Archived:Example Sound Player

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

The article is believed to be still valid for the original topic scope.

The following example shows how to load sound files from the Internet and play them.

Note: This example searches for sound files from the loopmasters.com Web page owned and administrated by a 3rd party. If the HTML structure of this page changes, regex filters will not be able to match the content. In real world widgets using XML structures and interfaces are recommended.

Note 2. In some devices, you need to activate Alert sounds on your current profile to hear sounds played by Java applications.

sound_player.he

```
class
{
  //Thanks and Copyrights goes to LoopMasters
  const String URL = "http://www.loopmasters.com/WAV-Downloads.asp";

  const int CMD_BACK        = 1;
  const int CMD_PLAY        = 2;
  const int CMD_STOP        = 3;
  const int CMD_VOLUME_MENU = 4;
  const int CMD_VOL         = 100;

  MenuItem BACK    = new MenuItem(CMD_BACK, "Back");
  MenuItem PLAY    = new MenuItem(CMD_PLAY, "Play");
  MenuItem STOP    = new MenuItem(CMD_STOP, "Stop");
  MenuItem VOLUME  = new MenuItem(CMD_VOLUME_MENU, "Volume");

  Menu MENU = new Menu()
    .add(CMD_VOL+100, "100%")
    .add(CMD_VOL+80, "80%")
    .add(CMD_VOL+60, "60%")
    .add(CMD_VOL+40, "40%")
    .add(CMD_VOL+20, "20%")
    .add(CMD_VOL+10, "10%")
    .add(CMD_VOL+5, "5%");

  Flow flow = null;
  Shell shell = null;
  Player player = null;

  int thumbX = 50;
  int thumbY = 50;
  int volume = 100;


  // --- Widget callbacks ---


  void startWidget()
  {
    thumbX, thumbY = getScreenSize();
    thumbX = thumbX / 2;
    thumbY = thumbY / 2;

    setMinimizedView(createMinimizedView("viewMini", getStyle("default")));
  }


  Shell openWidget()
  {
    flow = createFlow();

    loadCdList();

    shell = new Shell(flow);

    return shell;
  }


  void closeWidget()
  {
    stopPlayer();
  }


  MenuItem getSoftKey(Shell shell, Component focused, int key)
  {
    if (key == SOFTKEY_BACK) {
      return BACK;

    } else if (key == SOFTKEY_MIDDLE) {
      return (player != null)? STOP : PLAY;

    } else if (key == SOFTKEY_OK) {
      return VOLUME;
    }
    return null;
  }


  void actionPerformed(Shell shell, Component source, int action)
```


(C) Copyright Nokia 2013. All rights reserved.

```
{
  if (action == CMD_BACK) {
    popShell(shell);

  } else if (action == CMD_STOP) {
    stopPlayer();

  } else if (action == CMD_PLAY) {
    stopPlayer();
    loadSound(String(source.getData()));

  } else if (action == CMD_VOLUME_MENU) {
    MENU.open(shell);

  } else if (action > CMD_VOL) {
    //volume % is the action id minus CMD_VOL
    volume = action-CMD_VOL;
    setBubble(null, "Volume "+volume+"%");
  }
}


// --- UI functions ---

void showCds(final Value cds)
{
  flow.clear();
  int i = 0;
  foreach (Value cd : cds) {
    flow.add(createCdTable(cd));
    if (++i > 5) break; //lets show only few
  }
}


Flow createCdTable(final Value value)
{
  Flow flow = createFlow();
  flow.add(createHeader(cleanCdName(String(value.name))));

  String pictUrl = normalizeUrl(URL, String(value.image).trim());
  Picture pict = getPicture(pictUrl, thumbX, thumbY, "jpeg");
  pict.setStyle(getStyle("image"));
  pict.setPreferredWidth(-100);
  pict.setFlags(VISIBLE|LINEFEED);
  flow.add(pict);

  foreach (Value sound : value.sounds) {
    String path = String(sound);
    String url = normalizeUrl(URL, encodeURL(path.trim()));
    url = url.replace(' ', '+');
    String file = getFilename(path);
    if (file.length() > 0) {
      flow.add(createLink(url, file));
    }
  }

  return flow;
}


Flow createFlow()
{
  Flow flow = new Flow(getStyle("flow"));
  flow.setPreferredWidth(-100);
  flow.setFlags(VISIBLE|LINEFEED);
  return flow;
}


Label createHeader(final String text)
{
  Label label = new Label(getStyle("header"), text);
  label.setPreferredWidth(-100);
  label.setFlags(VISIBLE|LINEFEED);
  return label;
}


Label createLink(final String url, final String name)
{
  Label label = new Label(getStyle("link"), name);
  label.setPreferredWidth(-100);
  label.setFlags(VISIBLE|WRAP|FOCUSABLE);
  label.setAction(CMD_PLAY);
  label.setData(url);
  return label;
}


// --- Loading and playing sounds ---


void loadCdList()
{
  if (getStore().has("list")) {
    printf("using cached list");
    showCds(getStore().getValue("list"));
    return;
  }

  call(null, "sounds" , "get", ["url" => URL], ok, nok);
  Prompt prompt = new Prompt(null, "Loading sounds...", null, null);
  prompt.push();

  void ok(Object state, Value ret)
  {
    if (prompt != null) {
      prompt.pop();
      prompt = null;
    }

    if (ret.size() > 0) {
      getStore().put("list", ret);
      showCds(ret);

    } else {
      setBubble(null, "No Cds found ;(");
    }
  }
```


