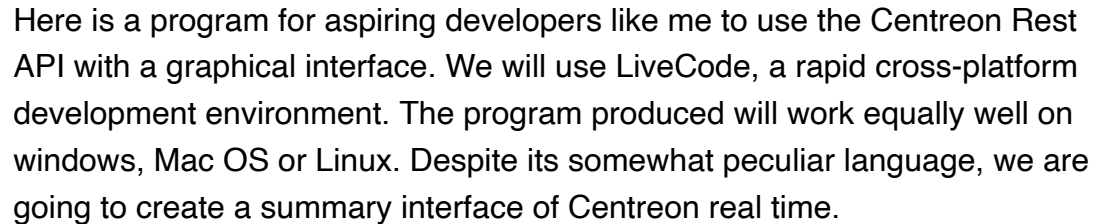


Kermith's workshop ([https://translate.googleusercontent.cc depth=1&hl=en&prev=search&pto=aue&rurl=translate.goog](https://translate.googleusercontent.com/deepth=1&hl=en&prev=search&pto=aue&rurl=translate.goog))

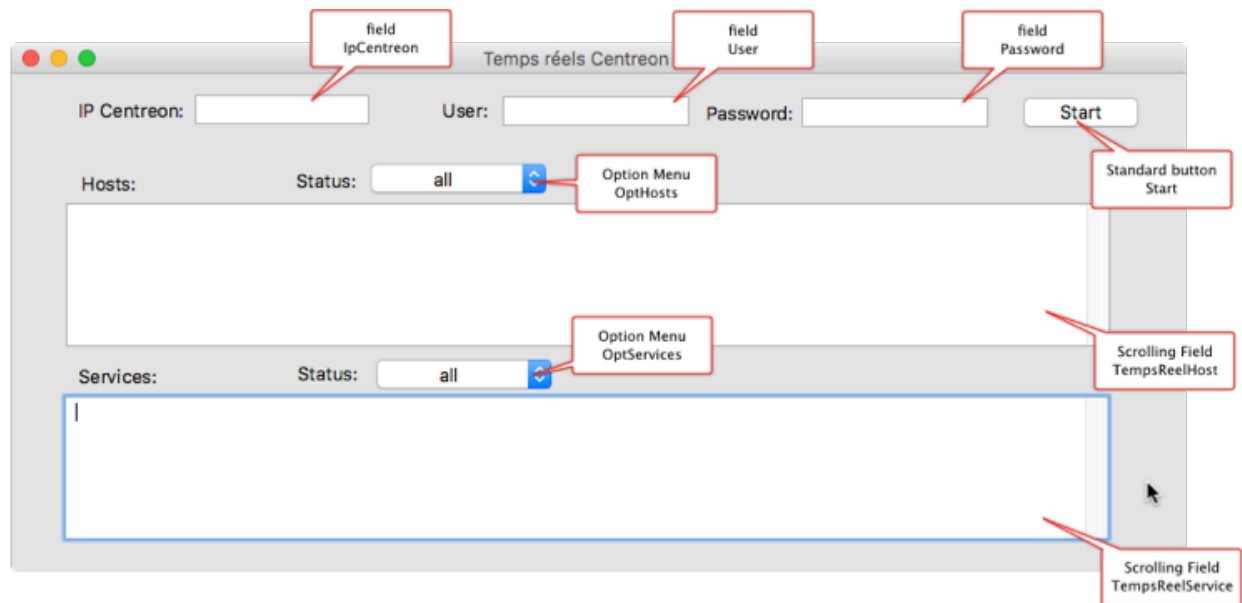
Article created or modified on: 18/1/2018

clapi: UNSUPPORTED



We will use LiveCode Community 9. We will need a Centreon supervision server with an account authorized to perform API Rest requests.

