**ALTUI : Evolutions depuis Avril 2015**

Altui, alexis mermet, oct 2015

**AltUI** est un plugin VERA pour lui offrir une nouvelle user interface (qu’elle soit sous UI5 ou UI7 ). ALTUI tourne sur n’importe quel périphérique client ( PC MAC Tablette Phone ) et s’adapte automatiquement au format. Il est accessible en local mais aussi en mode remote à partir d’internet.

Cette application a déjà été présentée en début d’année dans les précédents articles

1. Introduction <http://www.domotique-info.fr/2015/04/altui-pour-vera-une-alternative-ui5-ou-ui7/>
2. Les Pages Custom : <http://www.touteladomotique.com/index.php?option=com_content&view=article&id=1444:altui-user-interface-alternative-pour-vera-custom-pages&catid=5:domotique&Itemid=89>
3. L’Extensibilité : <http://www.touteladomotique.com/index.php?option=com_content&view=article&id=1445:altuin-user-interface-alternative-pour-vera-extensions&catid=5:domotique&Itemid=89>

Mais l’application a beaucoup évolué depuis, il est donc utile de revenir sur ces dernières évolutions pas toujours très connues, entre autre :

* La Localisation
* Les dashboards des modules
* Le mode multi contrôleur
* La page principale et les favoris
* Les scènes et l’éditeur de scène
* La page des plugins
* La table des évènements
* La tables des périphériques
* Les graphiques
* Les pages customs
* Le Menu Divers ( Misc )
* Les options
* La sélection des thèmes bootstrap sur bootswatch
* La page debug

# Les fondamentaux

Les fondamentaux restent les mêmes.

ALTUI reste gratuit sur le principe de la donation volontaire par Paypal. Cela reste un développement bénévole (même si on parle centaines d’heures et que aujourd’hui le niveau de complexité est important) réalisé en dehors d’un contexte professionnel donc toute donation est plus que bienvenue mais pas exigée. Les sources sont disponibles mais bien sûr, par pour un usage commercial sans autorisation.

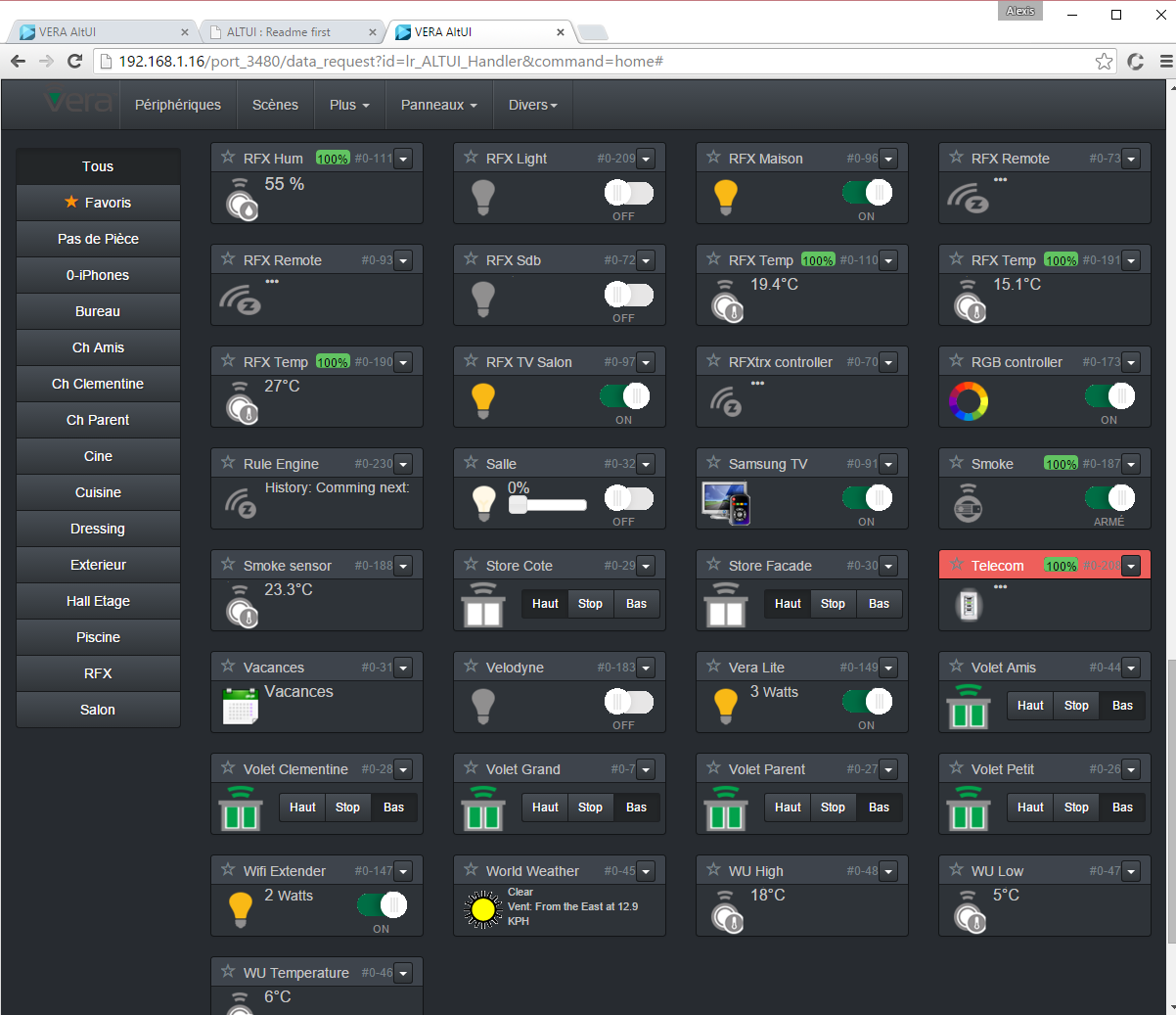
AU niveau installation, la règle reste que seul un simple plugin VERA est nécessaire et tout est quasiment fait en sur le client local, sous la forme d’une application web/javascript a page unique. Il y a maintenant quelques exceptions a cette règle que l’on verra ci-dessous et en particulier

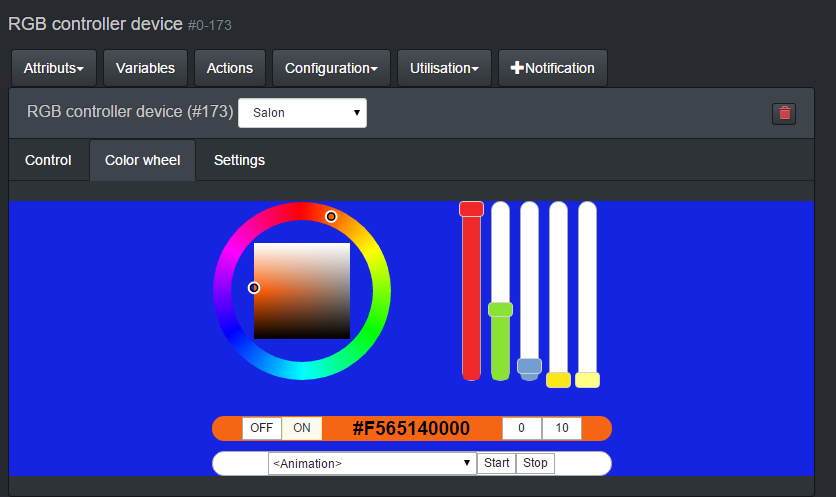
1. le device ALTUI effectue le suivi de certaines variables de module, telles que demandes par l’utilisateur et peux déclencher des scènes donc l’utilisateur n’est plus limite aux « events » tel que définis par VERA ou par les auteurs de plugin
2. de même le device ALTUI s’intègre avec thingspeak et peut envoyer la valeur d’une variable qui change sur un channel thingspeak.

La plupart des plugins UI5 UI7 fonctionnent sur ALTUI mais la compatibilité ne peut pas être a 100% étant donné que ALTUI n’utilise que bootstrap & jquery et que bien des plugins font des mélanges avec d’autres framework javascripts que UI5 utilisait par exemple. Mais il est possible d’écrire un plugin spécifiquement pour ALTUI et l’environnement de programmation est alors beaucoup plus riche que sur UI5 UI7.

