

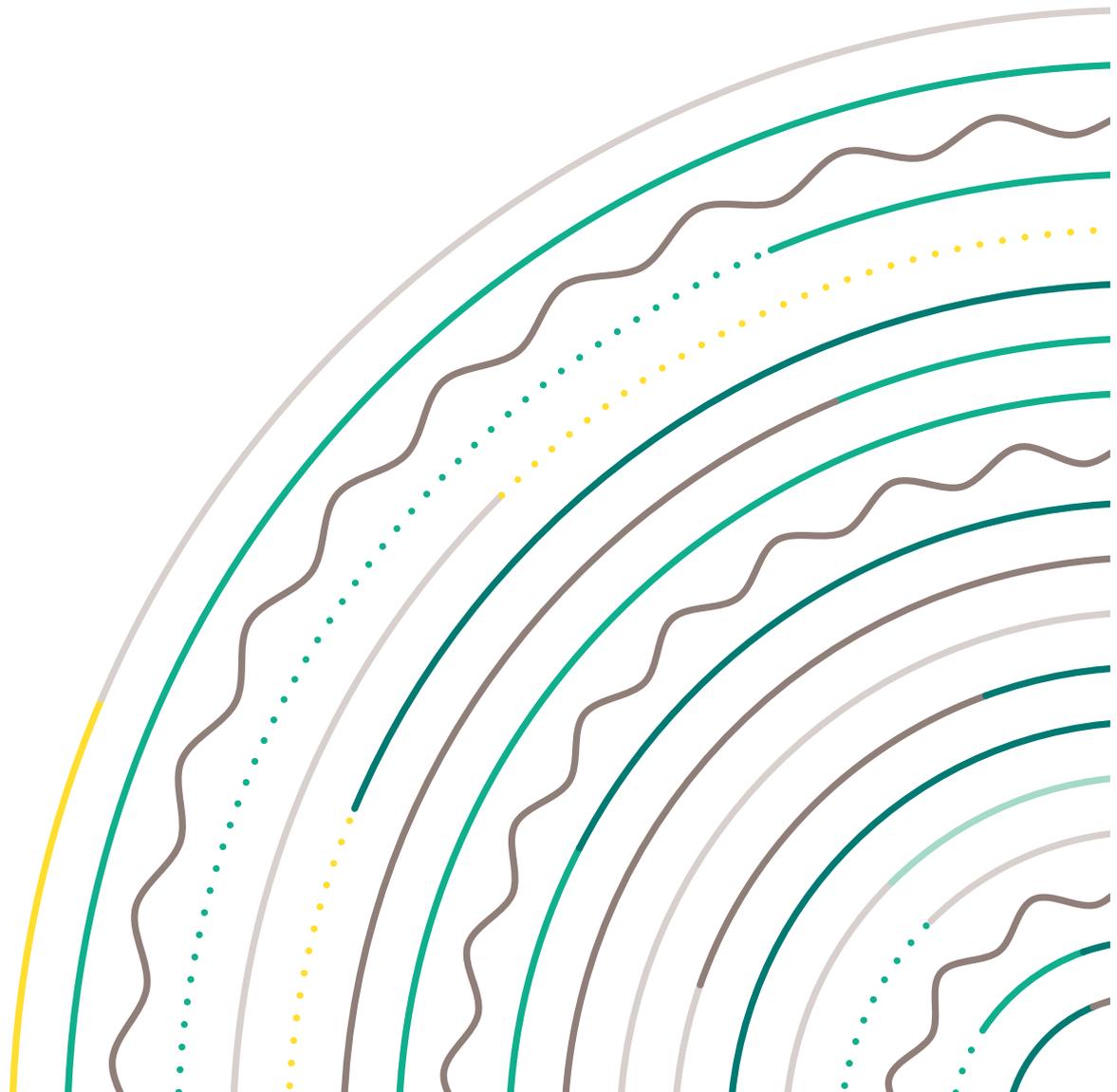


# Lifedomus

# Universal

29/01/2018

Versión 1.2



# Protocolo Universal

## Índice

1	Principio de funcionamiento .....	3
2	Creación del conector .....	3
2.1	Propiedades comunes .....	3
2.1.1	Importar/Exportar .....	3
2.1.2	Depurador .....	3
2.1.3	Lectura cíclica.....	3
2.1.4	Persistencia de los retornos de estados .....	3
2.2	Conector HTTP .....	4
2.3	Conector IP .....	6
3	Javascript.....	9
3.1	Variables reservadas: .....	9
3.2	Funciones útiles .....	9
3.3	Tipos enumerados.....	9
3.4	Tipo Lista .....	10
3.5	Parámetros de acciones.....	10
4	Propiedades de los equipos .....	11
4.1	Edición de acciones.....	12
4.1.1	Propiedades específicas del conector HTTP .....	13
4.2	Edición de variables .....	14
5	Widgets .....	14
5.1	URL .....	14
5.2	Slidebar.....	15
5.3	TrackBar.....	15
5.4	Liste .....	16

# 1 Principio de funcionamiento

El conector universal permite agregar, en Lifedomus, un protocolo personalizado que permite controlar un equipo de audio/vídeo, un protocolo domótico o cualquier equipo cuyo control se pueda realizar en IP, RS o incluso HTTP.

El principio es simple, consiste en enviar comandos ASCII o hexadecimales a un módulo ubicado en la red e identificado por una dirección específica.

## 2 Creación del conector

### 2.1 Propiedades comunes

#### 2.1.1 Importar/Exportar



: Puede exportar su módulo para compartirlo. La exportación comprende la configuración del conector, de JavaScript, de los equipos vinculados a este conector así como de los modelos asociados a los equipos si son diferentes de los modelos del sistema.



: Puede importar un script en JavaScript para realizar el análisis sintáctico de los datos. Durante la importación del script, se compilará y se le notificará en caso de error de compilación. Si procede, se conservará el antiguo script. Habrá que reiniciar el conector para que el JS sea realmente tenido en cuenta.

#### 2.1.2 Depurador

El icono  le permite abrir un navegador web que muestra el depurador de su conector. Esta vista contendrá todas las tramas recibidas así como todos los mensajes que haya podido escribir en su JS con la ayuda del método `ldprint()`.

Si su JavaScript cuenta con un bucle infinito, Lifedomus detendrá el conector y mostrará «Break Bl» en el depurador.

#### 2.1.3 Lectura cíclica

En las propiedades del conector, puede determinar el plazo de actualización de determinados estados. Este plazo corresponde al envío repetitivo de comandos cuya lectura cíclica esté activada (véase Propiedades del equipo)

#### 2.1.4 Persistencia de los retornos de estados

Por defecto, todos los retornos de estados se reinician al reiniciarse el servidor. Se puede conservar la persistencia de estos estados en cada reinicio.