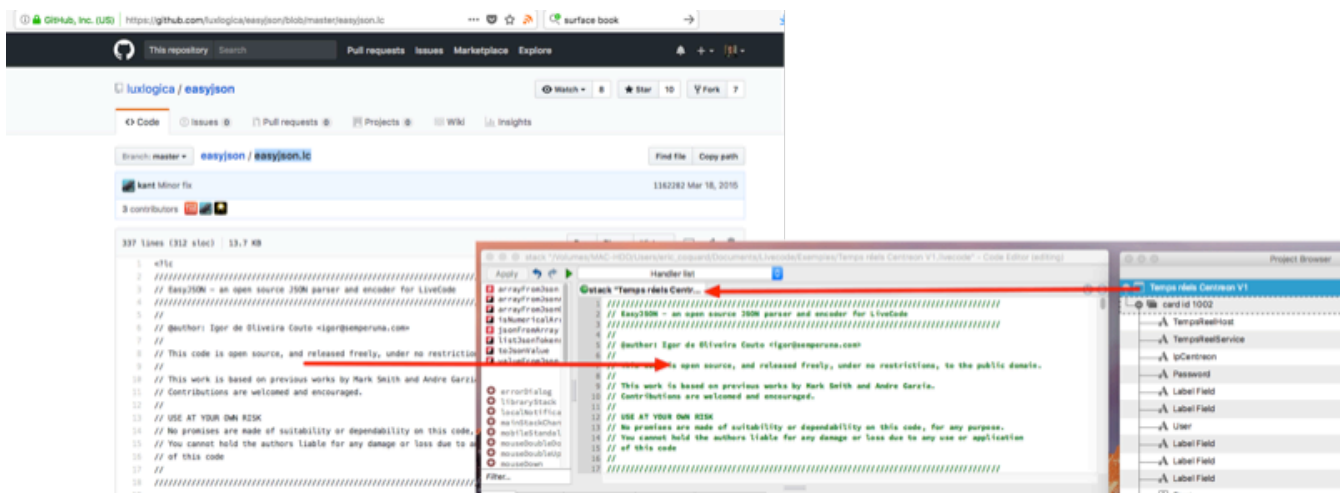
Our interface will be relatively simple for now. We will have a single window called stack.



Interface creation

3 The interface code

It will be necessary to add the library for the management of JSON results. You will find this library at this address <https://github.com/luxlogica/easyjson> (https://translate.googleusercontent.com/translate_c?depth=1&hl=en&prev=search&pto=aue&rurl=translate.google.com&sl=fr&sp=nmt4&u=https://c). Just copy the content of the easyjson.lc file and paste it into the program project.



Adding the library

Actuellement notre projet est réduit à sa plus simple expression :

- paramétrage de la connexion Api Rest
- gestion du mot de passe (il faut bien reconnaître un peu tordu avec LiveCode)
- gestion des status des hôtes et services

- lancement de la commande avec le résultat.

3.1 paramétrage de la connexion Api Rest

Nous aurons besoin de l'adresse IP du serveur Centreon, le user et son mot de passe qui est habilité au Api Rest. Il y aura aussi la gestion du mot de passe de l'objet Field password.

on keyDown pKeyName

put the hiddenText of me into temp

put pKeyName after temp

set the hiddenText of me to temp

put "*" after me

put temp

end keyDown

on backspaceKey

set the hiddenText of me to empty

set the text of me to empty

end backspaceKey

Pour la préparation de la requête, nous utiliserons l'objet button. On récupère le token d'authentification avant de lancer la commande pour obtenir le temps réel.