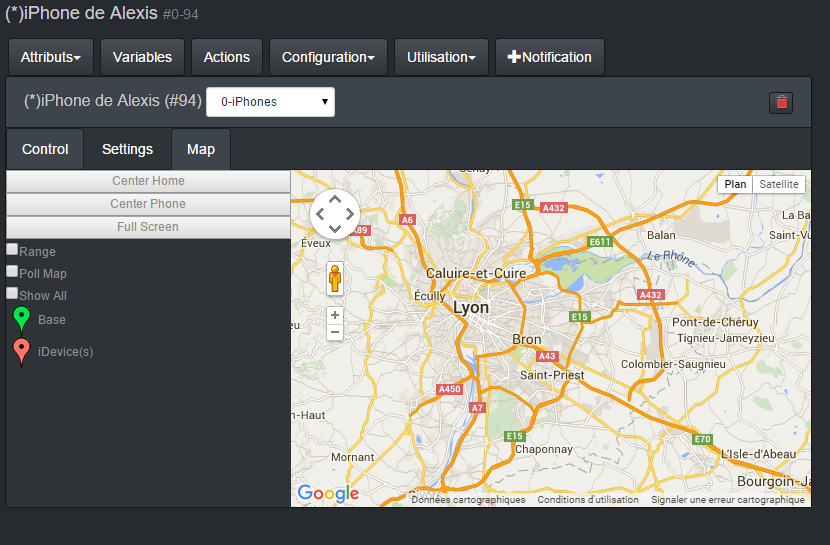
# Pourquoi ALTUI ou ce que fait ALTUI que UI5 UI7 ne fait pas

* afficher les icones, même en mode remote
* afficher les thumbnail camera en video stream
* supporter la persistance des données de variable par le cloud thingspeak
* supporter les changements de variable et des expressions booléennes comme trigger des scènes
* permettre à l’utilisateur de faire ses propres pages composites a base d’éléments de modules
* permettre d’exécuter des commandes à distance sur la VERA , de tester du code LUA
* afficher des tables et des graphiques du réseau de module
* se mettre à jour automatiquement
* permettre à l’utilisateur de choisir son « skin »
* être affichable sur toute taille d’écran sans autre application a charger

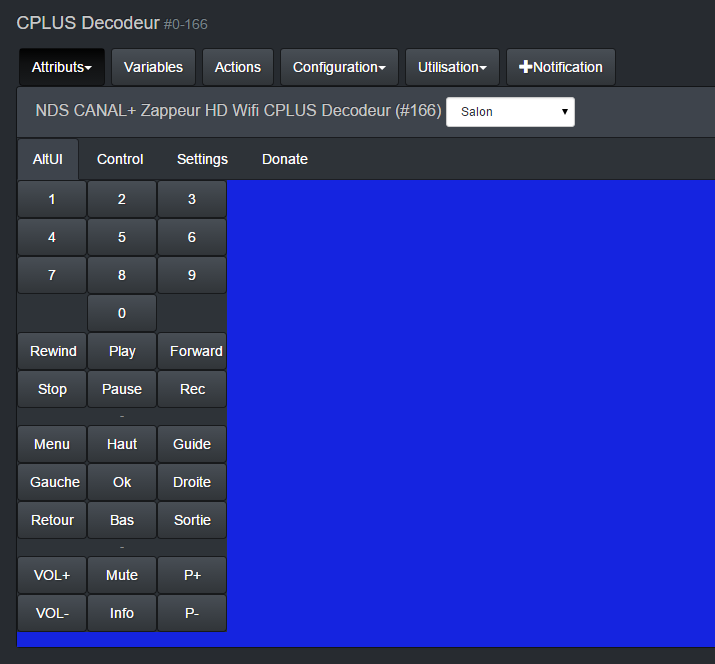


Exemple de Travail de vosmont auteur du plugin RGBW sur Vera:

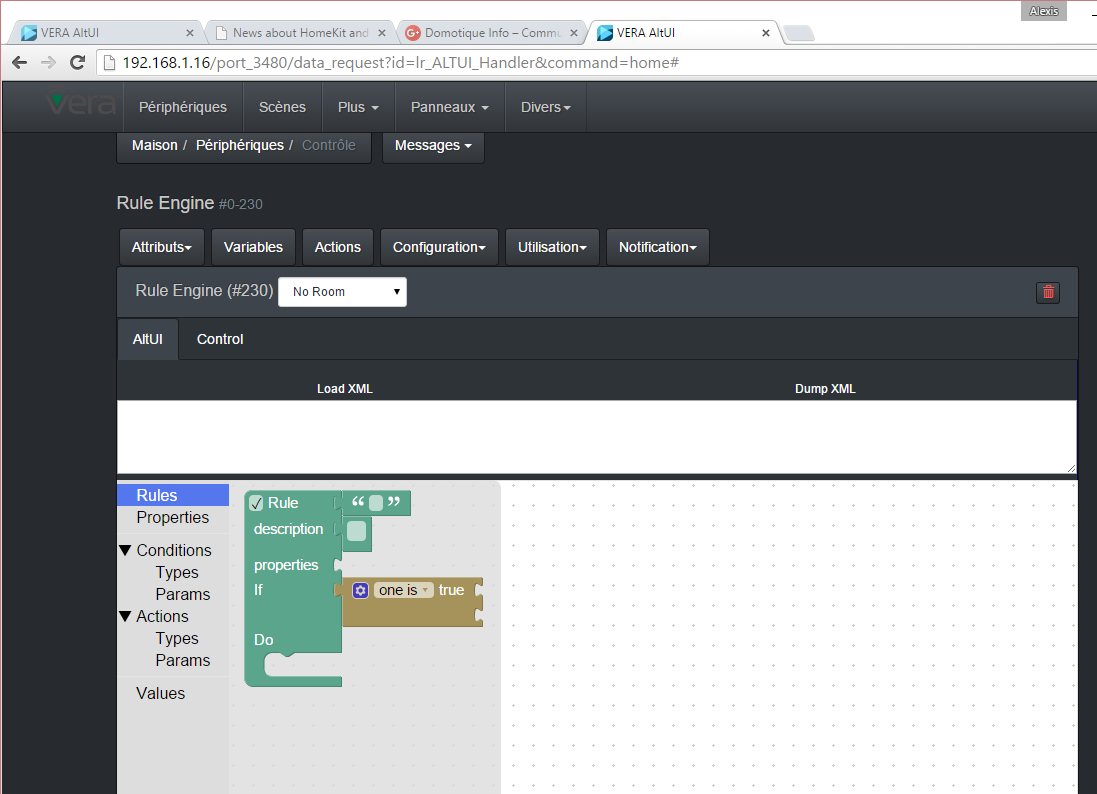
Exemple sur IPhoneLocator



Il est aussi possible de beneficier entierement de l’api de ALTUI et de javascript pour offrer des user interfaces de plugin qui sont plus complexes et spécifiques a ALTUI. Par exemple celle du plugin CPLUS



Ou celle de @vosmont pour son moteur de règle dont la user interface est à base de Google Blockly



# La page d’accueil

## L’auto-update

Altui étant base sur le développement « agile » les versions sont fréquentes et même si une version officielle est toujours sur le store MCV, les versions intermédiaires, lorsque je les publie, sont proposées a l’utilisateur, et installées automatiquement s’il l’accepte.

## L’icône de départ sur IOS

il est possible, une fois que la page ALTUI est ouverte, de sauver la page sur le bureau de IOS afin que cela devienne une application comme les autres, sans la barre de navigation de Safari IOS, le ressenti est alors celui d’une application native, car ALTUI est essentiellement exécuté sur le poste client et de fait est très rapide.

## La Localisation

ALTUI est complètement localisable et est fourni avec l’anglais et le français. Un fichier italien avait été commencé par un internaute mais n’a pas été terminé. Pour rajouter une langue il suffit de copier J\_ALTUI\_loc\_en et de rajouter un nouveau fichier J\_ALTUI\_loc\_xx ou xx sont les 2 lettres standard données par le navigateur. Tout n’est pas fini dans ce domaine mais la plus grosse partie est couverte

La langue sélectionnée sera par défaut celle du navigateur utilise, cependant il est possible de forcer une langue en passant lang=xx sur l’url d’ouverture. Exemple

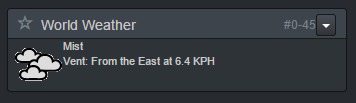
http://192.168.1.16/port\_3480/data\_request?id=lr\_ALTUI\_Handler&command=home&lang=en

# Les dashboards des modules

ALTUI présente un dashboard par default pour tous les modules, mais grâce à certaines contributions d’internaute, le module des types suivants bénéficient d’un display spécifique