## 2.2 Conector HTTP

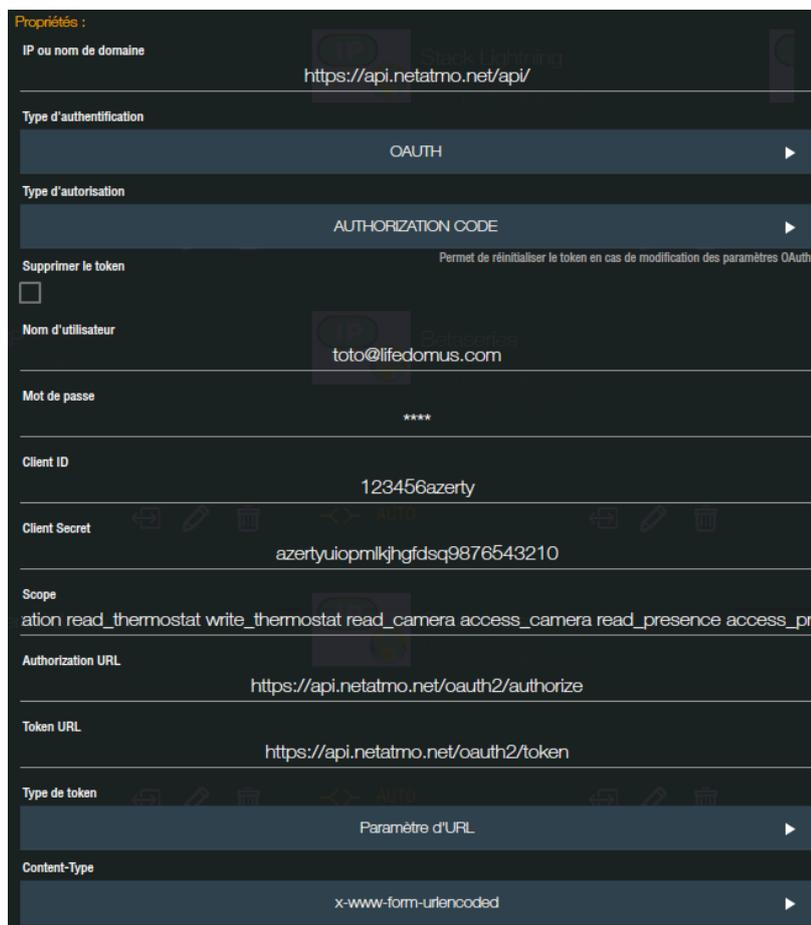
En gestión de conectores, seleccione en la cinta de la izquierda «Universal» y «TCP/IP Universal» y cree un conector «Universal HTTP».

En primer lugar, indique la URL de la interfaz de programación de aplicaciones (API) o del equipo y, a continuación, si procede, complete los parámetros de autenticación.

### Autenticación OAUTH:

Si la autenticación es de tipo OAUTH (solo se admite la V2), tendrá que completar las propiedades siguientes:

- Tipo de autorización: Corresponde al parámetro «grant\_type», especificado generalmente al nivel de la API
- Eliminar token: Permite reinicializar el token en caso de modificación de los parámetros (ámbito, ID de cliente,...)
- Nombre de usuario, contraseña, ID de cliente, cliente secreto: Identificadores de la cuenta
- Ámbito: Especifique los diferentes ámbitos (separados por un espacio) necesarios para el funcionamiento de los distintos comandos
- Tipo de token:
  - Portador: Envía el token a través de un encabezado «Authorization: Bearer {token}» (Autorización: Portador {token})
  - Parámetro de URL: Añade un parámetro «access\_token={token}» en la URL
- Tipo de contenido: La manera en que se publicarán los parámetros OAuth durante la llamada a «Token URL» (URL del token).



The screenshot shows the configuration interface for an OAUTH connector. The fields are as follows:

- IP ou nom de domaine:** https://api.netatmo.net/api/
- Type d'authentification:** OAUTH
- Type d'autorisation:** AUTHORIZATION CODE
- Supprimer le token:**  (Note: Permet de réinitialiser le token en cas de modification des paramètres OAuth)
- Nom d'utilisateur:** toto@lifedomus.com
- Mot de passe:** \*\*\*\*
- Client ID:** 123456azerty
- Client Secret:** azertyuiopmlkjhgfdsq9876543210
- Scope:** ation read\_thermostat write\_thermostat read\_camera access\_camera read\_presence access\_pr
- Authorization URL:** https://api.netatmo.net/oauth2/authorize
- Token URL:** https://api.netatmo.net/oauth2/token
- Type de token:** Paramètre d'URL
- Content-Type:** x-www-form-urlencoded

Durante la 1ª conexión, su navegador de Internet se abrirá solicitándole el acceso de la aplicación Lifedomus a los datos de la API. Valide y obtendrá un mensaje que le informa de si la conexión ha sido o no correcta.

### Tramas de inicialización:

Para controlarlos o antes de poder comunicarse con ellos, algunos módulos necesitan una autenticación u estar en un modo particular. Puede indicar aquí una o todas las tramas enviadas en el inicio del conector. Las tramas se enviarán una tras otra.

Haga clic en el botón «+» para configurar las diferentes tramas:

The screenshot shows a REST client interface with a dark theme. On the left, there is a sidebar with two items: 'Get Datas' and 'Init status'. 'Init status' is selected and highlighted in orange. The main area displays the configuration for the 'Init status' endpoint:

- Label :** Init status
- Commande :** /init\_status
- Méthode :** POST (selected in a dropdown menu)
- Paramètres :** A text area containing a JSON object: 

```
{  "on" : true,  "xy" : [0.123, 0.456]}
```
- En-tête :** A table with two columns: 'Nom' and 'Valeur'. One entry is visible: 'Accept' with the value 'application/json'.

Indique la etiqueta, el comando a enviar así como el método HTTP de comunicación (GET, POST, PUT, DELETE). Si el método de envío es diferente de GET, puede indicar parámetros de control, con formato bruto (ej.: `on=true&value=1234`), en el formato JSON.

Por último, indique uno o varios encabezados http y valide.

## 2.3 Conector IP

Indique la dirección IP y el puerto de comunicación del módulo y especifique el protocolo utilizado para el envío de los comandos.

Los campos siguientes son opcionales:

Carácter(es) de escape al final de una trama enviada:

Carácter o caracteres puestos al final de cada trama enviada al módulo. La mayoría de las veces son los caracteres «\n» y/o «\n». Estos caracteres se representan en las documentaciones de varias formas.

