;name type="cancel"&gt;Cancel&lt;/name&gt;     &lt;name&gt;Ok&lt;/name&gt;   &lt;/buttonNames&gt;           </pre> |         |

## Interactive Mobile Device App API

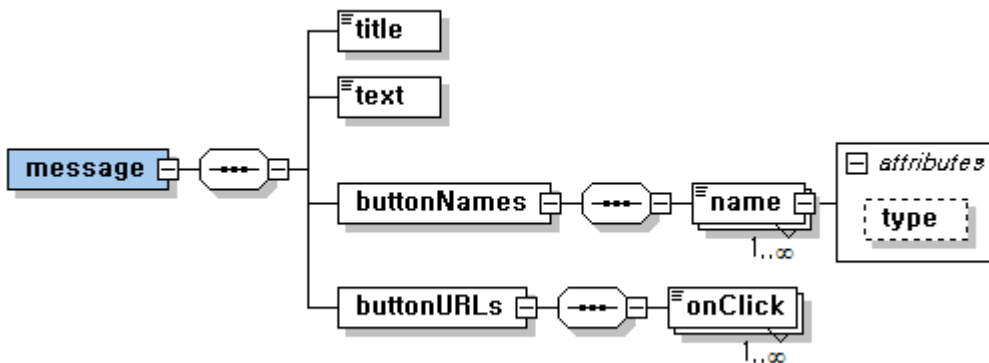
```
<buttonURLs>
  <onClick>action?message=cancel</onClick>
  <onClick>action?message=ok</onClick>
</buttonURLs>
</message>
```

※ If you want to close message box, send this metadata.

```
<message>
  <action>close</action>
</message>
```

### 7.3.1. Message Schema

#### 7.3.1.1. Schema Design View



#### 7.3.1.2. Schema Document

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="nameType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:element name="message">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="text" type="xs:string"/>
        <xs:element name="buttonNames">
          <xs:complexType>
```

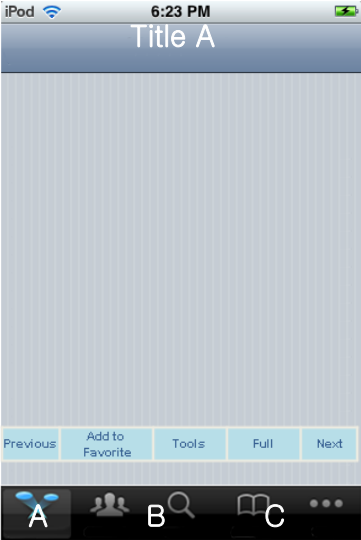
## Interactive Mobile Device App API

---

```
<xs:sequence>
  <xs:element name="name" maxOccurs="unbounded">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="nameType">
          <xs:attribute name="type" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="buttonURLs">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="onClick" type="xs:anyURI" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

## 8. XML Example

### 8.1. Menu XML

| XML Data   | Result  |
|--|---|
| <pre> &lt;menu&gt;   &lt;menuItem&gt;     &lt;identifier&gt;menuID1&lt;/identifier&gt;     &lt;type&gt;table&lt;/type&gt;     &lt;title&gt;Title A&lt;/title&gt;     &lt;name&gt;A&lt;/name&gt;     &lt;icon&gt;example/a.png&lt;/icon&gt;     &lt;onClick&gt;action?menu=a&lt;/onClick&gt;     &lt;multiTitles&gt;       &lt;title&gt;Previous&lt;/title&gt;       &lt;title&gt;Add to Favorite&lt;/title&gt;       &lt;title&gt;Tools&lt;/title&gt;       &lt;title&gt;Full&lt;/title&gt;       &lt;title&gt;Next&lt;/title&gt;     &lt;/multiTitles&gt;     &lt;multiURLs&gt;       &lt;url&gt;action?tool=previous&lt;/url&gt;       &lt;url&gt;action?tool=add&lt;/url&gt;       &lt;url&gt;action?tool=tools&lt;/url&gt;       &lt;url&gt;action?tool=full&lt;/url&gt;       &lt;url&gt;action?tool=next&lt;/url&gt;     &lt;/multiURLs&gt;   &lt;/menuItem&gt;   &lt;menuItem&gt;     &lt;identifier&gt;menuID2&lt;/identifier&gt;     &lt;type&gt;search&lt;/type&gt;     &lt;title&gt;Title B&lt;/title&gt;     &lt;name&gt;B&lt;/name&gt;     &lt;icon&gt;example/b.png&lt;/icon&gt;     &lt;onClick&gt;action?menu=b&lt;/onClick&gt; </pre> |  <p>Make menu A~F</p> <p>If make more than 6, from fifth menu to end menu are in etc menu table.</p> <p>Menu B(search) and Menu C(qwerty) send keyboard string to TV, &lt;onEvent&gt; tag use script [\${&lt;identifier&gt;.value\$} like “action?search=[\${menuID2.value\$}]”</p> <p>If keyboard string is “ABCD”, real URL to send is “action?search=ABCD”</p> <p>Then App receive this XML.</p> <pre> &lt;action&gt;   &lt;search&gt;ABCD&lt;/search&gt; &lt;/action&gt; </pre> <p>Menu E’s type is mousepad.</p> <p>The mousepad type can send mouse x position</p> |