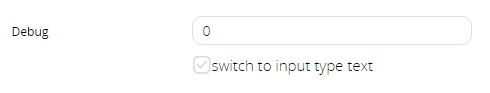
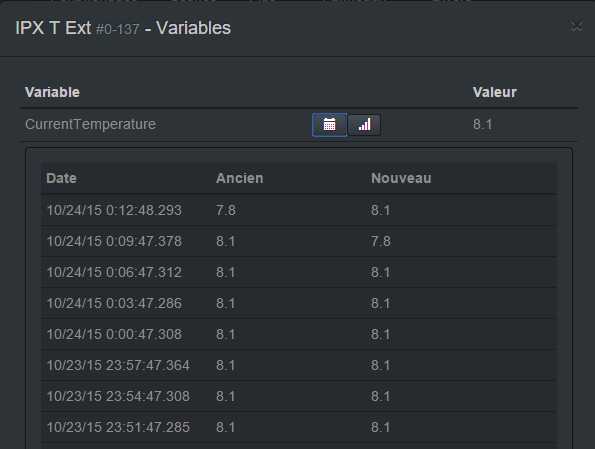
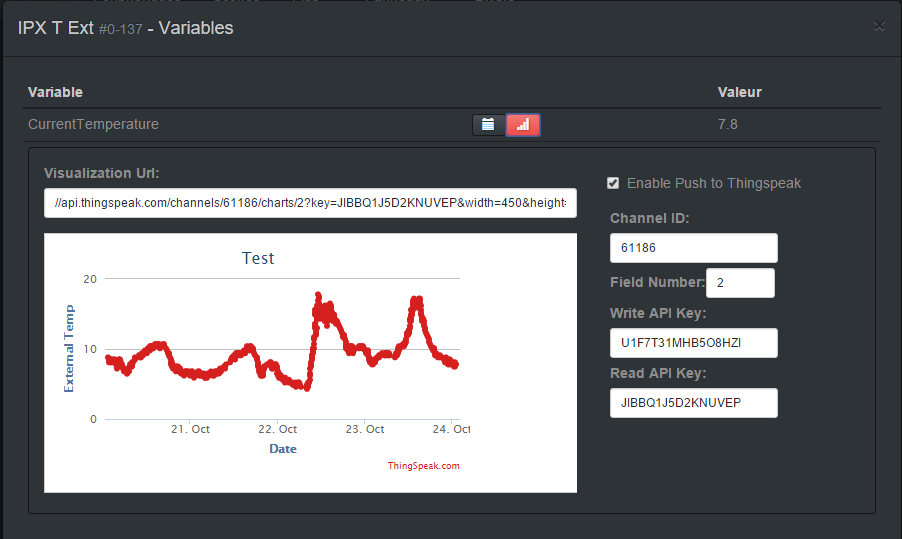
* urn:schemas-upnp-org:device:BinaryLight:1
* urn:schemas-upnp-org:device:RGBController:1
* urn:antor-fr:device:SamsungTVRemote:1
* urn:schemas-micasaverde-com:device:DoorLock:1
* urn:schemas-micasaverde-com:device:DoorSensor:1
* urn:schemas-micasaverde-com:device:TemperatureSensor:1
* urn:schemas-upnp-org:device:Heater:1
* urn:schemas-upnp-org:device:HVAC\_ZoneThermostat:1
* urn:schemas-micasaverde-com:device:HumiditySensor:1
* urn:schemas-micasaverde-com:device:LightSensor:1
* urn:schemas-cd-jackson-com:device:DataMine:1
* urn:schemas-a-lurker-com:device:InfoViewer:1
* urn:demo-micasaverde-com:device:weather:1
* urn:schemas-upnp-org:device:DimmableLight:1
* urn:schemas-micasaverde-com:device:MotionSensor:1
* urn:schemas-micasaverde-com:device:SmokeSensor:1
* urn:schemas-micasaverde-com:device:WindowCovering:1
* urn:schemas-micasaverde-com:device:PowerMeter:1
* urn:schemas-micasaverde-com:device:PowerMeter:2
* urn:schemas-upnp-org:device:VSwitch:1
* urn:schemas-upnp-org:device:DigitalSecurityCamera:1
* urn:schemas-upnp-org:device:DigitalSecurityCamera:2
* urn:schemas-upnp-org:device:cplus:1
* urn:schemas-upnp-org:device:altui:1
* urn:schemas-futzle-com:device:holidayvirtualswitch:1
* urn:schemas-futzle-com:device:CountdownTimer:1
* urn:schemas-upnp-org:device:IPhoneLocator:1
* urn:schemas-upnp-org:device:IPX800:1
* urn:schemas-rts-services-com:device:ProgramLogicEG:1
* urn:schemas-utz-com:device:GCal:1
* urn:schemas-futzle-com:device:CombinationSwitch:1
* urn:schemas-rts-services-com:device:DayTime:1
* urn:schemas-micasaverde-com:device:Sonos:1
* urn:schemas-cd-jackson-com:device:SystemMonitor:1
* urn:richardgreen:device:VeraAlert:1
* urn:schemas-micasaverde-com:device:TempLeakSensor:1
* urn:schemas-upnp-org:device:VContainer:1
* urn:schemas-futzle-com:device:UPnPProxy:1
* urn:schemas-rts-services-com:device:ProgramLogicTS:1
* urn:schemas-arduino-cc:device:arduino:1
* urn:schemas-dcineco-com:device:MSwitch:1

le champ des possibilités est large car il est possible de rajouter des plugins en javascript que ALTUI charge dynamiquement. Exemple de rendu sur le weather plugin

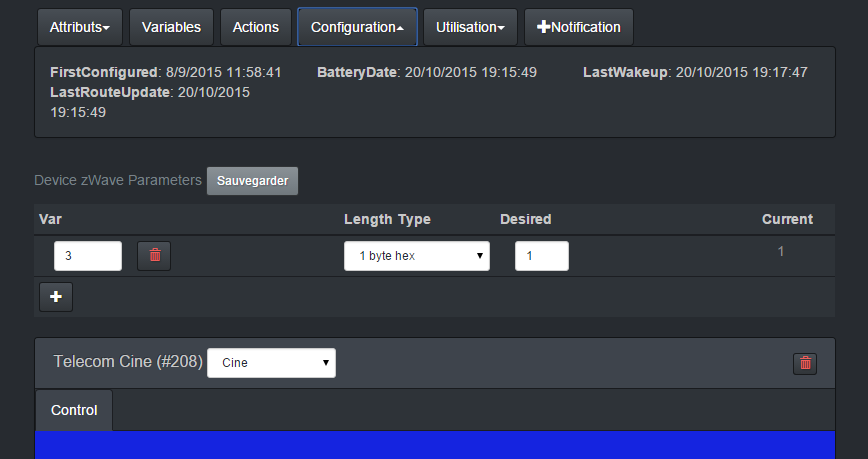


# Les variables et l’historique des changements

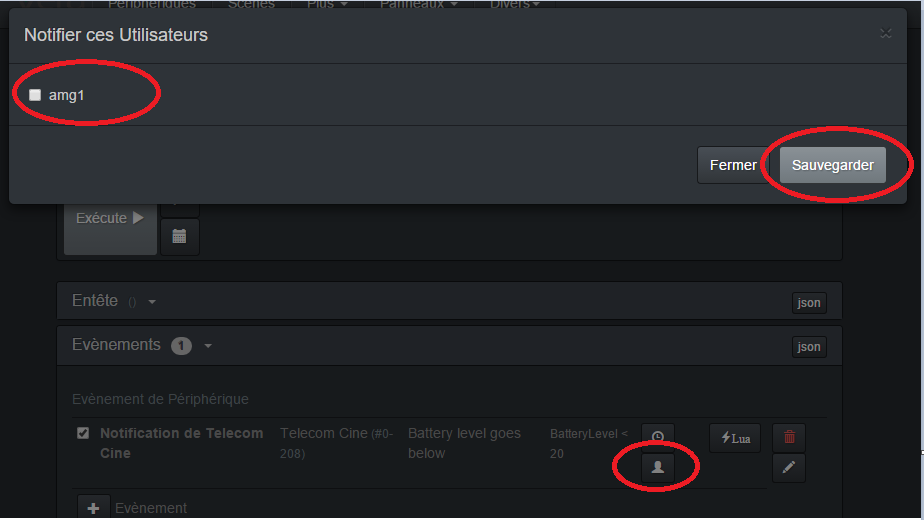
Toutes les variables d’un module sont visibles dans la boite de dialogue variable mais ALTUI apporte 3 améliorations par rapport a UI7

1. on peut cliquer sur une variable et rentrer la valeur même avec des caractères spéciaux HTML ou JSON sans avoir à jouer avec l’option discutable de UI7 (switch to input type text ). Je doute que les utilisateurs soient familiers avec la différence entre un <div> et un <input> et les implications sur l’encodage html des textes présentés. Pourtant MCV vous le demande !  
   in UI7 :
2. On peut directement chercher dans les logs l’historique des changements sur la variable ( mais seulement pour la période de temps couverte par les logs )  
   
3. Mieux encore, en cliquant sur la petite icone (bar chart ) , on peut paramétrer un suivi sur thingspeak pour cette variable. ALTUI s’occupera d’envoyer à thingspeak chaque changement de valeur sur votre channel/field thingspeak automatiquement ( après un reload LUA néanmoins pour laisser a ALTUI la chance d’enregistrer un variable watch sur cette variable ).   
     
     
   Aucune base de données locale, clé USB ou autre support de stockage n’est nécessaire et puisque thingspeak permet des analyses avec MATLAB et de faire des graphiques, on peut aussi afficher le graphique MATLAB en changeant l’url de visualisation par défaut. 