En Lifedomus hay que introducir el código hexadecimal del carácter o caracteres enviados:

- 0A para \n o <LF>
- 0D para \r o <CR>

Si, por ejemplo, los dos son necesarios, hay que introducir 0A0D para que el comando de escape enviado esté en el orden 0A y luego en 0D.

Si tiene que enviar un código diferente, introduzca el código correspondiente en hexadecimal.

Tiempos de espera entre los comandos enviados:

Algunos módulos gestionan mal el envío demasiado próximo de comandos. Por tanto, puede especificar el tiempo de espera mínimo entre el envío de diferentes comandos. Este tiempo se expresa en ms.

Trama(s) de inicialización del conector:

Para controlarlos o antes de poder comunicarse con ellos, algunos módulos necesitan una autenticación u estar en un modo particular. Puede indicar aquí una o todas las tramas enviadas en el inicio del conector.

Durante el arranque del conector, si no se realiza correctamente, un mensaje indicará el error correspondiente.

Indique el tipo de tramas que se van a enviar: ASCII o HEXA.

En HEXA, los comandos estarán compuestos de códigos hexadecimales pegados unos a otros.

Ejemplo: trama enviada a un lport



### Sent from Controller

RequestIPodName	
Value	Meaning
0x2A	Start character
0x49	Device type (iPort)
0x00	Packet Length High Byte
0x03	Packet Length Low Byte
0x38	Packet Command Type
0x3C	Packet Command High Byte
0x28	Packet Command Low Byte
0x18	Checksum

Esta trama se escribirá de este modo en Lifedomus: 2A490003383C2818

#### Carácter(es) de escape de fin de trama recibida:

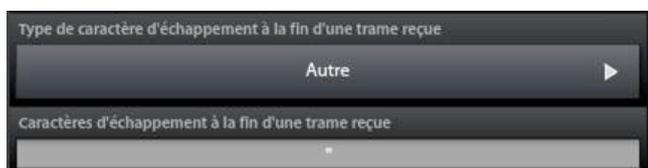
*Este parámetro permite activar la recuperación de información a través de JavaScript. Si solo quiere realizar el control, puede dejar «?» en este campo.*

Es el carácter o caracteres que permiten realizar el análisis sintáctico de los datos recibidos por el conector. Puede seleccionar <CR>, <LF> o los dos.

En caso de que no haya carácter de escape y/o que quiera calcular usted mismo la longitud de la trama, puede indicar «Ninguno».

Si es otro carácter, seleccione «Otros» en la lista desplegable e indique el carácter o caracteres directamente en el campo siguiente.

Ejemplo: trama terminada por una estrella



Finalmente, el último caso, si el tamaño de la trama es variable y solo puede conocerse leyendo una parte de la trama, puede indicar cómo calcularlo.

Para ello, seleccione «Tamaño» en el combo y rellene los campos siguientes:

Valor a agregar al tamaño a calcular: si el tamaño de la trama no tiene en cuenta el encabezado, agregue el tamaño de este encabezado.

Tipo de cálculo del tamaño: puede seleccionar "byte" o "bit". Si el tamaño se calcula con los bytes de la trama, seleccione "byte" (mayoría de casos). Si el tamaño se calcula con los X primeros o últimos bits de un byte, hay que especificar "bit".

Primer Byte/bit y Último Byte/bit: Especifique el 1<sup>er</sup> y último byte o bit para el cálculo

Ej.: lport:

## Packet Structure

The iPort communication packet structure is constructed using a combination of ASCII characters and hex data. A packet consists of seven elements described below.

Byte Number	Value	Meaning
0x00	0x2A	Start character
0x01	0x49	Device type (iPort)
0x02	0xNN	Packet Length High Byte
0x03	0xNN	Packet Length Low Byte
0x04	0xNN	Packet Command Type
0x05	0xNN	Packet Command High Byte
0x06	0xNN	Packet Command Low Byte
0x07 – 0xNN	0xNN	Packet Data Bytes
0xNN	0xNN	Checksum = 2's compliment of the sum of all bytes excluding the start character and checksum

Valeur à ajouter à la taille calculée

4

---

Type de calcul de la taille

Octet ▶

---

Premier octet ou bit

2 ←

---

Dernier octet ou bit

3 ←

## 3 Javascript

Para el análisis sintáctico de los datos, es totalmente libre del desarrollo y puede usar todas las funciones JavaScript. He aquí algunas funciones útiles y variables reservadas:

### 3.1 Variables reservadas:

- trama: contiene la respuesta reenviada por el módulo y, por tanto, es la base de cada analizador sintáctico. Manipule pues esta variable para actualizar sus retornos de estados.  
Según el tipo de caracteres de escape al final de una trama, se escribirá «trama» en cadena de caracteres, en bytes o en tabla de byte:
  - Ninguno: La trama contendrá un byte
  - Otros, \r, \n, \r\n: La trama contendrá una cadena de caracteres
  - Tamaño: La trama contendrá una tabla de byte
- comando: como en los comandos de equipo tipo JS detallados a continuación, existe una variable reservada a Lifedomus que es la variable «comando». Si escribe alguna cosa en esta variable, Lifedomus la enviará al final de la ejecución de su JavaScript, lo que le permitirá crear un encadenamiento de comandos.
- ip: en el caso de un conector Universal IP, la variable IP está reservada por Lifedomus para suministrarle la dirección IP introducida en las propiedades del conector.

### 3.2 Funciones útiles

- `ldprint(mensaje)`: este método permite mostrar un mensaje en el depurador
- `utf8_decode(utf8_texto)`: decodifique una cadena al formato UTF-8
- `JSON.parse(str)` : analice sintácticamente una cadena de caracteres en JSON y construya el valor JavaScript o el objeto descrito por esta cadena.
- `eval(str)`: evalúe el código JavaScript representado en forma de una cadena de caracteres. Muy útil para asignar variables de manera dinámica.

