



Contenu

Et maintenant ? Musique maestro !	2
Lui tirer les oreilles	8
Regarde, mon ami lapin, je suis publié !	12
Le fichier descriptor.xml.....	13
Le fichier screen.xml	14
Dernière ligne droite (jusqu'à la prochaine ;))	18



karotz - tutoriel pour débutant sous Windows -

Et maintenant ? Musique maestro !

Plus j'avance dans ces tutoriels et plus je me dis qu'il y a encore mille choses à voir et pourtant il va bien falloir s'arrêter un jour que vous puissiez voler de vos propres ailes.

J'avais décidé de laisser de côté la gestion multimédia « musique, lecture de titres de plages », j'avoue que je ne suis pas du tout musique et puis un appel lancé sur le forum officiel en a décidé autrement.

Je sais que des applications sont dans les cartons, toutes plus belles les unes que les autres qui permettront de commander le player à la voix, avec les oreilles, de choisir par titre, par auteur, par genre que sais-je encore ?

Vous vous souvenez de ma devise ? Pourquoi faire compliqué lorsqu'on peut faire simple ? En plus la demande sur le forum est aussi simple que cela : « je veux lire une playlist à une heure donnée par déclencheur, une autre playlist à un autre moment de la journée ». Point !

Alors je me suis dit que c'était un beau sujet pour ce tutoriel.

Qu'allons-nous faire ?

- 1 – lire de la musique qui se trouve sur une clé USB branchée à l'arrière du lapin
- 2 – Cette clé contiendra de multiples musiques et des « playlists»
- 3 – Nous lirons une playlist et une seule par déclenchement (les profils de l'application permettant de lire différentes playlists à des moments différents
- 4 – Nous pourrons mettre en pause (pour répondre au téléphone par exemple) et reprendre là où nous nous sommes arrêtés.
- 5 – Nous mettrons notre application en ligne (en privé inutile qu'il y ait 300 applications playlist ;)) ce qui permettra d'une part d'apprendre à publier des applications mais aussi de la paramétrer pour qu'elle se déclenche avec différentes clés et / ou à différentes heures

Ce que nous ne ferons pas (mais vous aurez tout loisir pour essayer de votre côté) c'est de faire de la lecture aléatoire, pouvoir sauter un titre qui est lu, revenir en arrière, demander un titre en particulier, un auteur etc.

Si vous vous rendez sur notre site favori à l'instruction « multimedia » <http://dev.karotz.com/sdk/#karotz-multimedia> vous vous rendez compte que les options sont multiples.

Mais car il y a un mais elles ne sont pas faciles à mettre en œuvre.

Avant de mettre les mains dans le cambouis où plus exactement de commencer à coder il me paraît sympa de faire quelques rappels sur la musique telle qu'elle est lue aujourd'hui par tous nos appareils.

Le lapin sait lire la majorité des formats reconnus à ce jour, mp3, ogg, flac,wma,mpc...

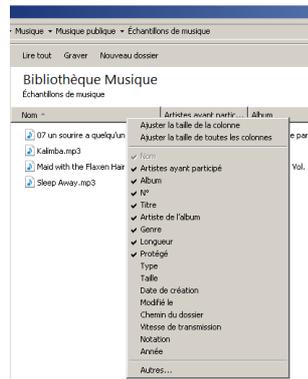
karotz - tutoriel pour débutant sous Windows -



Lorsque vous affichez un répertoire de votre disque contenant des fichiers avec ces extensions il affiche normalement des colonnes supplémentaires comme « album », « auteur », « titre » etc. Comme ceci :

Nom	Artistes ayant partic...	Album	N°	Titre	Artiste de l'album	Genre	Longueur	Protégé
07 un sourire à quelqu'un.mp3	Carine Erseng	la maison de parapluies	7	un sourire à quelqu'un	Carine Erseng	Alternatif	00:03:45	Non
Kalimba.mp3	Mr. Scruff	Ninja Tuna	1	Kalimba	Mr. Scruff	Electronic	00:05:48	Non
Maid with the Flaxen Hair.mp3	Richard Stoltzman; ...	Fine Music, Vol. 1	2	Maid with the Flaxen Hair	Richard Stoltzman	Classical	00:02:49	Non
Sleep Away.mp3	Bob Acri	Bob Acri	3	Sleep Away	Bob Acri	Jazz	00:03:20	Non

Si vous n'avez pas toutes ces informations un clic droit sur la barre des titres vous permet de choisir les colonnes à afficher



Par ailleurs vous savez combien notre sympathique lapin a horreur des caractères accentués et autres cédille et caractères spéciaux.

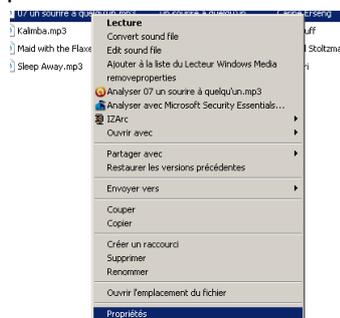
Regardez ici en rouge :

Nom	Titre
07 un sourire à quelqu'un.mp3	un sourire à quelqu'un

S'il vous prenez l'envie de faire lire ce titre à votre lapin, il ne le lira pas, pire c'est toute l'application qui ne fonctionnera pas ! Pensez-y si ça ne fonctionne pas.