|  |  |
|--|--|
| <pre> &lt;onEvent&gt;action?search=[\$menuID2.value\$]&lt;/onEvent&gt; &gt; &lt;/menuItem&gt; &lt;menuItem&gt;   &lt;identifier&gt;menuID3&lt;/identifier&gt;   &lt;type&gt;qwerty&lt;/type&gt;   &lt;title&gt;Title C&lt;/title&gt;   &lt;name&gt;C&lt;/name&gt;   &lt;icon&gt;example/c.png&lt;/icon&gt;   &lt;onClick&gt;action?menu=c&lt;/onClick&gt;   &lt;onEvent&gt;action?search=[\$menuID3.value\$]&lt;/onEvent&gt; &gt; &lt;/menuItem&gt; &lt;menuItem&gt;   &lt;identifier&gt;menuID4&lt;/identifier&gt;   &lt;type&gt;custom&lt;/type&gt;   &lt;title&gt;Title D&lt;/title&gt;   &lt;name&gt;D&lt;/name&gt;   &lt;icon&gt;example/d.png&lt;/icon&gt;   &lt;onClick&gt;action?menu=d&lt;/onClick&gt; &lt;/menuItem&gt; &lt;menuItem&gt;   &lt;identifier&gt;menuID5&lt;/identifier&gt;   &lt;type&gt;mousepad&lt;/type&gt;   &lt;title&gt;Title E&lt;/title&gt;   &lt;name&gt;E&lt;/name&gt;   &lt;icon&gt;example/e.png&lt;/icon&gt;   &lt;onClick&gt;action?menu=e&lt;/onClick&gt;   &lt;onEvent&gt;mousepad?x=[\$mouse.x\$]&amp;amp;y=[\$mouse.y\$]&amp;a mp;width=[\$mouse.width\$]&amp;amp;height=[\$mouse.height\$]&lt;/onEvent&gt; &lt;/menuItem&gt; &lt;menuItem&gt;   &lt;identifier&gt;menuID6&lt;/identifier&gt;   &lt;type&gt;media&lt;/type&gt;   &lt;title&gt;Title F&lt;/title&gt;   &lt;name&gt;F&lt;/name&gt; </pre> | <p>and mouse y position to TV with predefine script [\$mouse.x\$],[\$mouse.y\$]</p> <p>If &lt;onEvent&gt; is<br/> ”mousepad?x=[\$mouse.x\$]&amp;amp;y=[\$mouse.y\$]&amp;a<br/> mp;width=[\$mouse.width\$]&amp;amp;height=[\$mouse.h<br/> eight\$]”<br/> and mouse position is x=30, y=40, width=320,<br/> height=400 real URL to send is<br/> “mouse?x=30&amp;y=40&amp;width=320&amp;height=400”<br/> .</p> <p>Then App receive this XML.</p> <pre> &lt;mousepad&gt;   &lt;x&gt;30&lt;/x&gt;      &lt;y&gt;40&lt;/y&gt;   &lt;width&gt;320&lt;/width&gt; &lt;height&gt;400&lt;/height&gt; &lt;/mousepad&gt; </pre> |
|--|--|

```

        <icon>example/f.png</icon>
        <onClick>action?menu=f</onClick>
    </menuItem>
</menu>