### 3.3 Tipos enumerados

Para los equipos no genéricos (ej.: lámpara, persiana motorizada,...), por defecto hay disponibles un determinado número de acciones y retornos de estados. Estos estados pueden ser de tipo simple (BOOLEANO, CADENA, DOBLE), o de los tipos enumerados definidos en el sistema Lifedomus. Estos tipos se recogen en JS, y los diferentes valores son:

```
CLIM_MODE =
{
    AUTO: 0,
    HEAT: 1,
    COOL: 3,
    FAN: 9,
    DRY: 14,
    OFF: 99
}

BATTERY_LEVEL =
{
    NO_STATUS_AVAILABLE: 0,
    LOW: 1,
    MEDIUM: 2,
    FULL: 3
}

FAN_SPEED =
```

```
{
    LOW: 0,
    HIGH: 1,
    MEDIUM: 2,
    AUTO: 3,
    TOP: 4
}

THERMOSTAT_HEAT_MODE =
{
    OFF: 0,
    HEAT: 1,
    COOL: 2,
    AUTO: 3
}

THERMOSTAT_MODE =
{
    OFF: 0,
    FREEZEOUT: 1,
    ECO : 2,
    CONFORT_MINUS_1: 3,
    CONFORT_MINUS_2: 4,
    CONFORT: 5,
    MANUAL: 6,
    PROGRAM: 7,
    AWAY: 8,
    REDUCED: 9,
    PROTECT: 10
}
```

Así pues, por ejemplo, si utiliza una variable denominada «mode» para gestionar el modo de una climatización universal, puede asignar la variable en su analizador sintáctico del modo siguiente:

```
mode = CLIM_MODE.COOL;
```

### 3.4 Tipo Lista

En Universal existe un último tipo específico, Lista. Este tipo sirve únicamente para la gestión de la lista de widgets en Design Studio, y se representa bajo el formato de una tabla de 2 dimensiones de la manera siguiente:

```
var liste = new Array() ;
liste[0] = new Array() ;
liste[0][0] = '1';
liste[0][1] = 'Element 1';
liste[1] = new Array() ;
liste[1][0] = '2';
liste[1][1] = 'Element 2';
...
```

### 3.5 Parámetros de acciones

Antes de la ejecución de una acción universal hacia un equipo, puede que quiera recuperar el valor de una variable para modificarla o utilizarla como parámetro de acción. Para ello, solo tiene que poner el nombre de la variable entre llaves. Los parámetros utilizables pueden ser un parámetro de acción o cualquier variable javascript definida o no como variable de equipo.

Ejemplo con la utilización de una variable «value»:

```
value = valor + 1;
command = '/setCpt={value}';
```

**Atención:** Si se utiliza esta funcionalidad a nivel de los parámetros HTTP en formato JSON (véase Propiedades de los equipos), el parámetro «value» se debe escribir de esta manera:  $\$(value)$

## 4 Propiedades de los equipos

El conector universal está disponible para todos los equipos genéricos, así como para todos los equipos siguientes:

<b>Audio/Vídeo</b>	Lector de DVD Televisión Satélite Amplificador Sintonizador Videoprojector
<b>Calefacción/Climatización</b>	Termostato de ambiente Radiador Calentador de toallas Caldera Calentador de agua Climatización Suelo radiante
<b>Iluminaciones y tomas</b>	Iluminación Tomas eléctricas Variador 230 V Variador 1-10 V LED RVB
<b>Motorización</b>	Pantalla de cine Portal eléctrico Puerta eléctrica Puerta de garaje Persiana-toldo Persiana de piscina Persiana motorizada Toldo Tragaluz Cortina horizontal Cortina vertical
<b>Ventilación</b>	Ventilación mecánica controlada (VMC)
<b>Consumo</b>	Consumo de gas Consumo de agua Consumo de combustible Consumo de líquido Contador electrónico Energía eléctrica Intensidad Tensión Frecuencia eléctrica
<b>Sensores</b>	Anemómetro Humedad del aire Humedad del suelo Luminosidad Nivel de ruido Nivel de CO2 Precipitaciones Presión atmosférica Termómetro
<b>Detectores</b>	Rotura de cristal Cierre Fondo de puerta Fuga de gas Fuga de agua Fuga de combustible Fuga de líquido Humo/fuego Movimiento Apertura Presencia Caída Llamada de emergencia

Cree el equipo que desea y edite sus propiedades. Si no es de tipo genérico, se presentará por defecto una lista de acciones y/o retornos de estados, útil para el funcionamiento correcto del widget asociado. Todos sus comandos por defecto no se pueden eliminar del equipo.

Las propiedades de los equipos se agrupan en 2 pestañas: por un lado los comandos y por otro las variables.

## 4.1 Edición de acciones

Por defecto se llega a la lista de acciones definidas, con una indicación en cada una de ellas sobre el tipo de trama asociada (ASCII, JS o HEXA), así como botones de acción rápida (edición, eliminación y ejecución de la acción).

Los comandos creados por el usuario están encuadrados en amarillo y se pueden eliminar

Commandes	Variables
LED allumée / ASCII	 
LED éteinte / ASCII	 
Choix de la couleur RVB / JS	 
Read / ASCII	  
Commuter la LED	 
Variation de la LED en %	 
Positionner la valeur du blanc froid	 
Positionner la valeur du blanc chaud	 

-  Ejecución de una acción
-  Edición de una acción
-  Eliminar una acción (solo acciones creadas por el usuario)

LED allumée	
LED éteinte	
Variation de la LED en % 	<p><b>Label :</b> Variation de la LED en %</p> <hr/> <p><b>Type :</b>  <input type="radio"/> Ascil <input checked="" type="radio"/> JS</p> <p><b>Commande :</b></p> <pre>bri = parseInt({percentage} * 2.5, 10); command = '/api/Lifedomus1234/lights/13/state';</pre> <p><b>Lecture cyclique :</b> <input type="checkbox"/></p>
Choix de la couleur RVB	
Read	
Commuter la LED	
Positionner la valeur du blanc fr...	
Positionner la valeur du blanc c...	

Para cada acción, indique la etiqueta, el tipo de comando (ASCII, JS o HEXA), así como lo que el comando ha realizado. Si la acción tiene parámetros, puede ver sus nombres haciendo clic en el icono .

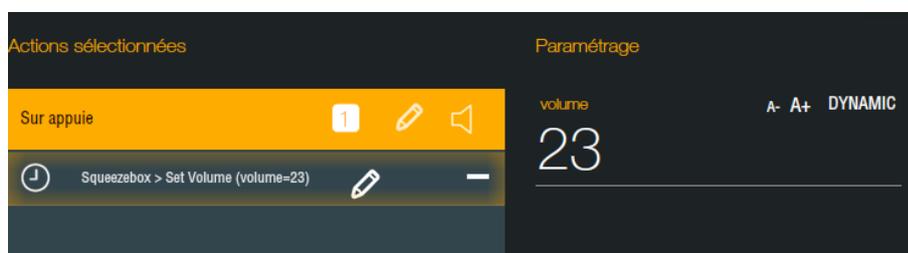
En el ejemplo anterior, la acción «Variación del LED» tiene un parámetro denominado «porcentaje», que va de 0 a 100. Reutilizo pues el valor del parámetro para convertirlo en un intervalo 0 – 255 que asigno a mi variable «bri». Por último, modifico el valor de la variable reservada «command» con la url a llamar.

También es posible enriquecer una acción de parámetros suplementarios, agregando en el comando parámetros entre llaves.