Comme corriger cela ?

Pour le nom ça n'est pas compliqué, F2 sur le nom et on change à par a, pour le titre il faut faire clic droit sur le nom puis propriétés



Puis onglet détail et on corrige l'information qui nous pose problème

karotz - tutoriel pour débutant sous Windows -



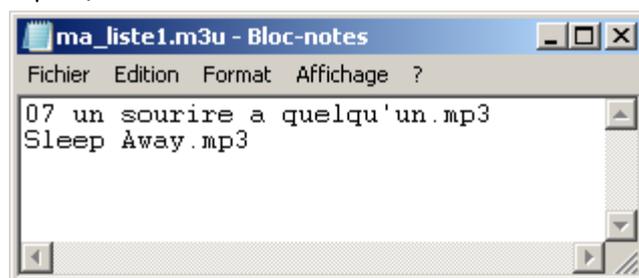
Nous allons donc pouvoir glisser tous ces titres sur notre clé pour qu'elle soit lue par notre lapin.

Il ne reste plus qu'à créer nos playlists. Qu'entendons-nous par playlist d'ailleurs ?

Tout simplement une liste (qui portera un nom comme liste_matin, liste_tonique, liste_rap, liste_groove mais ça peut très bien être 1, 2 ou 3 en fait le nom n'a pas d'importance c'est juste plus facile de savoir ce qu'elle contient quand le nom est précis.

Justement ce que contiendra la liste ce sera tout simplement les noms des fichiers musicaux que nous voulons entendre lorsque ce sera cette liste qui sera jouée.

Plusieurs façons bien entendu pour créer ces listes, la première, artisanale, on crée un fichier txt comme nous l'avons vu dans le tutoriel 1 (clic droit là où il n'y a rien dans le répertoire des musiques => nouveau => document texte puis nous changeons le nom proposé « Nouveau document texte.txt » par « ma_liste1.m3u », pour le nom on met ce que l'on veut (sans accent ni espace) par contre le « .m3u » est obligatoire. Puis clic droit sur le nom => modifier et vous saisissez tous les titres que vous voulez, un à chaque ligne (pour éviter les fautes de frappe faites des copier / coller



Si vous avez beaucoup de titre vous pouvez aussi créer un fichier bat (comme dans le premier tutoriel) et dans ce fichier bat vous tapez :



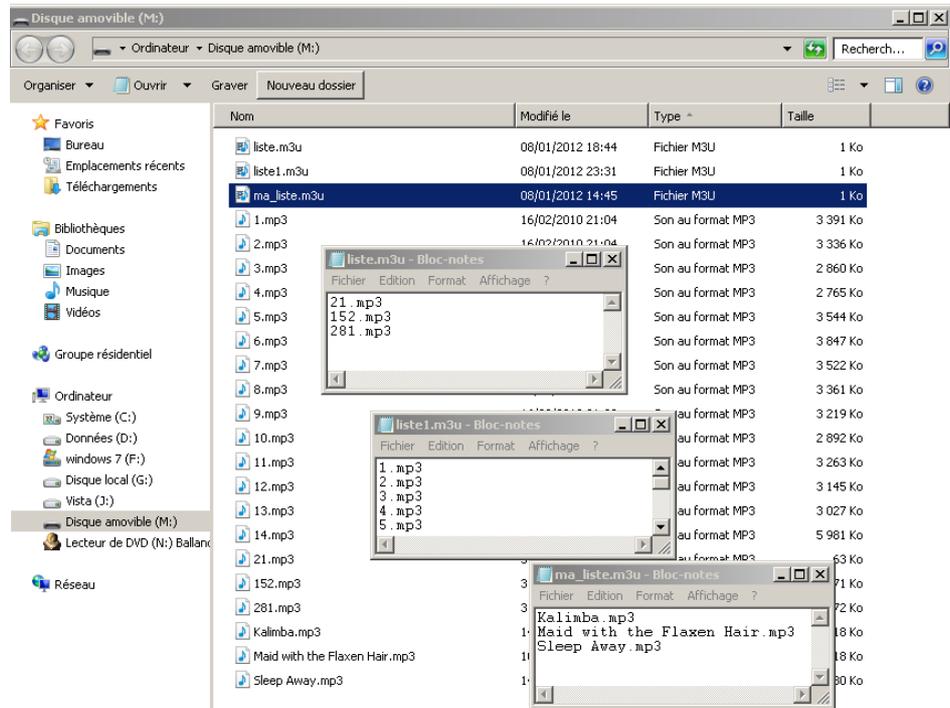
karotz - tutoriel pour débutant sous Windows -



*.mp3 signifie : tous les fichiers mp3 donc si vous avez des .ogg vous mettez *.ogg.

Ce batch va créer ma_liste.m3u avec TOUS vos fichiers présents dans le répertoire, il vous suffit donc de dupliquer cette liste en autant de listes désirées en changeant le nom à chaque fois puis de les ouvrir pour supprimer les titres que vous ne voulez pas.

Votre clé doit donc ressembler à cela



Il ne reste plus qu'à connecter la clé sur le lapin et à le programmer.