```

## 8.2. Source XML - Table

| XML Data  | Result  |
|---|---|
| <pre> &lt;source&gt;   &lt;menuID&gt;menuID1&lt;/menuID&gt;   &lt;sourceItem&gt;     &lt;type&gt;imagedesc&lt;/type&gt;     &lt;identifier&gt;option&lt;/identifier&gt;     &lt;title&gt;Movie 1&lt;/title&gt;     &lt;description&gt;Movie          Description 1&lt;/description&gt;      &lt;image&gt;http://youtube.com/image/xxx_1.jpg&lt;/image&gt;     &lt;onClick&gt;action?select=1&lt;/onClick&gt;   &lt;/sourceItem&gt;   &lt;sourceItem&gt;     &lt;type&gt;imagedesc&lt;/type&gt;     &lt;identifier&gt;option&lt;/identifier&gt;     &lt;title&gt;Movie 2&lt;/title&gt;     &lt;description&gt;      Movie      Description 2&lt;/description&gt;      &lt;image&gt;http://youtube.com/image/xxx_2.jpg&lt;/image&gt;     &lt;onClick&gt;action?select=2&lt;/onClick&gt;   &lt;/sourceItem&gt;   &lt;sourceItem&gt;     &lt;type&gt;imagedesc&lt;/type&gt;     &lt;identifier&gt;option&lt;/identifier&gt;     &lt;title&gt;Movie 3&lt;/title&gt;     &lt;description&gt;      Movie      Description 3&lt;/description&gt; </pre> |  |

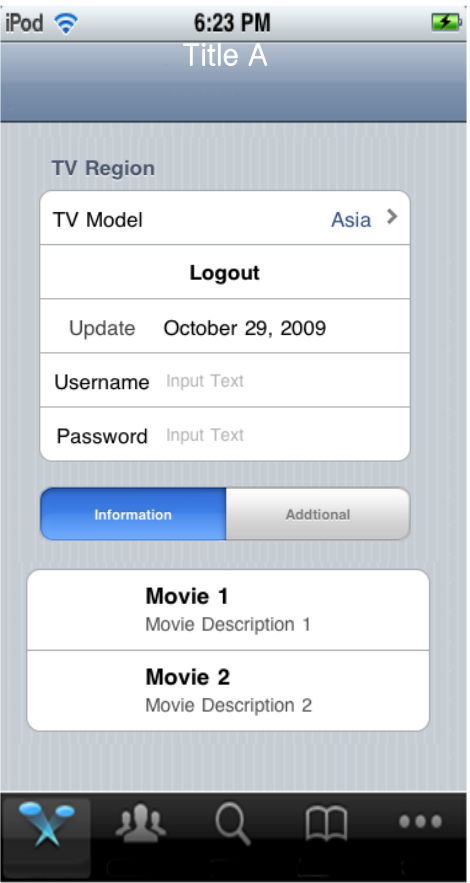
```
<image>http://youtube.com/image/xxx_3.jpg</image>
  <onClick>action?select=3</onClick>
</sourceItem>
<sourceItem>
  <type>imagedesc</type>
  <identifier>option</identifier>
  <title>Movie 4</title>
  <description>Movieescription4</description>

<image>http://youtube.com/image/xxx_4.jpg</image>
  <onClick>action?select=4</onClick>
</sourceItem>
<sourceItem>
  <type>imagedesc</type>
  <identifier>option</identifier>
  <title> Movie 5</title>
  <description>      Movie      Description
5</description>

<image>http://youtube.com/image/xxx_5.jpg</image>
  <onClick>action?select=5</onClick>
</sourceItem>
<sourceItem>
  <type>imagedesc</type>
  <identifier>option</identifier>
  <title>Movie 6</title>
  <description>      Movie      Description
6</description>

<image>http://youtube.com/image/xxx_6.jpg</image>
  <onClick>action?select=6</onClick>
</sourceItem>
</source>
```