Ejemplo:



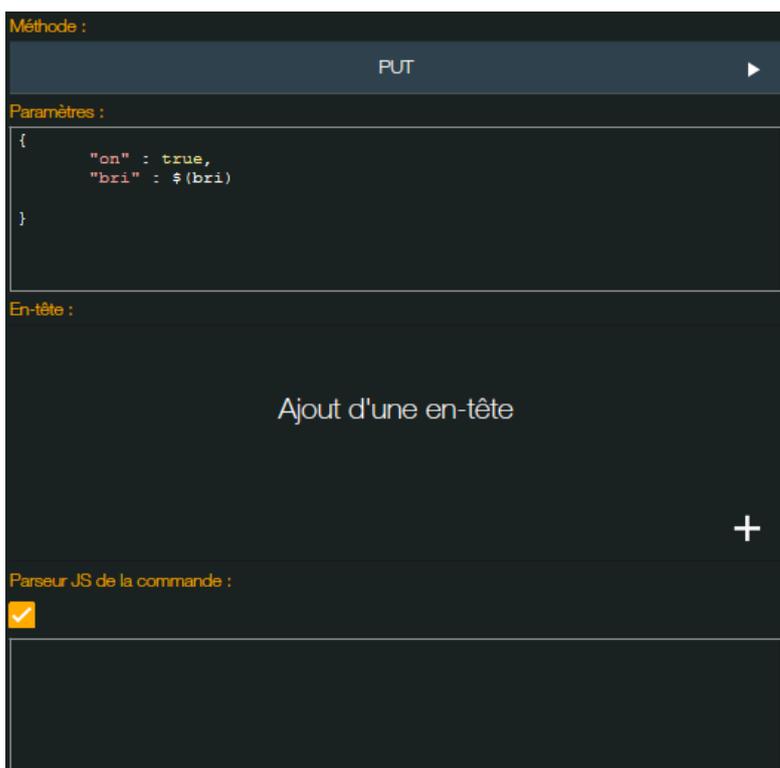
Aquí, {zone} hace referencia a una variable global, y no se interpreta por tanto como parámetro de acción, solo {volume} se usará como parámetro. En el lado del reproductor se encontrará con esto:



Por último, también se puede repetir un comando de forma cíclica (para por ejemplo actualizar el conjunto de retornos de estados).

El plazo de envío de los comandos cíclicos se especifica al nivel de las propiedades del conector.

#### 4.1.1 Propiedades específicas del conector HTTP



Si el equipo está asociado a un conector HTTP, hay que cumplimentar otros parámetros:

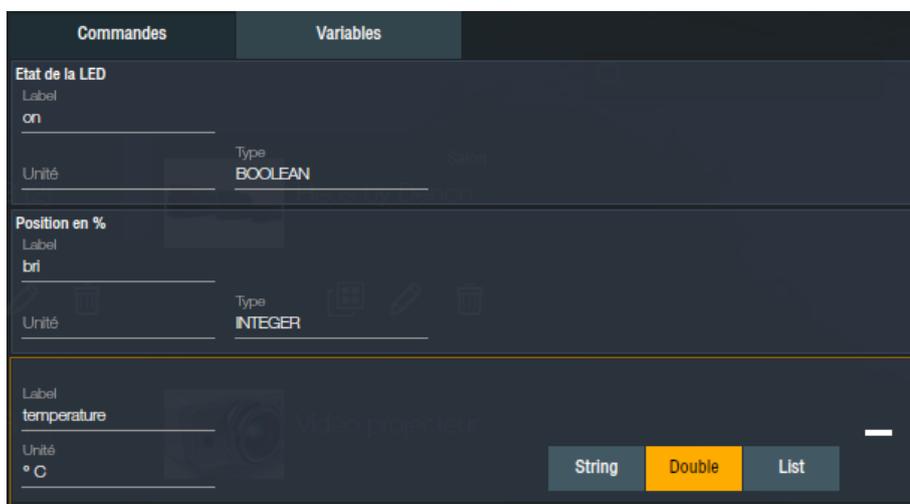
De igual modo que para las tramas de inicialización del conector, especifique el método HTTP de comunicación (GET, POST, PUT, DELETE). Si el método de envío es diferente de GET, puede indicar parámetros de control, con formato bruto (ej.: `on=true&value=1234`), en el formato JSON.

Fíjese aquí el uso de la variable «bri» en los parámetros JSON, recuperada del modo siguiente: `${bri}`

También puede indicar los encabezados, así como un analizador sintáctico específico asociado al comando.

Si no se define ningún analizador sintáctico de comandos, se llamará al analizador sintáctico del conector para evaluar la respuesta de la solicitud.

## 4.2 Edición de variables



En caso de un equipo no genérico, que tiene por defecto retornos de estados predefinidos, el tipo no es modificable. Puede vincular un retorno de estado a una variable JS especificando la etiqueta de la variable.

La unidad también puede modificarse (campo de entrada libre).

En el caso de una variable de usuario, hay que especificar de qué tipo es la variable (solo disponibles Cadena, Doble y Lista). El registro de una variable se realiza de forma automática al final de la introducción de un campo.

## 5 Widgets

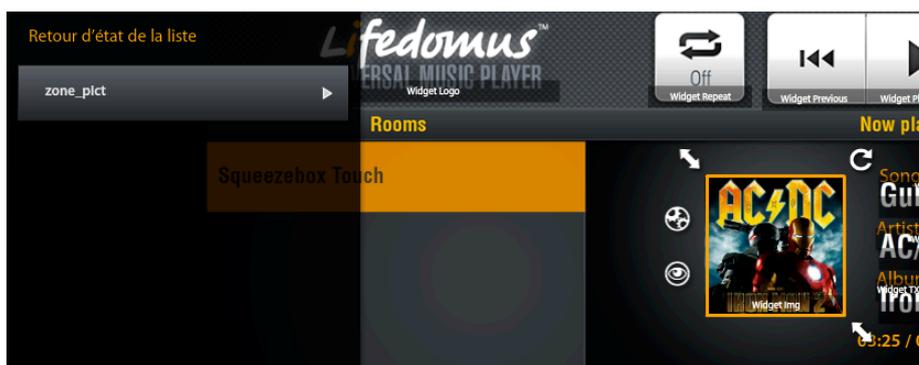
Para los equipos no genéricos, encontrará todos los widget predefinidos y utilizados para los demás protocolos.

Para los demás equipos, existen widgets dedicados al uso del modelo universal bidireccional:

### 5.1 URL

Permite crear un widget que tiene como retorno de estado una URL hacia una imagen.