Nous commençons par vérifier que le fichier descriptor contient bien l'accès multimedia et le file (pas sur que celui-ci soit nécessaire mais il en vaut mieux plus que moins n'est-ce pas ?)

```
descriptor.xml - Bloc-notes
Fichier Edition Format Affichage ?
<descriptor>
  <name>lecteur_playlist</name>
  <version>0.0.1</version>
  <accesses>
    <access>tts</access>
    <access>serial</access>
    <access>file</access>
    <access>button</access>
    <access>ears</access>
    <access>rfid</access>
    <access>led</access>
    <access>http</access>
    <access>multimedia</access>
    <access>asr</access>
  </accesses>
</descriptor>
```

karotz - tutoriel pour débutant sous Windows -

Vous êtes à présent familier avec le code, la seule particularité qu'il faut connaître si on veut utiliser la clé plutôt qu'un chemin internet c'est :
`/mnt/usbkey/`

Donc si on veut lire un fichier mp3 de la clé on écrira

```
karotz.multimedia.play("/mnt/usbkey/Kalimba.mp3", exitFunction);
```

Si on veut lire une playlist de la clé on écrira

```
karotz.multimedia.play("/mnt/usbkey/ma_liste.m3u", exitFunction);
```

Voici donc le programme minimal

```
include("util.js");
//var karotz_ip = "localhost"
var karotz_ip = "192.168.1.46"//29 jessica 96 dany 46 papa

var buttonListener = function(event) {
  if (event == "DOUBLE") {
    karotz.multimedia.stop();
    exit();
  }
  return true;
}
var exitFunction = function(event) {
  if (event == "TERMINATED") {
    exit();
  }
  return true;
}
var onKarotzConnect = function(data) {
  karotz.button.addListener(buttonListener);
  karotz.multimedia.play("/mnt/usbkey/ma_liste.m3u", exitFunction);
}
karotz.connectAndStart(karotz_ip, 9123, onKarotzConnect, {});
```

Je vous laisse essayer, changer le nom de la playlist il devrait jouer d'autres morceaux.

Nous allons agrémenter (un peu) notre application, tout d'abord en permettant la possibilité de mettre le lecteur en pause lorsque nous pressons une fois le bouton et un nouvel appui pour relancer la lecture (fonction RESUME)

Nous essaierons la variante en faisant bouger les oreilles car nous ne l'avons pas encore utilisé var s'il ne vous a pas échappé que nous pouvions faire bouger les oreilles par programmes on peut aussi commander le programme en bougeant les oreilles.

D'abord avec le bouton.

```
if (event == "SIMPLE") {
  karotz.multimedia.pause();
}
```

Bien ! Donc si nous appuyons une fois sur le bouton le lecteur se met en pause.

Mais comment utiliser le même bouton pour faire exactement le contraire ?

Nous allons utiliser ce que nous appelons un flip/flop ou un oui/non, un vrai/faux.

Nous utiliserons un « drapeau », un flag en langage informatique. Au départ ce drapeau sera initialisé à (faux) et à chaque fois que nous

karotz - tutoriel pour débutant sous Windows -

appuions sur le bouton (une fois) nous inverserons la valeur de ce drapeau il passera donc à l'état vrai puis faux puis vrai puis faux etc (0 -1 - 0 - 1)

Et le test du bouton simple sera complété de la sorte :

Si bouton simple et drapeau à 0 alors mets drapeau à 1 et fait pause

Si bouton simple et drapeau à 1 alors mets drapeau à 0 et fait resume

Cela s'écrit :

```
if ((event == "SIMPLE") && (je_suis_en_pause == 'non')) {
    je_suis_en_pause = 'oui';
    karotz.multimedia.pause();
}
else if ((event == "SIMPLE") && (je_suis_en_pause == 'oui')) {
    je_suis_en_pause = 'non';
    karotz.multimedia.resume();
}
```

Pas mal mais pas de chance ça ne fonctionne pas !

Je pense que le fait de ne pas gérer la fonction event dans pause fait que le programme continue de s'exécuter et arrive à l'instruction exitFunction qui a l'événement à TERMINATED (celui de la pause) et donc sort du programme.

Pas facile la vie d'artiste, je vous le dit ;)

J'ai donc ajouté un test supplémentaire dans la fonction exitFunction comme ceci :

```
if ((event == "TERMINATED") && (je_suis_en_pause == 'non')) {
```

Voici le main.js qui va bien

```
include("util.js");
//var karotz_ip = "localhost"
var karotz_ip = "192.168.1.46" //29 jessica 96 dany 46 papa
var je_suis_en_pause = 'non'; //ou faux ou 0
var buttonListener = function(event) {
    if (event == "DOUBLE") {
        karotz.multimedia.stop();
        exit();
    }
    if ((event == "SIMPLE") && (je_suis_en_pause == 'non')) {
        je_suis_en_pause = 'oui';
        karotz.multimedia.pause();
    }
    else if ((event == "SIMPLE") && (je_suis_en_pause == 'oui')) {
        je_suis_en_pause = 'non';
        karotz.multimedia.resume();
    }
    return true;
}
var exitFunction = function(event) {
    if ((event == "TERMINATED") && (je_suis_en_pause == 'non')) {
        exit();
    }
    return true;
}
var onKarotzConnect = function(data) {
    karotz.button.addListener(buttonListener);
    karotz.multimedia.play("/mnt/usbkey/ma_liste.m3u", exitFunction);
}
karotz.connectAndStart(karotz_ip, 9123, onKarotzConnect, {});
```

Pour le fun nous allons donc utiliser les oreilles pour faire la même chose (oreille droite quand on regarde le lapin (donc gauche ou LEFT pour lui) met en pause, l'autre oreille (la droite ou RIGHT pour lui) redémarre

karotz - tutoriel pour débutant sous Windows -

Lui tirer les oreilles

Mais pourquoi me direz-vous, la dernière fois on lui a déjà tapé sur la tête, il est pourtant gentil ce lapin !