Les modules zWaves peuvent être configurés avec des variables zWave définies par le constructeur du module, ALTUI permet d’éditer ces variables par l’intermédiaire du buttons configuration sur l’écran control panel de ce device. Par exemple ci-dessous comment mettre la télécommande en mode Scène Activation afin d’avoir les 16 buttons possibles sur la Vera

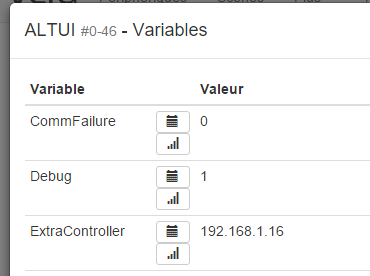


Enfin les notifications sont aussi configurables maintenant par le bouton **Notifications** et la sélection des utilisateurs à notifier dans l’éditeur de scène

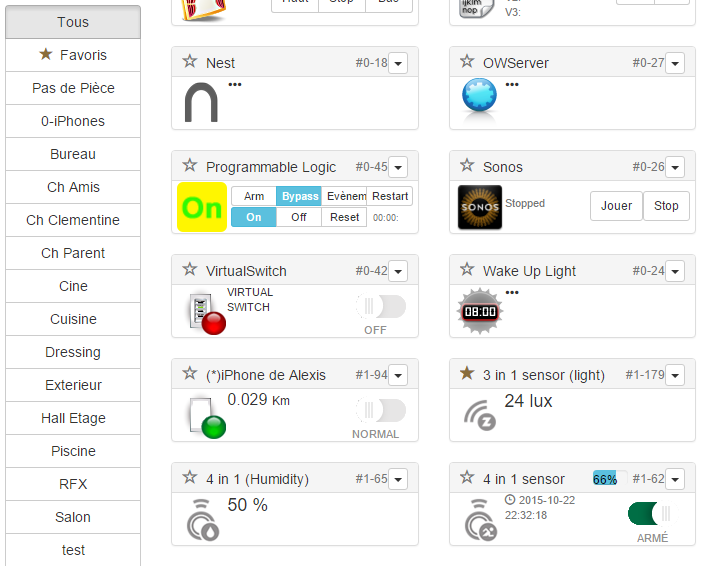


# Le mode multi contrôleur

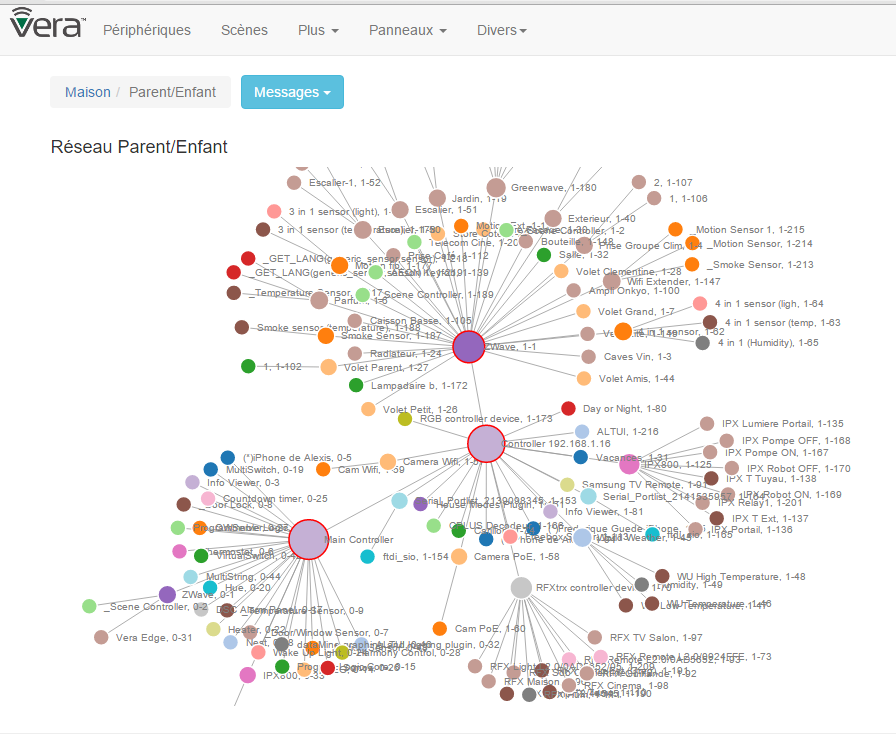
ALTUI, grâce à sa variable ‘**ExtraController**’ permet de gérer à partir de la même interface plusieurs vera. En théorie plus mais à partir de 3 la performance commence à s’en ressentir.



Une fois ceci fait ALTUI va afficher les périphériques de la VERA de base et de la VERA secondaire au sein de la même page. Remarquer les « ID » des modules qui commencent soit par 0- pour la vera principale et 1- pour la vera secondaire.

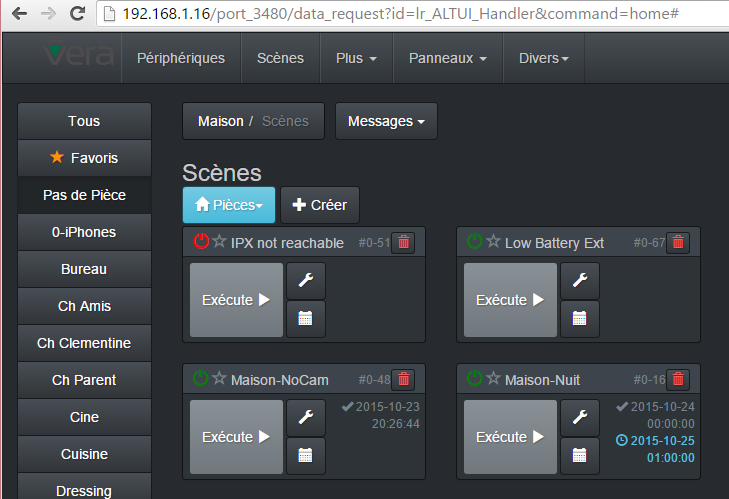


Le diagramme de relation parent/enfant affiche bien la vera de base et la vera secondaire ainsi que leurs modules respectifs. Diagramme a base de D3js et du forcelayout qui simule les particules électriques et forces répulsives ce qui donne une certaine dynamique au résultat…

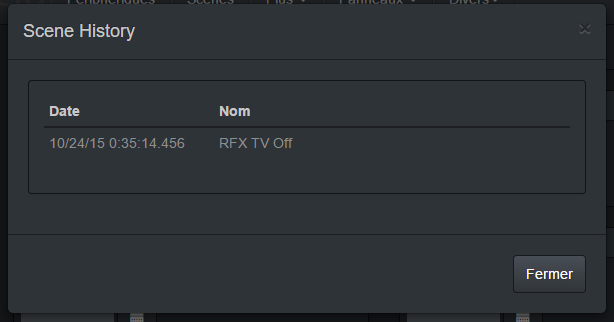


# Les scènes et l’éditeur de scène

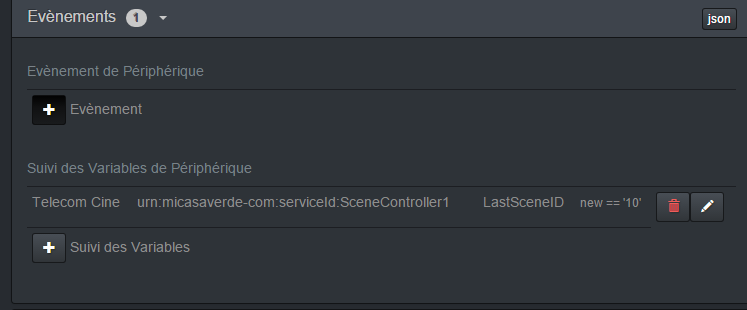
Les scènes aussi ont eu droit à certaines évolutions, entre autre, la possibilité de mettre une scène en pause en cliquant sur le petit signe On Off de la scène

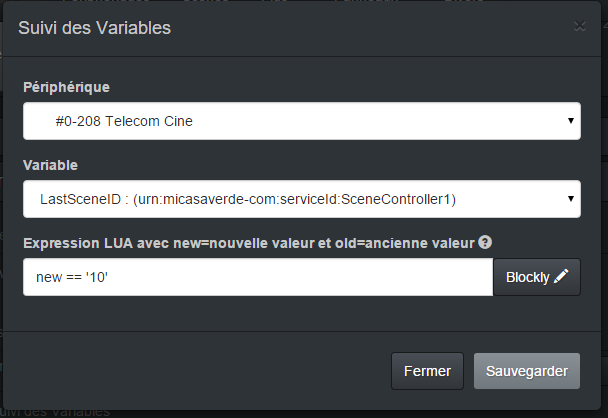


De même il est maintenant possible de voir l’historique récente de l’exécution de la scène ( si et seulement si elle est toujours dans les log )



Mais le gros changement est la possibilité de déclencher une scène sur le changement de valeur d’une variable d’un module, a associe a une condition précise grâce a une condition exprimée en code lua. Ainsi la vera a un problème avec la télécommande NODON octan et il faut faire du code lua pour s’en sortir afin de faire un variable\_watch sur la variable LastSceneID. Avec ALTUI on peut s’en sortir sans code.