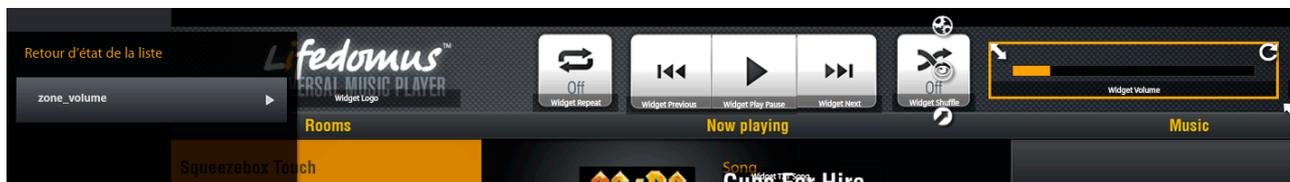
Ejemplo: una portada para el soporte de un equipo de Audio.



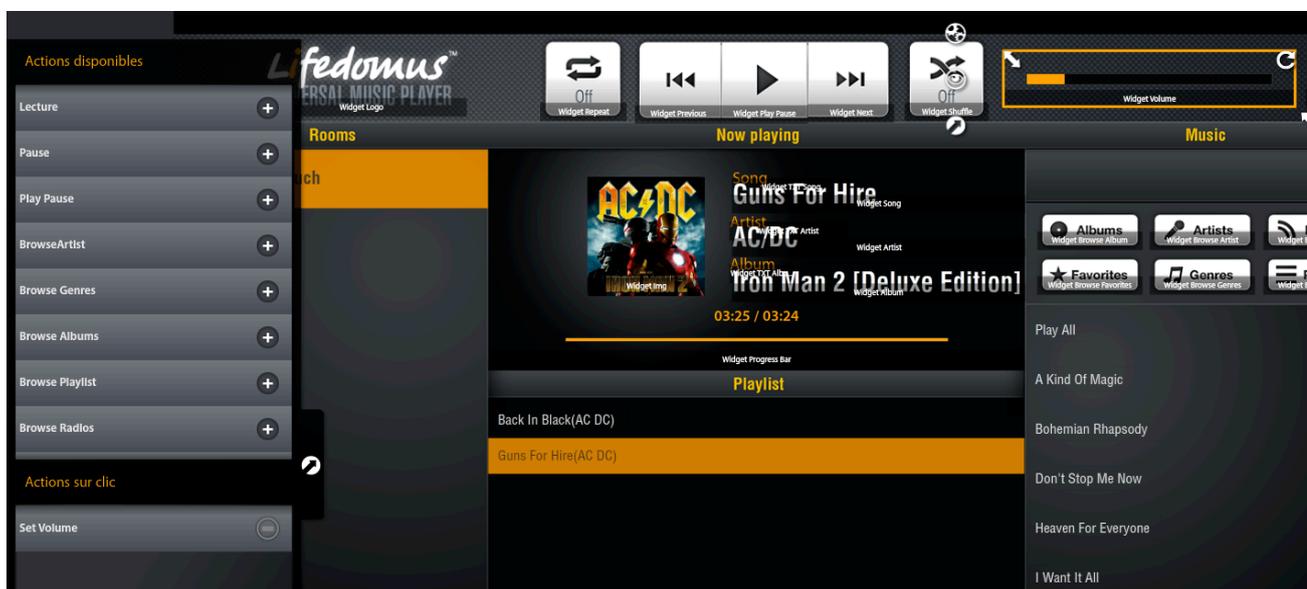
## 5.2 Sidebar

Permite crear una barra de progreso o de medida para gestionar por ejemplo el volumen, los graves, los agudos,...

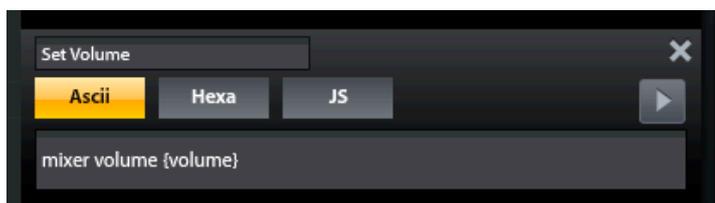
Este widget se asocia a un retorno de estado de tipo numérico en porcentaje



y una o varias acciones configuradas que sustituirán al parámetro del comando por el porcentaje introducido:



La acción SetVolume se declara como sigue en el configurador:

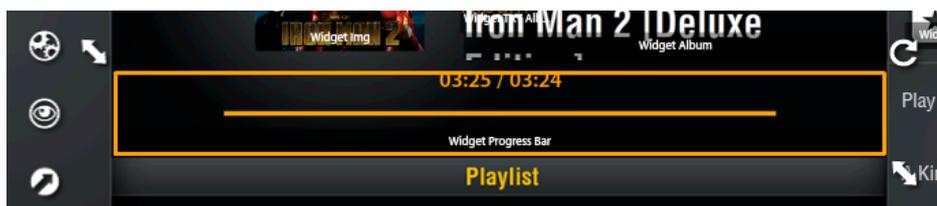


En este ejemplo, cuando el usuario pone la barra de progreso en el medio, (volumen) se sustituirá por 50 en el comando.

El comando enviado al módulo será: «mixer volume 50» (mezclador volumen 50).

## 5.3 TrackBar

Principio de funcionamiento idéntico al del widget SlideBar al que se asocian varios estados.



El primero es la duración de una canción de tipo "número" para conocer el tamaño máximo de la barra de progreso.

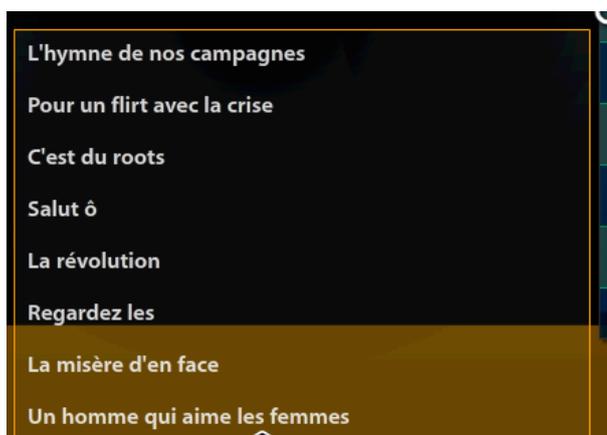
El segundo estado es la posición actual (en tiempo) de la canción, para sincronizar la evolución de la barra de progreso.