Alors là je me marre je me demande bien comment il va s'y prendre 😊

En fait si nous pouvons lui faire bouger les oreilles par programmes (tutoriel 2) nous pouvons également utiliser les oreilles comme évènement, en effet si nous bougeons les oreilles avec nos mains on peut le tester par programme et réagir en conséquence. Voyons à quoi ressemble cette instruction sur notre site favori

<http://dev.karotz.com/sdk/>

ears

```
karotz.ears.addListener( function( event , step , length ) ;
```

add a listener on ears

value of event : { "START_LEFT"; "START_RIGHT"; "STOP_LEFT"; "STOP_RIGHT"; "BLOCKED_LEFT"; "BLOCKED_RIGHT"; }
step : number of step the ear has made (0 for START_LEFT or START_RIGHT events)
length : time elapsed in ms during the ear move (0 for START_LEFT or START_RIGHT events)

Dans notre cas nous nous intéresserons à l'évènement « START_LEFT » et « START_RIGHT »

Si ça n'est pas déjà fait pensez à compléter votre descriptor.xml avec ears comme access ;)

Dans le onKarotzConnect on ajoute le listener pour les oreilles

```
var onKarotzConnect = function(data) {  
  karotz.button.addListener(buttonListener);  
  karotz.ears.addListener(earsListener);  
  karotz.multimedia.play("/mnt/usbkey/ma_liste.m3u", exitFunction);  
}
```

Puis la gestion des évènements :

```
var earsListener = function(event) {  
  if (event == "START_LEFT") {  
    je_suis_en_pause = 'oui';  
    karotz.multimedia.pause();  
  }  
  if (event == "START_RIGHT") {  
    je_suis_en_pause = 'non';  
    karotz.multimedia.resume();  
  }  
  return true;  
}
```

karotz - tutoriel pour débutant sous Windows -



Notre main.js

```
include("util.js");
//var karotz_ip = "localhost"
var karotz_ip = "192.168.1.46"//29 jessica 96 dany 46 papa
var je_suis_en_pause = 'non'; //ou faux ou 0
var buttonListener = function(event) {
  if (event == "DOUBLE") {
    karotz.multimedia.stop();
    exit();
  }
  return true;
}
var earsListener = function(event) {
  if (event == "START_LEFT") {
    je_suis_en_pause = 'oui';
    karotz.multimedia.pause();
  }
  if (event == "START_RIGHT") {
    je_suis_en_pause = 'non';
    karotz.multimedia.resume();
  }
  return true;
}
var exitFunction = function(event) {
  if ((event == "TERMINATED") && (je_suis_en_pause == 'non')) {
    exit();
  }
  return true;
}
var onKarotzConnect = function(data) {
  karotz.button.addListener(buttonListener);
  karotz.ears.addListener(earsListener);
  karotz.multimedia.play("/mnt/usbkey/ma_liste.m3u", exitFunction);
}
karotz.connectAndStart(karotz_ip, 9123, onKarotzConnect, {});
```

Nous avons de quoi être satisfait ! Si, si je vous assure, moi en tout cas je le suis ;)

Mais en avons-nous fini pour autant ?

Que nenni !



Que nenni : expression un peu désuète de l'époque médiévale. A l'origine l'expression était synonyme de « non » puis le temps passant elle a pris la connotation de « pas du tout »

Que se passe-t-il lorsque la liste a été lue dans son intégralité ? Le programme s'arrête. Mais peut-être aimerions-nous recommencer la lecture au début ! Ici, en local nous allons boucler, le double appui sur la tête permettant de toute façon de quitter. Une fois en ligne on peut imaginer un choix utilisateur permettant à certains profils de boucler et à d'autres de ne jouer qu'une fois.

Il suffit de remplacer le exit() dans exitFunction par onKarotzConnect()

```
var exitFunction = function(event) {
  if ((event == "TERMINATED") && (je_suis_en_pause == 'non')) {
    onKarotzConnect();//exit();
  }
  return true;
}
```

karotz - tutoriel pour débutant sous Windows -

Je vous laisse profiter de votre musique...

Quoi ? Que dites-vous ? Cela fait 15 mn que le lapin jouait de la musique, tout allait pour le mieux dans le meilleur des mondes et le lapin est revenu au vert en arrêtant de jouer ?

Aie aie aie encore une fois, il s'en passe des drôles de chose avec notre lapin.

Oui il s'en passe des drôles mais en l'occurrence pourquoi tant de haine ?