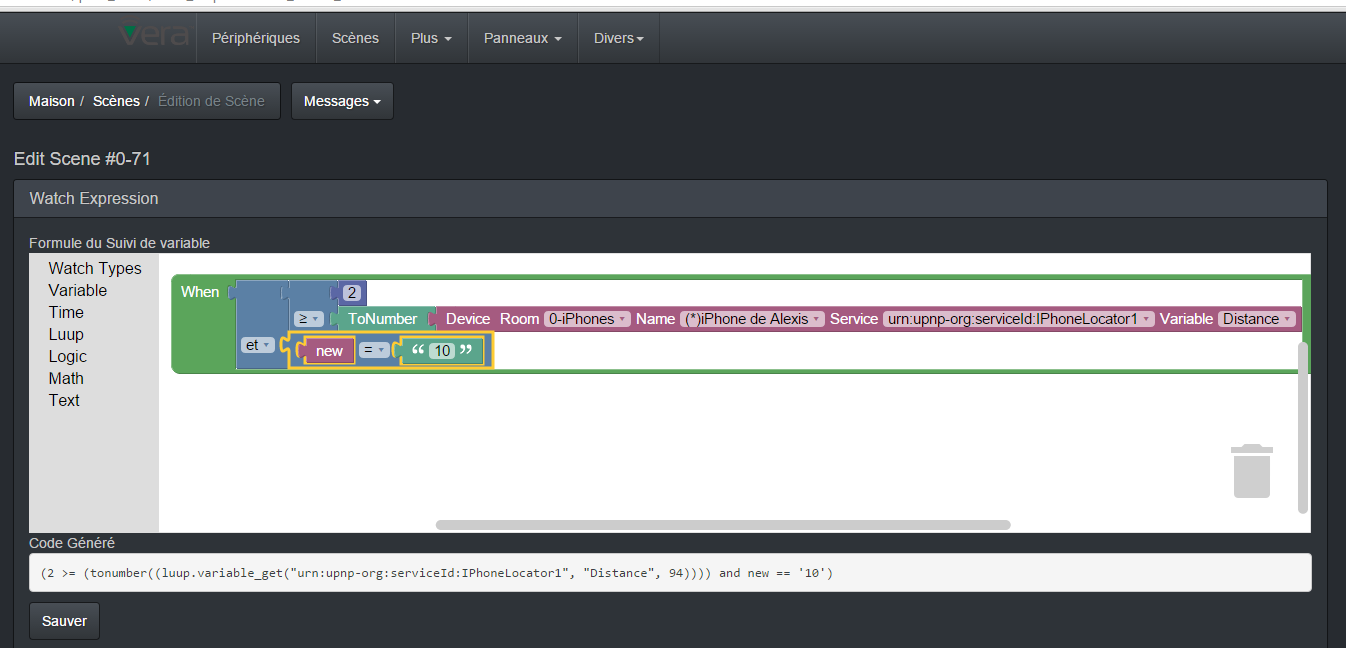




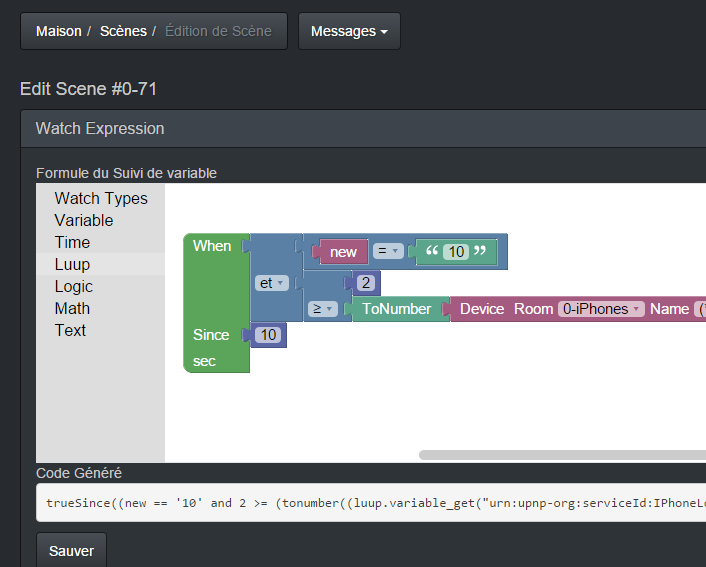
Une fois le suivi de variable déclaré ( et après un restart Luup afin de donner une chance à ALTUI d’enregistrer le suivi de variable ), tout est géré automatiquement par le plugin ALTUI sur la VERA.

La variable doit changer et la condition indiquée doivent être vraie pour que la scène soit lancée. **new** et **old** étant des mots clés réservés pour donner la nouvelle et l’ancienne valeur de la variable. Ainsi des expressions comme **new**<**old** sont possibles.

Et pour les plus réticent à écrire des expressions en Lua, il est possible d’utiliser l’éditeur Blockly de google pour avoir une interface graphique pour programmer ces conditions. Exemple



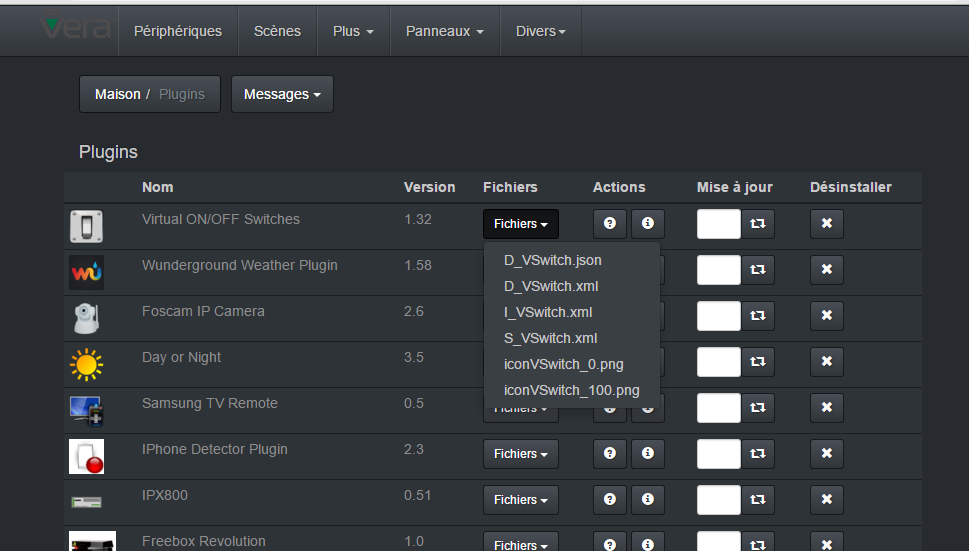
Il est possible de déclarer des évaluations différées dans le temps, ainsi en utiliser la fonction trueSince() la condition suivante devra être vraie **pendant** toutes les 10 secondes pour que la scène soit lancée

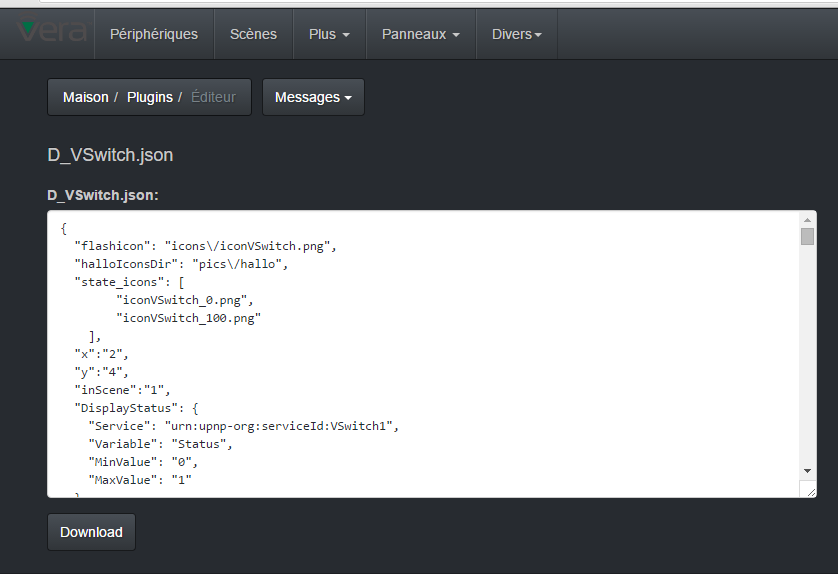


Voilà de quoi donner un peu plus de puissance a notre VERA…

# La page des plugins

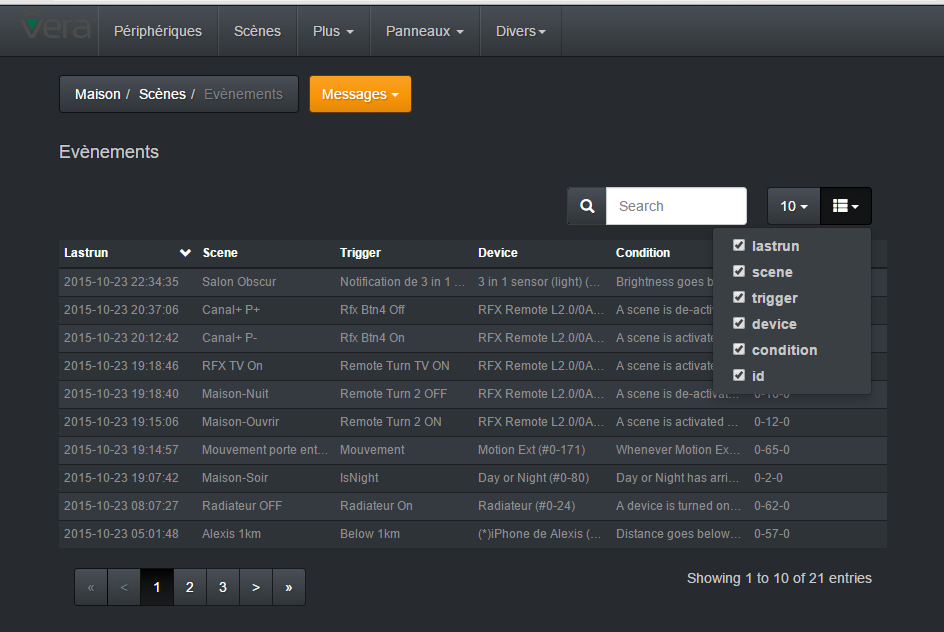
Il est possible de voir les plugins installés, leur version, et leur fichier dans un éditeur texte, ainsi que de déclencher l’installation d’une version spécifique, un upgrade a la dernière version ou la désinstallation