(C) Copyright Nokia 2013. All rights reserved.

```
    void nok(Object state, String error)
    {
      printf("There was a problem: "+error);
      if (prompt != null) {
        prompt.pop();
        prompt = null;
      }
    }
  }


  void loadSound(final String url)
  {
    if (getStore().has(url)) {
      printf("using cached: "+url);
      play(ByteArray(getStore().getValue(url)));
      return;
    }
    printf("loading: "+url);

    call(null, "http" , "get", ["url" => url], ok, nok);
    Prompt prompt = new Prompt(null, "Loading sound...", null, null);
    prompt.push();

    void ok(Object state, Value ret)
    {
      if (prompt != null) {
        prompt.pop();
        prompt = null;
      }

      getStore().put(url, ret);
      play(ByteArray(ret));
    }

    void nok(Object state, String error)
    {
      setBubble(null, "Loading failed");
      if (prompt != null) {
        prompt.pop();
        prompt = null;
      }
    }

    void play(ByteArray soundData)
    {
      player = getPlayer(ByteArray(soundData), "audio/mpeg", volume);
      player.start();
      shell.updateMenu();
    }
  }


  void stopPlayer()
  {
    if (player != null) {
      player.stop();
      player.close();
      player = null;
    }
    shell.updateMenu();
  }


  // --- Helper functions ---


  String normalizeUrl(final String base, final String url)
  {
    if (url.startsWith("http")) {
      return url;
    }

    final int i = base.lastIndexOf('/', base.length());
    if (i == -1) return null;
    final String baseDir = base.substring(0, i+1);

    if (url.startsWith("/")) {
      return baseDir + url.substring(1, url.length());

    } else {
      return baseDir + url;
    }
  }


  String getFilename(String file)
  {
    final int i = file.lastIndexOf('/', file.length());
    if (i == -1) return file;

    file = file.substring(i+1, file.length());

    if (file.toLowerCase().endsWith(".mp3")) {
      return file.substring(0, file.length()-4);
    }
    return file;
  }


  String cleanCdName(final String name)
  {
    int i = name.indexOf("-", 0);
    if (i < 0) {
      i = name.indexOf("Free", 0);
    }
    if (i < 0) {
      i = name.indexOf("Preview", 0);
    }

    if (i > 0) {
      return name.substring(0, i).trim();
    }
    return name;
  }

} //class
```


(C) Copyright Nokia 2013. All rights reserved.

widget.xml

```
<?xml version="1.0" encoding="utf-8"?>

<widget spec_version="2.0">
  <info>
    <name>example_soundplayer</name>
    <version>1.0</version>
    <author>example</author>
    <clientversion>1.0</clientversion>
    <shortdescription>Sound Player Example</shortdescription>
    <longdescription>Sound Player Example</longdescription>
    <tags>example soundplayer</tags>
  </info>

  <parameters>
    <parameter name="widgetname">Sound Player</parameter>
  </parameters>

  <services>
    <service type="http" id="http"/>
    <service type="http" id="sounds">
      <filter id="sounds_filter"/>
    </service>
  </services>

  <filters>
   <filter id="sounds_filter">
     <list>
       <foreach>
          <regex>
           <![CDATA[
             (?si)<table width="100%".*?>(.*?)</table>
           ]]>
         </regex>

          <list>
            <var id="cd">
              <regex_match group="1"/>
            </var>

            <get id="cd"/>
            <regex>
             <![CDATA[
               (?si)<td colspan="7" class="style3">(.*?)</td>
             ]]>
           </regex>
           <item name="name">
              <regex_match group="1"/>
            </item>

            <get id="cd"/>
            <regex>
             <![CDATA[
               (?si)<img src="(.*?)"
             ]]>
           </regex>
           <item name="image">
              <regex_match group="1"/>
            </item>

            <list name="sounds">
              <get id="cd"/>
              <foreach>
                <regex>
                 <![CDATA[
                   (?si)<a href="(.{5,50}?)">MP3</a>
                 ]]>
               </regex>
               <item>
                  <regex_match group="1"/>
                </item>
              </foreach>
            </list>
          </list>
        </foreach>
      </list>
    </filter>
  </filters>


  <resources>
    <code src="sound_player.he"/>

    <stylesheet>
      mini {
        background: solid white;
        color: black;
        align: vcenter hcenter;
      }

      maxi {
        background: solid white;
        padding: 5 5 5 5;
      }

      flow {
        background: solid white;
      }

      header {
        background: solid #FFCC00;
        align: vcenter hcenter;
      }

      .image {
        align: vcenter hcenter;
      }

      link {
        color: blue;
        font: small plain;
        padding: 0 0 0 5sp;

        focused {
          font: small bold;
          border: 0 0 1 0;
```


(C) Copyright Nokia 2013. All rights reserved.

```
          border-type: rectangle blue;
        }
      }
    </stylesheet>
  </resources>

  <layout minimizedheight="2em">
    <view id="viewMini" class="mini">
      <label class="mini">${widgetname}</label>
    </view>
  </layout>

</widget>
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.