C'est tout simple, pendant que vous écoutiez votre musique il ne vous est pas venu à l'esprit qu'il y avait des applications qui se bousculaient au portillon de votre lapin, une humeur ? La météo ? La mise en sommeil car il est l'heure d'aller se coucher que sais-je encore. Et si notre application était mal conçue et bouclait en permanence on serait bien content qu'à un moment donné le serveur puisse reprendre la main ! C'est exactement ce qu'il se passe en fait. S'il n'y a aucun contact entre le lapin et le serveur Karotz pendant 15 mn le lapin est programmé pour se réinitialiser.

Je vous entends me dire « c'est bien joli tout ça mais si moi j'ai envie d'écouter de la musique plus d'un ¼ d'heure qu'est-ce que je fais ?

Et bien nous allons jouer au ping pong 😊 !

Non non ça n'est pas une blague. Regardez ici

<http://dev.karotz.com/sdk/#karotz-ping>

Le ping envoie une information au serveur et le serveur répond au lapin, c'est transparent pour l'utilisateur mais cela suffit pour repartir pour ¼ d'heure. Ceci étant nous serons prudent et on n'attendra pas les 15 mn pour générer ce ping mais plutôt 8 ou 9 minutes.

L'instruction est simple :

```
Karotz .ping() ;
```

Attention cette instruction ne fonctionne pas en local ? Bug de la VM, certainement (mais elle fonctionne lorsque l'appli est en ligne), j'utilise donc l'ancienne instruction qui, elle, fonctionne aussi bien en local qu'en « embarquée » :

```
ping() ;
```

Le souci restant est d'exécuter cette instruction toutes les x minutes, j'ai choisi 10 mn soit 600 secondes soit 600 000 millisecondes.

Nous allons utiliser l'instruction setTimeout pour cela

```
var monping = function(event) {  
  ping();  
  log("ping");  
  setTimeout(600000, function() { monping(); return true; });  
};
```

Et bien entendu on n'oublie pas d'exécuter cette instruction une première fois (par la suite elle s'autorelance toutes les 10 mn) en ajoutant cette ligne dans onKarotzConnect :

```
var onKarotzConnect = function(data) {  
  karotz.button.addListener(buttonListener);  
  karotz.ears.addListener(earsListener);  
  monping();  
  karotz.multimedia.play("/mnt/usbkey/liste.m3u", exitFunction);  
};
```

karotz - tutoriel pour débutant sous Windows -

Voici le code définitif ;)

```
include("util.js");
//var karotz_ip = "localhost"
var karotz_ip = "192.168.1.46"//29 jessica 96 dany 46 papa
var je_suis_en_pause = 'non'; //ou faux ou 0
var buttonListener = function(event) {
  if (event == "DOUBLE") {
    karotz.multimedia.stop();
    exit();
  }
  return true;
}
var earsListener = function(event) {
  if (event == "START_LEFT") {
    je_suis_en_pause = 'oui';
    karotz.multimedia.pause();
  }
  if (event == "START_RIGHT") {
    je_suis_en_pause = 'non';
    karotz.multimedia.resume();
  }
  return true;
}
var exitFunction = function(event) {
  if ((event == "TERMINATED") && (je_suis_en_pause == 'non')) {
    onKarotzConnect();
  }
  return true;
}
var onKarotzConnect = function(data) {
  karotz.button.addListener(buttonListener);
  karotz.ears.addListener(earsListener);
  monping();
  karotz.multimedia.play("/mnt/usbkey/liste.m3u", exitFunction);
}
karotz.connectAndStart(karotz_ip, 9123, onKarotzConnect, {});
```

MA GNI FI QUE ! Vous ne trouvez pas, vous n'avez pas parfois l'impression d'être un génie en herbe ?

Mais (il y a souvent un « mais », vous avez remarqué ;)) vous n'avez pas l'impression que notre lapin a un fil à la patte, si on a besoin de l'ordinateur pour faire lire de la musique à notre lapin autant lire la musique sur l'ordinateur, non ?

Alors nous allons apprendre à mettre nos applications en ligne. Que ce soit pour le plus grand nombre (application mise dans le store par l'équipe Violet) ou pour vous et vos proches, application « privée » mais hébergée par le serveur Karotz.

karotz - tutoriel pour débutant sous Windows -

Regarde, mon ami lapin, je suis publié !

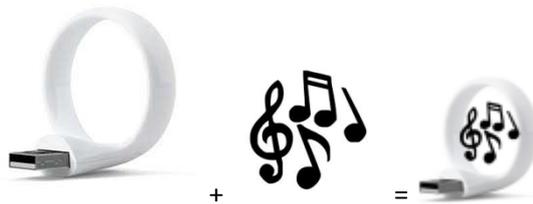
Ne serait-ce pas la plus belle récompense à vos efforts ?

Mais avant cela il va falloir un peu de discipline et encore quelques efforts. Et de la rigueur. Et quelques outils et un peu d'imagination...

Pour les outils :

1 - Un logiciel de dessin (Paint peut faire l'affaire) pour réaliser l'icône qui sera affichée sur le site. De préférence 64 x 64 pixels format jpg, gif ou png

Par exemple pour l'appli « lecteur de playlist » on pourrait choisir cette image



2 - Ensuite un outil pour zipper les fichiers, il en existe plusieurs, personnellement j'utilise IZarc qui est un gratuitciel

Quels sont les fichiers qu'il faut mettre dans le zip ?