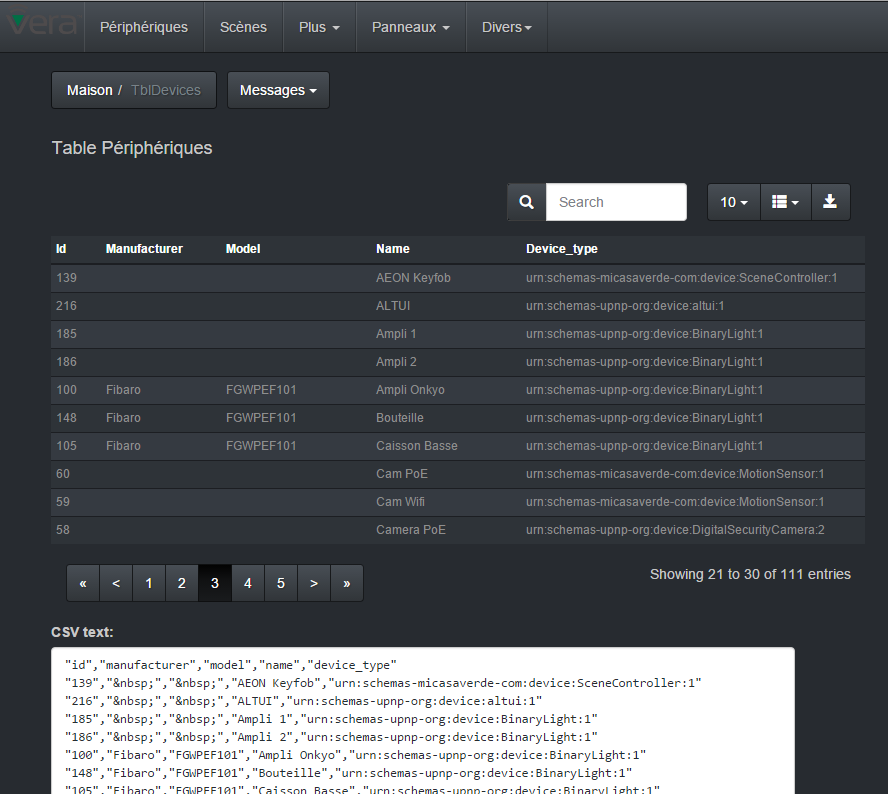
# La table des évènements

Les évènements utilisés dans les scènes peuvent être visible dans une table qui est ordonnable en cliquant sur les colonnes. On peut aussi choisir quelles colonnes on veut afficher



# La tables des périphériques

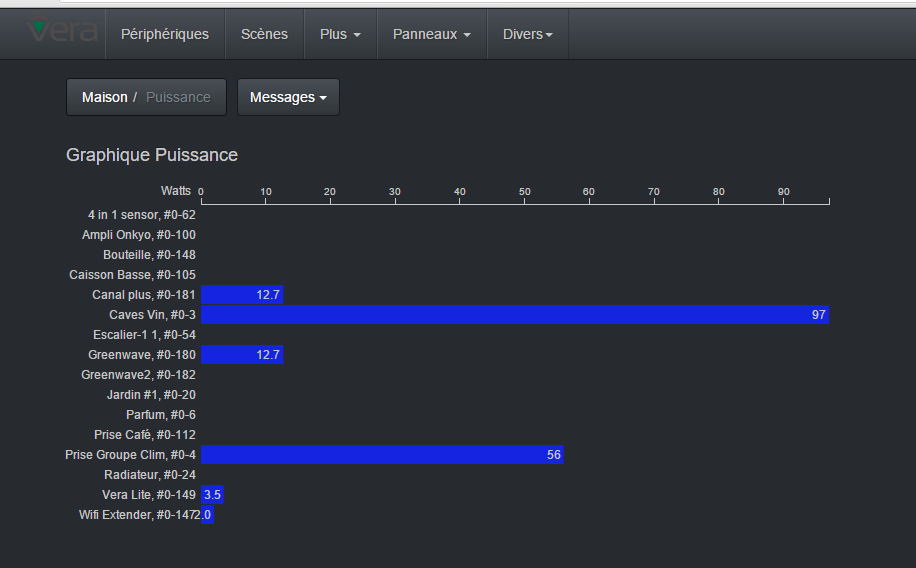
Il en va de même pour la table des périphériques qui est aussi exportable dans le clipboard



# Les graphiques

Certains graphiques animes sont fournis comme

* Puissance
* Les relations Parent / fils entre les périphériques (déjà vu ci-dessus )
* Le réseau et les routes zwave
* Un résume de la qualité des routes

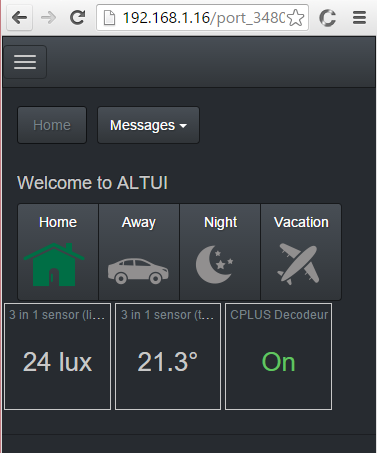


# La page principale et les favoris

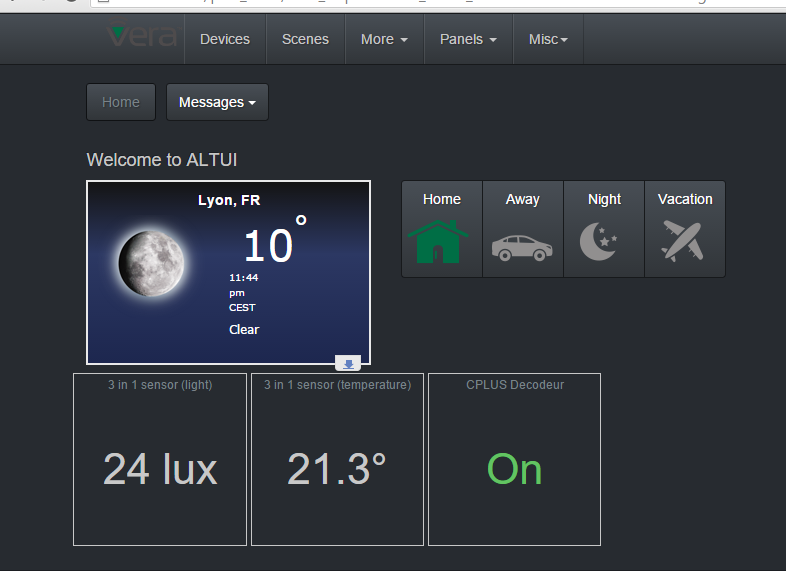
Par default la page principale de ALTUI affiche le menu, un widget météo ( dont l’affichage est optionnel et contrôlable par les options de ALTUI ) , le mode « house » si la Vera tourne UI7 ainsi que des tuiles pour les scènes ou les modules qui ont été marques comme favoris par l’utilisateur.

La page principale peut être choisie par le biais d’un paramètre sur l’url utilisée pour ouvrir altui. Il suffit de rajouter &**home**=xxx ou xx est l’une des valeurs parmis (pageHome , pageRooms , pageDevices , pageScenes , pageSceneEdit , pagePlugins , pageUsePages , pageEditPages , pageCredits , pageLuaTest , pageLuaStart , pageOptions , pageEditor , pageZwave , pageLocalization , pagePower , pageChildren , pageRoutes , pageQuality , pageTblDevices , pageOsCommand)

Ecran téléphone :

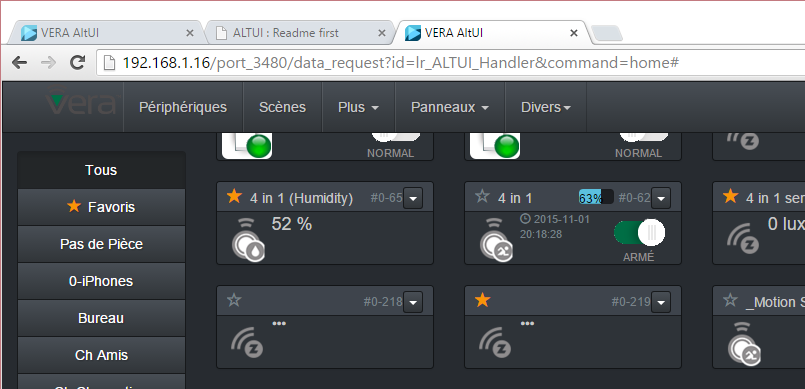


Ecran ordinateur ou Tablette

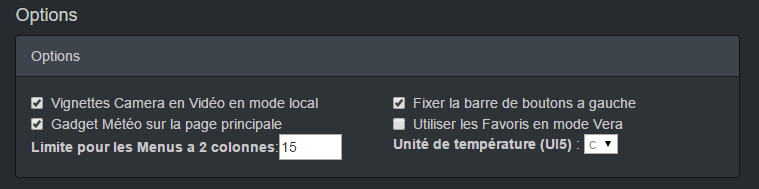


Cliquer sur un favoris scène lancera l’exécution de la scène, cliquer sur un favoris module lancera une action par défaut ( comme allumer/éteindre un binary switch )

Pour choisir un favori il suffit simplement de cliquer sur la petite étoile dans le coin supérieur gauche du module.