### 8.3. Source XML – Custom

| XML Data   | Result  |
|--|---|
| <pre> &lt;source&gt;   &lt;menuID&gt;menuID1&lt;/menuID&gt;   &lt;sourceItem&gt;     &lt;type&gt;group&lt;/type&gt;     &lt;identifier&gt;group1&lt;/identifier&gt;     &lt;title&gt;TV Region&lt;/title&gt;   &lt;/sourceItem&gt;   &lt;sourceItem&gt;     &lt;type default="4"&gt;combobox&lt;/type&gt;     &lt;identifier&gt;combobox1&lt;/identifier&gt;     &lt;title&gt;TV Model&lt;/title&gt;     &lt;multiTitles&gt;       &lt;title&gt;US&lt;/title&gt;       &lt;title&gt;Europe&lt;/title&gt;       &lt;title&gt;Italy&lt;/title&gt;       &lt;title&gt;Korea&lt;/title&gt;       &lt;title&gt;Asia&lt;/title&gt;     &lt;/multiTitles&gt;     &lt;multiURLs&gt;       &lt;url&gt;action?model=us&lt;/url&gt;       &lt;url&gt;action?model=europe&lt;/url&gt;       &lt;url&gt;action?model=italy&lt;/url&gt;       &lt;url&gt;action?model=korea&lt;/url&gt;       &lt;url&gt;action?model=asia&lt;/url&gt;     &lt;/multiURLs&gt;   &lt;/sourceItem&gt;   &lt;sourceItem&gt;     &lt;type&gt;button&lt;/type&gt;     &lt;identifier&gt;button1&lt;/identifier&gt;     &lt;title&gt;Logout&lt;/title&gt;     &lt;onClick&gt;action?select=logout&lt;/onClick&gt;   &lt;/sourceItem&gt; &lt;/sourceItem&gt; </pre> |  <p>The screenshot shows an iPod touch interface. At the top, the status bar displays 'iPod', signal strength, Wi-Fi, and the time '6:23 PM'. Below the status bar is a blue header with the text 'Title A'. The main content area has a light gray background with a vertical gradient. It features a 'TV Region' section with a 'TV Model' dropdown menu currently set to 'Asia'. Below this is a 'Logout' button. Further down, there are two rows of information: 'Update October 29, 2009', 'Username Input Text', and 'Password Input Text'. At the bottom of this section are two buttons: 'Information' (highlighted in blue) and 'Additional'. Below these are two movie entries: 'Movie 1' with 'Movie Description 1' and 'Movie 2' with 'Movie Description 2'. The bottom dock contains icons for Home, People, Search, Bookmarks, and a menu icon.</p> |

```
<type>text</type>
<identifier>text1</identifier>
<title>Update</title>
<description>October 29, 2009</description>
</sourceItem>
<sourceItem>
  <type>edit</type>
  <identifier>edit1</identifier>
  <title>Username</title>

  <onClick>action?username=[$edit1.value$]</onClick>
</sourceItem>
<sourceItem>
  <type>password</type>
  <identifier>edit2</identifier>
  <title>Password</title>

  <onClick>action?username=[$edit2.value$]</onClick>
</sourceItem>
<sourceItem>
  <type default="0">radiobutton</type>
  <identifier>radiobutton1</identifier>
  <multiTitles>
    <title>Information</title>
    <title>Additional</title>
  </multiTitles>
  <multiURLs>
    <url>action?select=information</url>

    <url>action?select=additional_movie</url>
  </multiURLs>
</sourceItem>
<sourceItem>
  <type>imagedesc</type>
  <identifier>option</identifier>
  <title>Movie 1</title>
```

## Interactive Mobile Device App API

---

```
<description>Movie Description 1</description>

<image>http://youtube.com/image/xxx_1.jpg</image>
  <onClick>action?select=1</onClick>
</sourceItem>
<sourceItem>
  <type>imagedesc</type>
  <identifier>option</identifier>
  <title>Movie 2</title>
  <description>Movie Description 2</description>

  <image>http://youtube.com/image/xxx_2.jpg</image>
    <onClick>action?select=2</onClick>
  </sourceItem>
</source>
```