Descriptor.xml

Screen.xml

Main.js

Util.js

+ les autres fichiers utilisés comme tts2.js si vous utilisez les lascars par exemple.

karotz - tutoriel pour débutant sous Windows -

Le fichier descriptor.xml

En plus des access que l'on connaît

On trouvera le nom de l'application

```
<name>nom_de_l_appli</name>
```

Et le numéro de version

```
<version>0.0.1</version>
```

A chaque mise à jour Karotz vérifiera que la version que vous chargez a bien un numéro supérieur à celui déjà présent.

On peut mettre plein d'autre chose mais c'est le minimum fonctionnel.

Le voici dans sa totalité :

```
<descriptor>
  <name>lecteur_playlist</name>
  <version>0.0.1</version>
  <accesses>
    <access>tts</access>
    <access>file</access>
    <access>button</access>
    <access>ears</access>
    <access>rfid</access>
    <access>led</access>
    <access>multimedia</access>
    <access>asr</access>
  </accesses>
</descriptor>
```

Attention : PAS de caractères accentués ni c cédille (ç) dans les fichiers xml (descriptor et screen)

Les access présents dans ce fichier se retrouvent dans l'écran de chargement de l'application



karotz - tutoriel pour débutant sous Windows -



Le fichier screen.xml

Avant de voir son contenu comprenons à quoi il sert.

Vous avez tous installé au moins une application et êtes donc arrivés à une page qui ressemble à ça :

suricate

NOUVEAU PROFIL

config

config 1

Nom du profil : config

Commande vocale : "lecteur playlist config"

Mode repeat : Oui

Playlist : liste1.m3u

Déclenchement manuel

Déclenchement permanent

Sélection d'un Nanoz : BLACK-8

moins souvent ————— plus souvent

3 h ————— 1 mn

Déclenchement planifié

Lun Mar Mer Jeu Ven Sam Dim

00 : 00

ANNULER VALIDER

Les 4 déclenchements (manuel, permanent, planifié et commande vocale) sont soit présents (true) soit absents (false) et sont écrits comme cela dans le fichier screen.xml

```
<screen nanoTrigger=" true"  
permanentTrigger=" true"  
scheduledTrigger=" true"  
scheduledDateTrigger=" true"  
voiceTrigger=" true">
```

Vous mettez "false" à la place de "true" si vous ne voulez pas d'un déclencheur.

Si vous ne mettez aucun déclencheur (votre fichier screen actuel) ils seront tous « true » par défaut donc ne mettez que ceux qui sont « false »

Ces champs

Mode repeat : Oui

Playlist : liste1.m3u

sont particuliers puisque dépendent entièrement de votre usage. Votre application s'adressant au plus grand monde on imagine bien que tout le monde n'a pas les mêmes goûts ni les mêmes envies. Dans l'application que nous mettons en ligne, le lecteur de playlist vous imaginez bien que les utilisateurs auront leur propre nom de playlist il va donc falloir qu'ils renseignent ce nom pour chacun des profils qu'ils utiliseront, c'est le champ « playlist ».

karotz - tutoriel pour débutant sous Windows -



Et puis nous avons fait en sorte que notre application lise les fichiers en boucle (mode repeat) mais l'utilisateur peut vouloir ne lire qu'une fois telle ou telle liste. C'est la raison d'être de la liste déroulante « mode repeat » qui contient 2 choix, « oui » et « non », un seul des deux étant possibles (il existe des listes multichoice)

Donc non seulement nous allons apprendre à créer ces champs (champ texte ou liste déroulante, avec ces valeurs, sa valeur par défaut. Seuls ces 2 types sont possibles (pas de case à cocher par exemple).

Pour mettre un champ texte (comme le champ playlist de cet exemple) :

Playlist :

```
<text label="Playlist" name="playlist" default="liste1.m3u" validation="" required="true" errorMessage="" />
```

Comme toujours, attention à ne pas mettre d'accents ni de ç.

```
label="Playlist" => Playlist est le texte qui s'affiche dans l'écran
name="playlist" => le nom du paramètre que l'on va récupérer dans notre main.js
default="liste1.m3u" => valeur par défaut présente dans le champ, permet ici de
rappeler qu'il faut penser à l'extension m3u par exemple
validation=""
required="true" => "true" champ obligatoire
errorMessage=""
```

Pour mettre une liste déroulante comme « mode repeat (oui /non) :

Mode repeat :
Playlist :

```
<select label="Mode repeat" name="repeat" type="one" required="true">
  <option label="Oui" key="oui" checked="true"/>
  <option label="Non" key="non" checked="false"/>
</select>
```

Comme toujours, attention à ne pas mettre d'accents ni de ç.

```
label=" Mode repeat " => Mode repeat est le texte qui s'affiche dans l'écran
(ici en anglais pour éviter les accents justement)
name="repeat" => le nom du paramètre que l'on va récupérer dans notre main.js
type="one" => choix unique
ou
type="multiple" => choix multiple
required="true" => "true" champ obligatoire, « false » non obligatoire
```

Pour le contenu de cette liste :

```
label="oui" => le mot oui sera affiché dans la liste
key="oui" => la valeur correspondant à ce choix qui sera testée dans le main.js
checked="true" => si true alors c'est le choix qui sera affiché par défaut,
sinon false, bien entendu sur un choix unique on ne peut avoir qu'un true dans
la liste
```