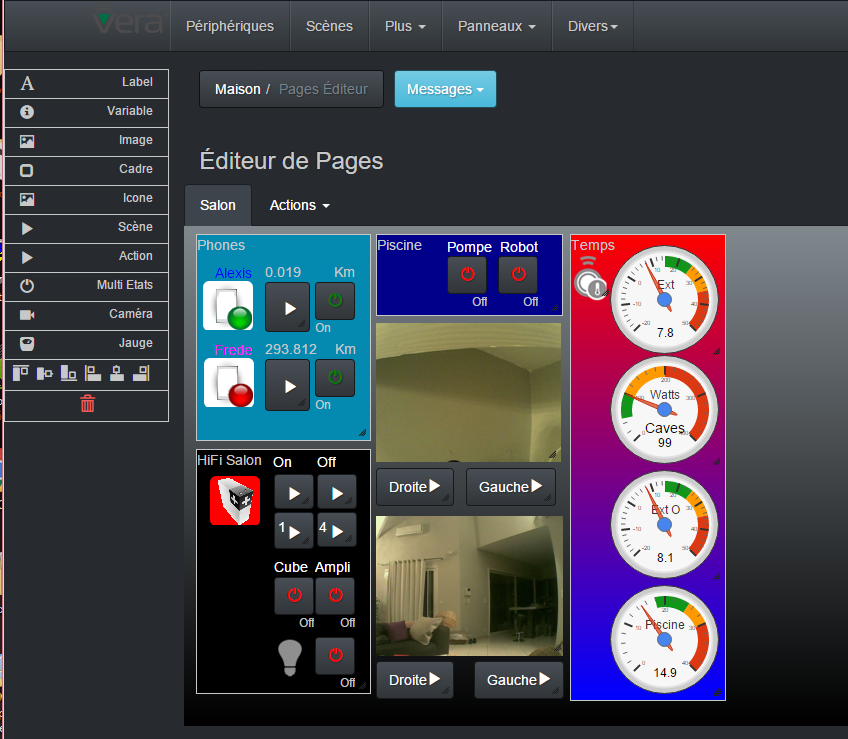


Par default, les favoris de ALTUI ne sont pas les mêmes que ceux de UI5 UI7. Chacun des ordinateurs clients utilisés pour accéder a ALTUI possèdera sa propre liste de favoris, ainsi l’ipad qui traine sur la table du salon pourra avoir des favoris différents de ceux de votre téléphone préféré. Par contre certains utilisateurs préfèreront utiliser les mêmes favoris que ceux de UI5 UI7, ce qui est possible grâce a une option (utiliser les favoris en mode vera )



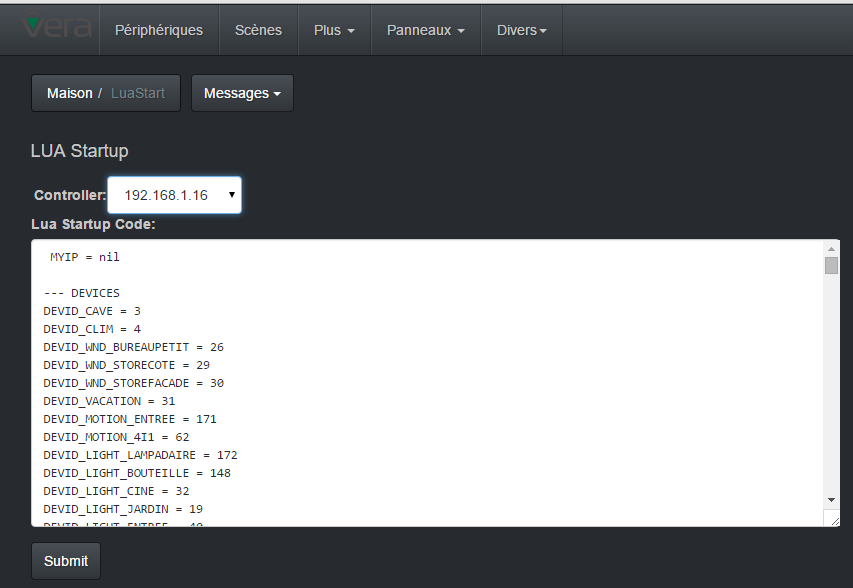
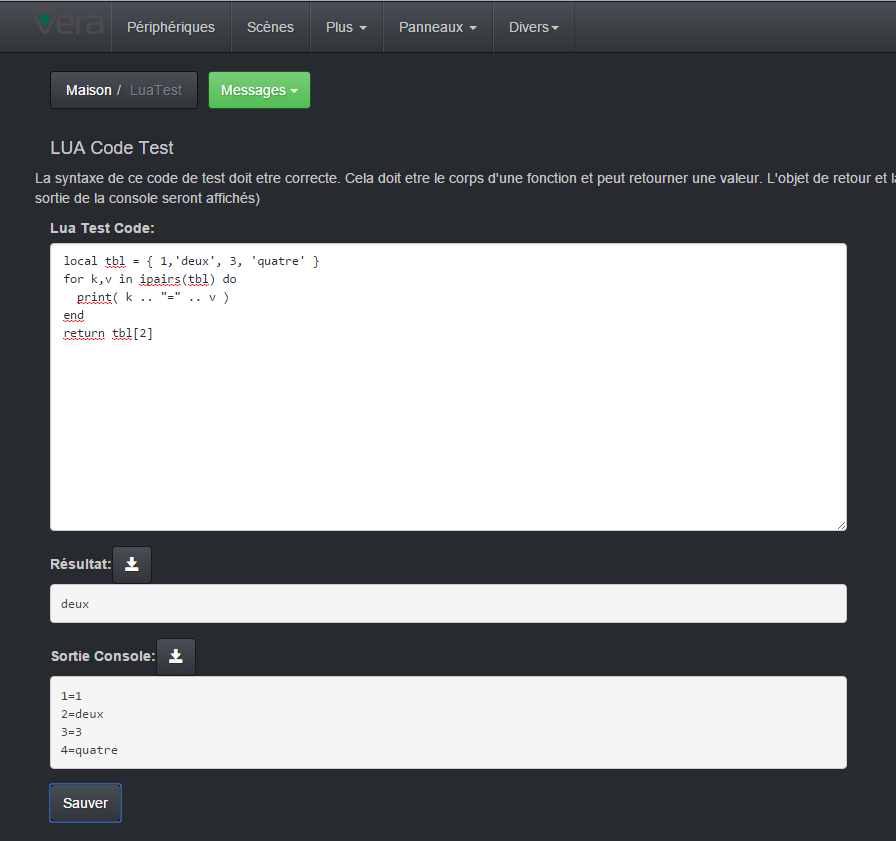
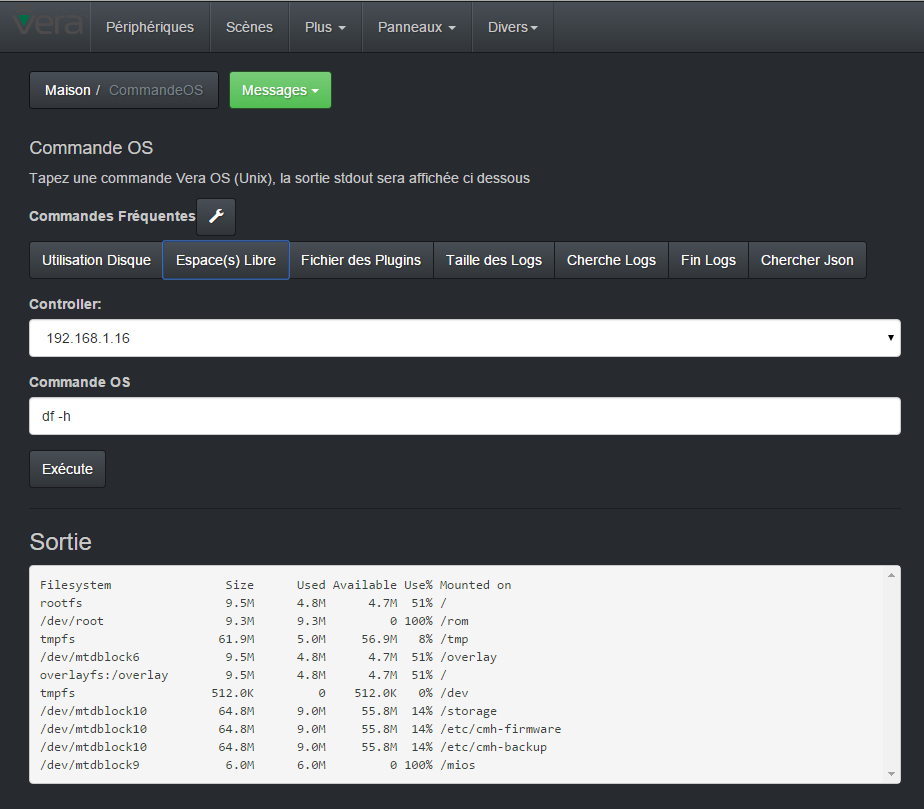
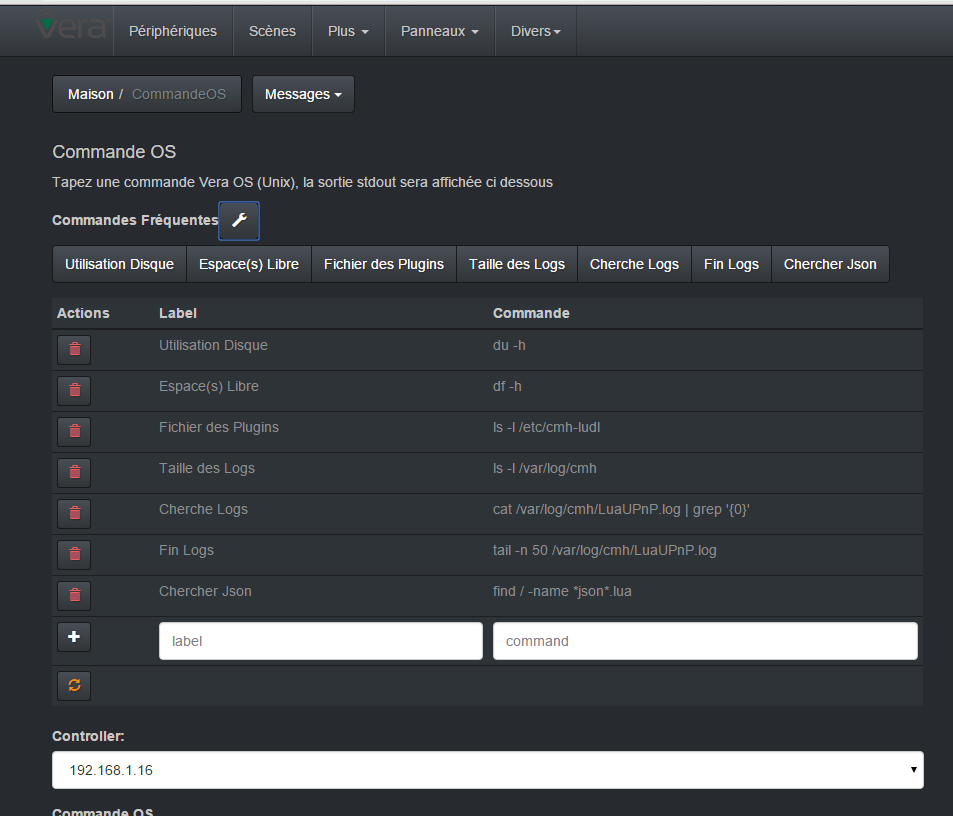
# Les pages customs