Si el módulo no devuelve automáticamente, cada segundo, el estado de la canción, puede poner un tercer estado en forma de número para incrementar automáticamente la barra:

- 0: parada
- 1: lectura
- 2: lectura X2

## 5.4 Liste

Widget lista al que solo se pueden asociar variables de tipo "lista" para la restauración de una lista (véase el punto 4, gestión del equipo). Haciendo clic en un elemento de la lista se puede asociar una acción que puede tener un parámetro que se sustituirá por la ID de parte de su lista:



```
var liste = new Array() ;
liste[0] = new Array() ;
liste[0][0] = '1' ;
liste[0][1] = 'L'hymne de nos campagnes' ;
liste[1] = new Array() ;
liste[1][0] = '2' ;
liste[1][1] = 'Pour un flirt avec la crise' ;
...
```

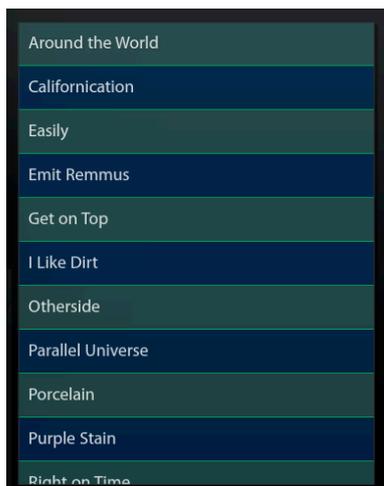
La acción asociada es la siguiente:



Del mismo modo que para el volumen, {id} será sustituido por el elemento 0 de su lista, aquí 1 o 2...

Esto da como trama enviada al módulo: «playlist index 2» (índice 2 lista de reproducción) si hago clic en «Pour un flirt avec la crise».

El elemento 0 puede ser un ID como en el ejemplo anterior, pero también una parte del comando enviado como en el ejemplo siguiente.



```
var liste = new Array() ;
liste[0] = new Array() ;
liste[0][0] = 'track_id :112';
liste[0][1] = 'Around the World' ;
liste[1] = new Array() ;
liste[1][0] = 'track_id:114';
liste[1][1] = 'Californication' ;
...
```

Para el comando siguiente:

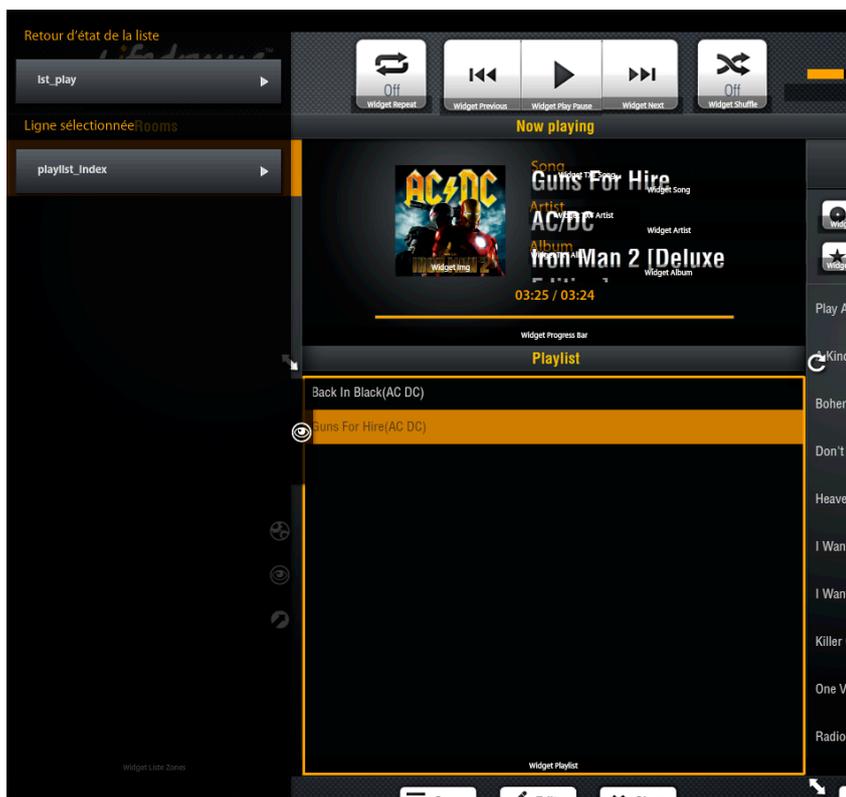


La trama enviada al módulo será: « playlistcontrol cmd:add track\_id:112 ».

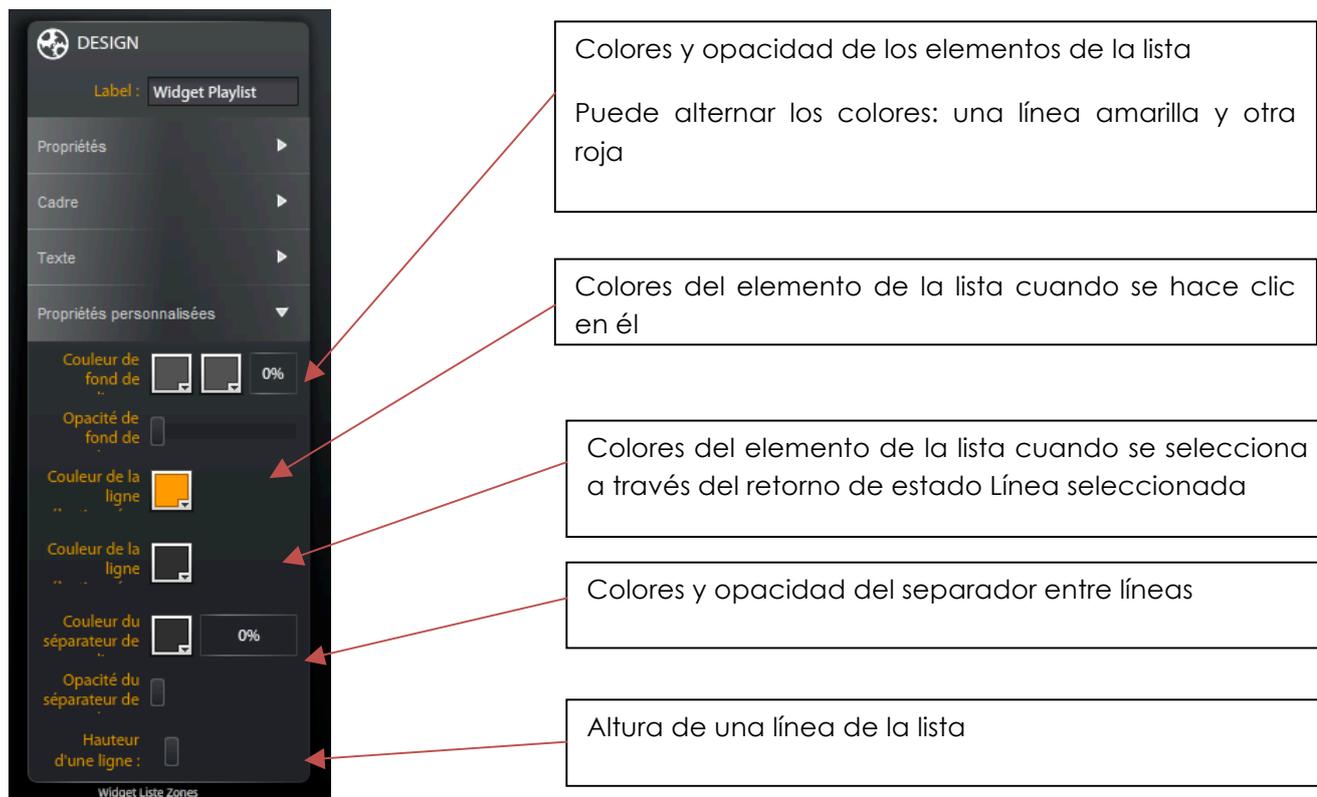
Resaltado de una línea de la lista:

Si quiere resaltar una línea de la tabla, puede asociar un retorno de estado en la propiedad "línea seleccionada". Este retorno de estado debe contener uno de los ID de la lista.

Ejemplo Lista de reproducción en curso:



El widget se puede configurar gráficamente en su totalidad a través del panel de propiedades:



The image shows a design panel for a 'Widget Playlist'. The panel includes sections for 'Propriétés', 'Cadre', 'Texte', and 'Propriétés personnalisées'. The 'Propriétés personnalisées' section contains several settings with callout boxes pointing to them:

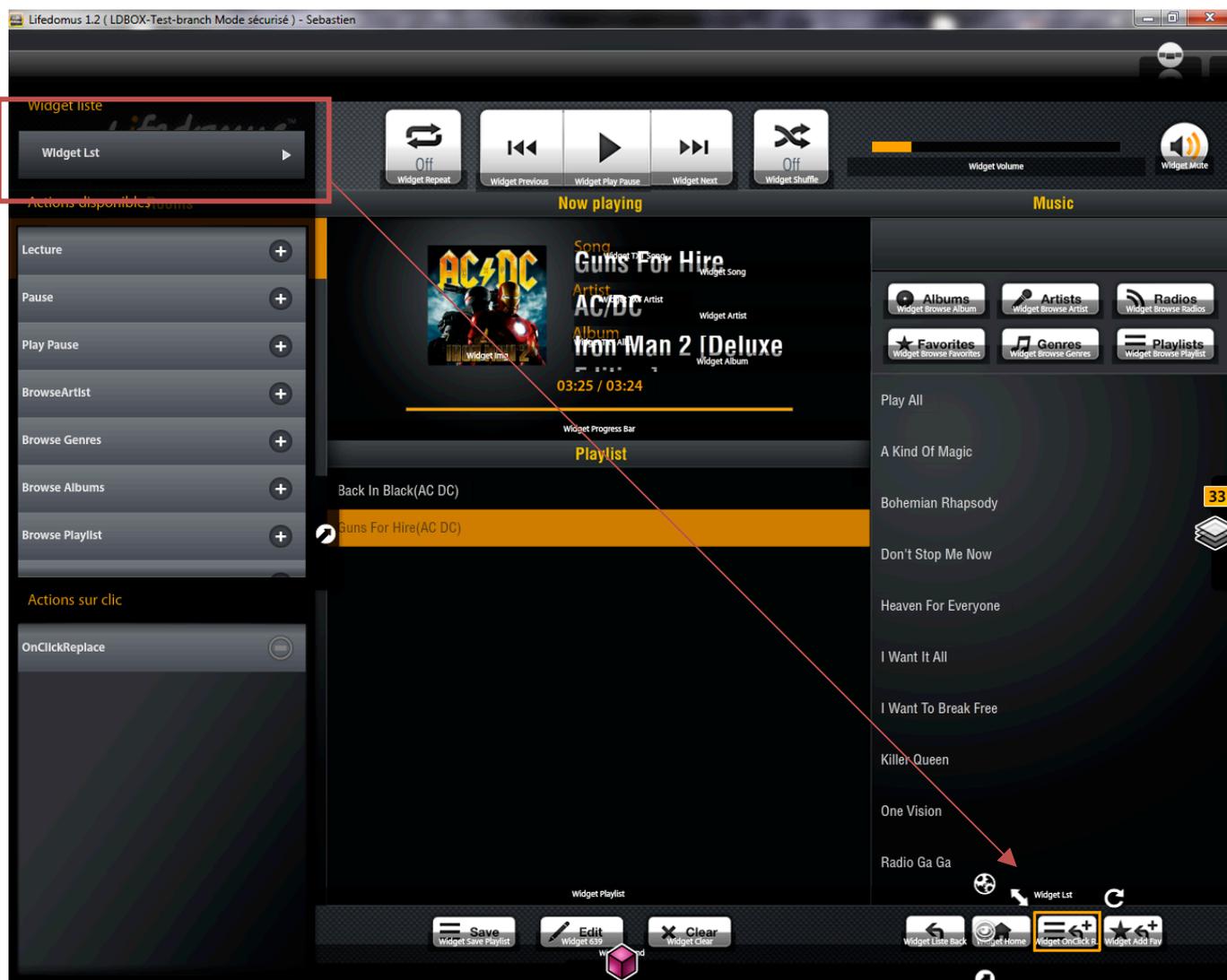
- Couleur de fond de ...**: Points to a callout box stating 'Colores y opacidad de los elementos de la lista' and 'Puede alternar los colores: una línea amarilla y otra roja'.
- Opacité de fond de ...**: Points to a callout box stating 'Colores del elemento de la lista cuando se hace clic en él'.
- Couleur de la ligne**: Points to a callout box stating 'Colores del elemento de la lista cuando se selecciona a través del retorno de estado Línea seleccionada'.
- Couleur de la ligne**: Points to a callout box stating 'Colores y opacidad del separador entre líneas'.
- Couleur du séparateur de ...**: Points to a callout box stating 'Colores y opacidad del separador entre líneas'.
- Opacité du séparateur de ...**: Points to a callout box stating 'Colores y opacidad del separador entre líneas'.
- Hauteur d'une ligne :**: Points to a callout box stating 'Altura de una línea de la lista'.

### Widget Liste action Switch

Este widget permite realizar otra acción en la lista que la prevista por defecto.

Por ejemplo, un clic en la lista ejecuta el comando asociado: comando de navegación en artistas, álbumes, título.

En cambio, un clic en el widget, ejecutará el comando asociado a este widget teniendo en cuenta el ID del elemento seleccionado en la lista. Por ejemplo, añada el artista, álbum o título a la lista de reproducción actual.



Simplemente asocie el widget action switch al widget liste con el que debería interactuar, y añada el(los) comando(s) a ejecutar (en el ejemplo anterior): OnClickReplace).