Voici donc le screen.xml que j'utilise

```
<screen>
  <select label="Mode repeat" name="repeat" type="one" required="true">
    <option label="Oui" key="oui" checked="true"/>
    <option label="Non" key="non" checked="false"/>
  </select>
  <text label="Playlist" name="playlist" default="liste1.m3u" validation="" required="true" errorMessage="" />
</screen>
```

karotz - tutoriel pour débutant sous Windows -

Voici en exemple screen que j'utilise pour l'appli « on ne dit pas »

```
<screen>
  <select label="Voix" name="voice" type="one" required="true">
    <option label="Electra" key="800" checked="false"/>
    <option label="Bicool" key="0" checked="true"/>
    <option label="Mamie" key="400" checked="false"/>
    <option label="Papy" key="100" checked="false"/>
    <option label="Helium" key="200" checked="false"/>
    <option label="Sidoo" key="700" checked="false"/>
    <option label="Ramboo" key="600" checked="false"/>
    <option label="Julie" key="300" checked="false"/>
    <option label="Matteo" key="500" checked="false"/>
    <option label="Au hasard" key="8" checked="false"/>
  </select>
</screen>
```

Vous noterez le « au hasard » et non pas « aléatoire » qui contient un accent.

C'est juste pour montrer que la valeur « key » peut être différente du label. Ici j'utilise en plus une astuce, j'ai 70 phrases, une est choisie au hasard et j'y ajoute la valeur « key » de la voix sélectionnée par l'utilisateur (les 70 phrases de mamie sont numérotées de 400 à 469, les 70 de Sidoo de 700 à 769, les 70 de bicool de 0 à 69 etc, le 8 du hasard me sert à choisir un nombre aléatoire entre 0 et 7 que je multiplie par 100 pour trouver une des valeurs des voix.

Revenons à nos paramètres et voyons comment les utiliser dans notre main.js

Les paramètres de l'écran sont utilisables dans le programme sous la forme :

params[instanceName].ici le nom de notre paramètre donc

`params[instanceName].repeat`

pour la liste déroulante et `params[instanceName].repeat`

prend la valeur « oui » ou la valeur « non »

et

`params[instanceName].playlist`

pour le nom de la playlist

Donc notre main.js se présentera comme cela (surlignées en rouges les anciennes instructions et surlignées en vert les nouvelles instructions

karotz - tutoriel pour débutant sous Windows -



```
include("util.js");
var karotz_ip = "localhost";
var karotz_ip = "192.168.1.100"; // 29 jessica 96 dany 46 papa
var je_suis_en_pause = 'non'; //ou faux ou 0
var buttonListener = function(event) {
    if (event == "DOUBLE") {
        karotz.multimedia.stop();
        exit();
    }
    return true;
}
var earsListener = function(event) {
    if (event == "START_LEFT") {
        je_suis_en_pause = 'oui';
        karotz.multimedia.pause();
    }
    if (event == "START_RIGHT") {
        je_suis_en_pause = 'non';
        karotz.multimedia.resume();
    }
    return true;
}
var exitFunction = function(event) {
    if ((event == "TERMINATED") && (je_suis_en_pause == 'non') && (params[instanceName].repeat == 'non')){exit();}
    if ((event == "TERMINATED") && (je_suis_en_pause == 'non') && (params[instanceName].repeat == 'oui')) {
        onKarotzConnect();
    }
    return true;
}
var onKarotzConnect = function(data) {
    karotz.button.addListener(buttonListener);
    karotz.ears.addListener(earsListener);
    monping();
    //karotz.multimedia.play("http://192.168.1.100:8080", exitFunction);
    karotz.multimedia.play("/mnt/usbkey/" + params[instanceName].playlist, exitFunction);
}
karotz.connectAndStart(karotz_ip, 9123, onKarotzConnect, {});
```

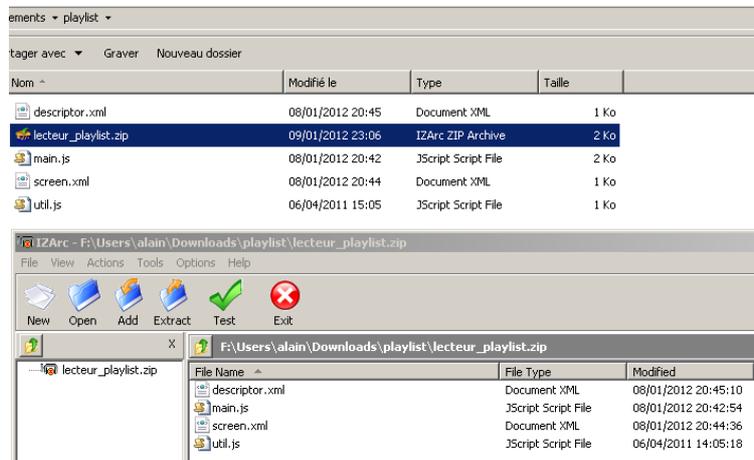
Oui cela fait bizarre de retrouver :

```
Var karot_ip = "localhost"
```

Mais vous l'avez compris, adieu NOTRE lapin, bienvenue aux lapins de la terre entière pour notre application.