Les pages customs sont toujours là, mais quelques options en plus et un widget « frame » ont fait leur apparition. Chaque frame peut avoir un fond ce qui permet de faire des choses comme ceci



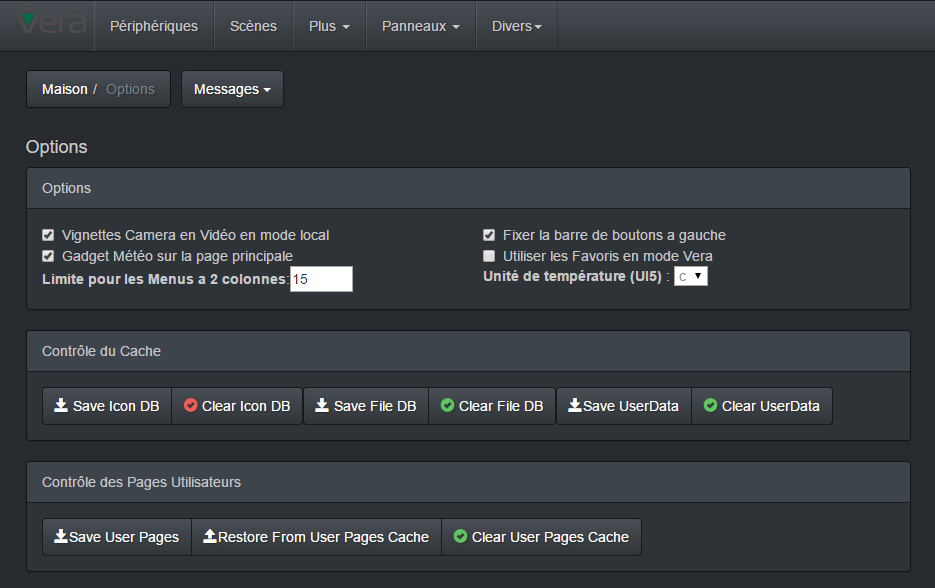
# Le Menu Divers ( Misc )

Le menu MISC/ Divers permet des choses un petit plus avancée comme

* Editer le lua startup  
  
* De tester un bout de code lua et de voir les sorties stdout ( print() )  
  
* D’executer a distance sur la VERA des commandes OS ( attention a savoir ce que vous faites avec ca ) , certaines commande de base sont fournies sous la forme de boutons preconfigurés  
    
  il est possible de modifier, rajouter des boutons préconfigurés et d’avoir des paramètres dans les commandes OS. Ces valeurs sont alors demandées a l’utilisateur au moment de l’exécution de la commande (voir le {0} ci-dessous )  
  

# Les options

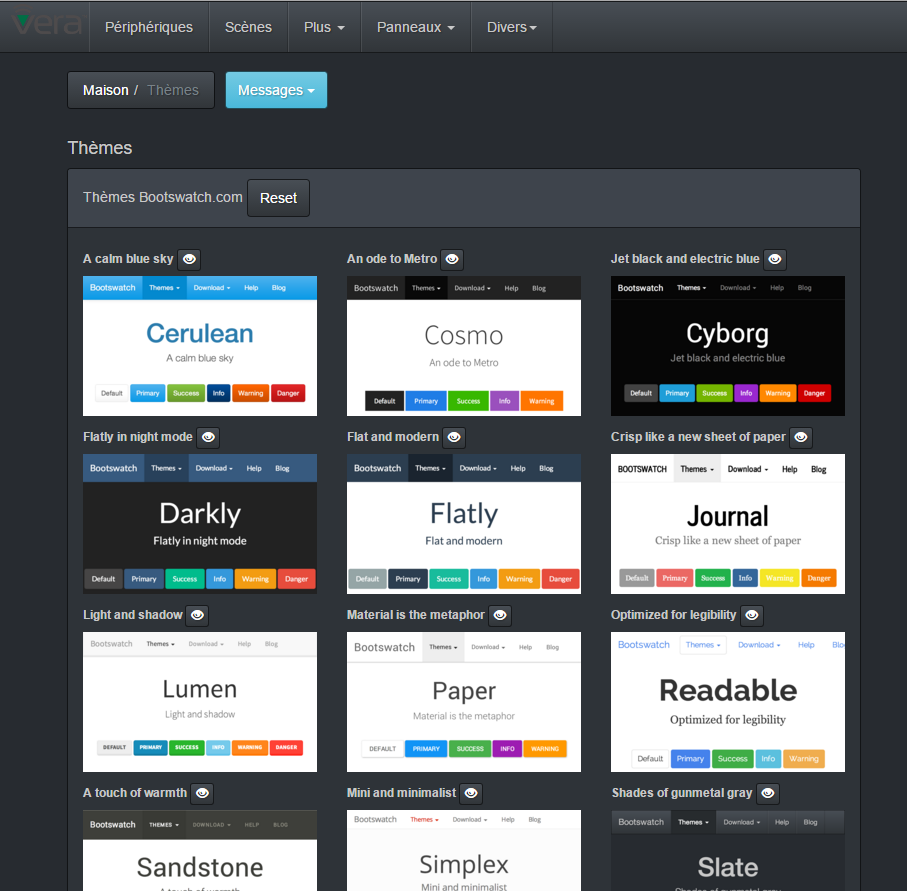
La page options regroupe les options possibles qui sont pour la plupart stockées dans la mémoire du poste client utilise ( pc, mac, téléphone, tablette etc. )



# La sélection des thèmes bootstrap sur bootswatch

ALTUI étant basé sur bootstrap, il est skinnable facilement a partir de thèmes bootstrap 3.3.5 que l’on trouve sur internet ou que l’on fait soit même avec le bootstrap configurateur sur le site bootstrap. Il est possible d’indiquer, par l’intermédiaire de variables du périphérique ALTUI, l’emplacement de fichier bootstrap spécifiques pour le theming ou pour des customisations avancées comme changer les marqueur des différentes taille d’écran et qui contrôle l’aspect responsive design de bootstrap

Afin de faciliter la vie de l’utilisateur, une page de thème sélectionnable, en liaison avec le site bootswatch est présentée, il suffit de cliquer sur un thème pour l’appliquer



# La page debug

Enfin la page debug permet d’aller puiser dans les structures de données et les API de ALTUI pour afficher , contrôler et debugger l’application dans certains cas. Il est possible d’afficher tous les modules et leur variable, de rechercher le nom d’une variable parmi les modules, d’exécuter un bout de code javascript dans le contexte de ALTUI ( et donc d’appeler les API de ALTUI )