Si par malheur vous laissez votre adresse IP, vos test connectés fonctionneraient tout comme tous les lapins ayant la même adresse mais malheureusement pas les autres. Ça n'a l'air de rien mais le premier test que l'on fait une fois l'application en ligne c'est avec notre lapin et on peut ne pas voir qu'on a oublié de modifier ce paramètre.

Que reste-il à faire de notre côté ? Créer le fichier zip contenant les 2 fichiers xml et tous les fichiers .js de notre application





Dernière ligne droite (jusqu'à la prochaine ;))

Connectons-nous ici

<http://www.karotz.com/login>

et identifions nous.

Puis en bas à droite cliquons sur :



Puis sur



La description est ce qui apparaît ici dans le choix de l'application



Help apparait lorsqu'on paramètre l'application



karotz - tutoriel pour débutant sous Windows -



Category:

Short description:

Description:

Help:

Icon:

Tags:

Lorsque vous validez vous obtenez ceci :

 **Ma premiere appli en ligne - FRA -** No version registered yet
API key: bb4...
Secret key: 5...
Status: IDLE
Bought 0 times
Installed on 0 Karotz

En cliquant sur « edit details » vous revenez au menu précédent. Nous ce que nous voulons c'est ajouter une nouvelle version on clique donc sur « Add a new version »

UPLOAD CODE

Update note:

Zip:

En validant, différents tests vont être faits, intégrité des fichiers xml (attention aux accents), vérification que le numéro de version est

karotz - tutoriel pour débutant sous Windows -

supérieur à celui déjà présent sur le serveur, que le champ update note ne soit pas vide et certainement d'autres vérifications que je n'ai pas eu le plaisir de rencontrer ;)

Si tout va bien vous obtenez ceci :



Ma premiere appli en ligne - FRA - 0.0.1
API key: bl
Secret key: ...
Version: 0.0.1
Status: IDLE
Bought 0 times
Installed on 0 Karotz

Edit details Add a new version Make private Delete

Il suffit de cliquer sur « make private » pour pouvoir tester votre appli



Ma premiere appli en ligne - FRA - 0.0.1
API key: bb4:
Secret key: !
Version: 0.0.1
Status: PRIVATE
Bought 0 times
Installed on 0 Karotz

Edit details Add a new version Submit Test! Delete

Le statut est passé de IDLE à PRIVATE et vous avez un bouton « test ! » qui vous permet d'installer l'appli sur votre lapin (ou en communiquant le lien qui apparait dans la barre d'adresse à vos amis / famille / collègues de tester également.

Add a new version permet de charger une mise à jour, version 0.0.2 par exemple

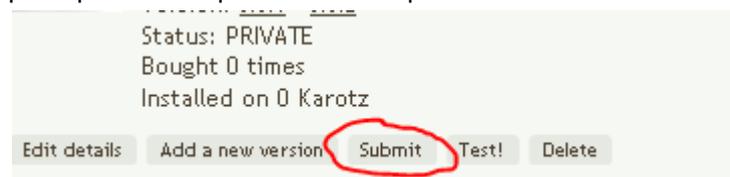


Version: 0.0.1 - 0.0.2
Status: PRIVATE
Bought 0 times
Installed on 0 Karotz

Edit details Activate next version Update next version Test! Delete

ATTENTION : à ce stade si vous cliquez sur Test ! c'est toujours l'ancienne version que vous testeriez, il faut d'abord activer la nouvelle version en cliquant sur le bouton « activate next version »

Lorsque tous vos essais sont concluants, que tout votre entourage vous félicite pour cette superbe application, que vous êtes convaincus qu'elle va plaire au plus grand nombre alors vous pouvez décider de la rendre publique. Il suffit pour cela de cliquer sur « Submit »



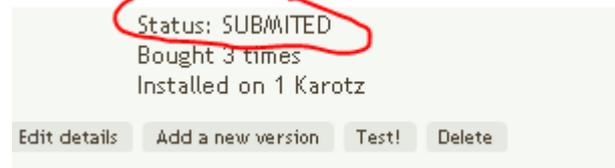
Status: PRIVATE
Bought 0 times
Installed on 0 Karotz

Edit details Add a new version **Submit** Test! Delete

karotz - tutoriel pour débutant sous Windows -



Le statut passera alors à SUBMITTED



Et le plus dur est là !!!!

Car il faut à présent ATTENDRE.

Attendre que l'équipe Karotz teste votre appli, vous communique éventuellement différentes remarques, correctifs à apporter ... Cela peut durer parfois 2 semaines, parfois plus, vous allez guetter, scruter, jusqu'au jour où....vous avez le statut ENABLED



Et vous croyez que c'est mieux à présent, que nenni, vous guettez le compteur de chargement, les commentaires...

Vous voulez que je vous dise ? pas de tout repos le métier de programmeur, c'est moi qui vous le dit.

Ceci étant je vous souhaite d'en publier plein et d'obtenir de grands succès.

Et ne croyez pas que nous en avons terminé, il y aura encore au moins un tutoriel, j'ai lu sur les différents forums que Violet devait sortir un pack Family qui n'est jamais sorti, jouer avec Karotz en société vous imaginez ? Par équipe, Karotz comme arbitre quoi de plus génial, moi l'idée me séduit, nous essaierons de faire une ébauche de jeu que chacun complètera à sa façon.

En attendant bon développement....