on mouseUp pMouseButton

-- empty fields

put empty into fld "TempsReelHost"

put empty into fld "TempsReelService"

put the fld "IpCentreon" into tIpCentreon

put the fld "User" into tUser

put the hiddenText of fld "Password" into tPassword

put "Accept: application/json, text/plain, text/event-stream" & cr into couchHeaders

put "Content-Type:application/x-www-form-urlencoded" after couchHeaders

set httpheaders to couchHeaders

put "username=" & tUser & "&password=" & tPassword into tData

put "http://" & tIpCentreon & "/centreon/api/index.php?action=authenticate" into tUrl

post tData to URL tUrl

put JSONToArray(the urlResponse) into tAuthenticate

put tAuthenticate["authToken"] into tToken

```

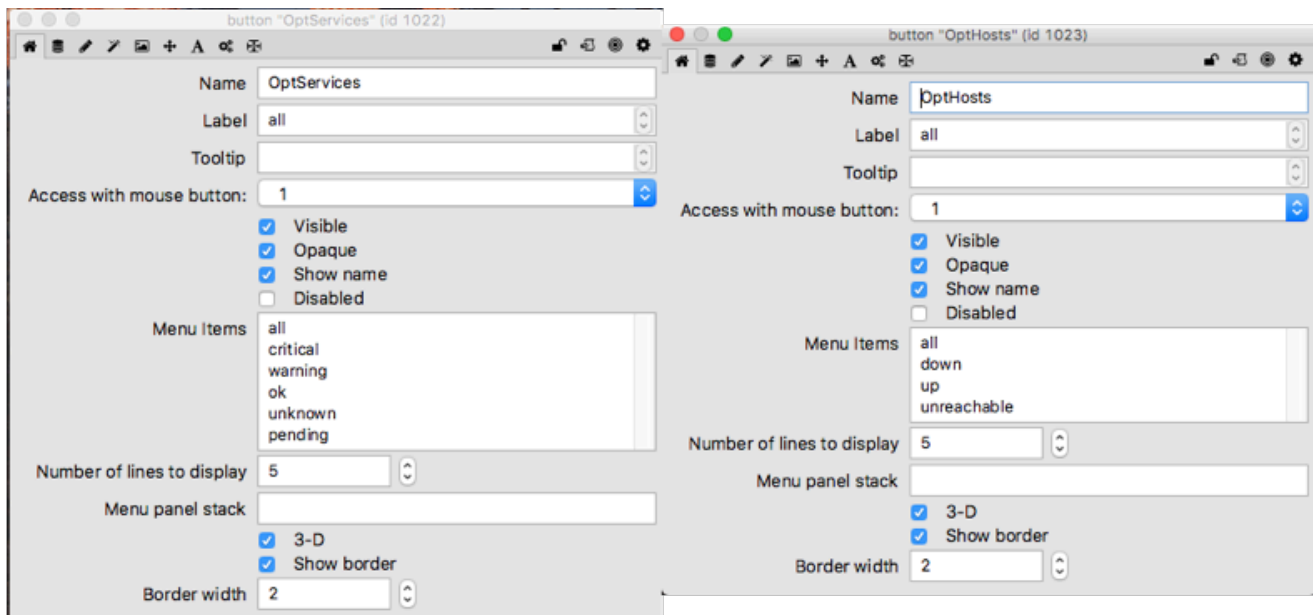
put "Content-Type: application/json" & cr into couchHeaders
put "centreon-auth-token: " after couchHeaders
put tToken after couchHeaders
set httpheaders to couchHeaders

```

.....

end mouseUp

Avant d'aller plus loin, vous allons ajouter deux objets Option Menu OptHosts et OptServices qui correspond au différents états des hôtes et services.



propriétés des objets Option Menu

Et voici le code pour OptHosts

global tOptHosts

-- Sent when a menu item is picked from the option menu

on menuPick pltemName

switch pltemName

case "all"

put "&status=all" into tOptHosts

break

case "down"

put "&status=down" into tOptHosts

break

```
    case "up"
      put "&status=up" into tOptHosts
      break
    case "unreachable"
      put "&status=unreachable" into tOptHosts
      break

  end switch
end menuPick

et pour OptServices

global tOptServices

-- Sent when a menu item is picked from the option menu
on menuPick pltemName
  switch pltemName
    case "all"
      put "&status=all" into tOptServices
      break
    case "warning"
      put "&status=warning" into tOptServices
      break
    case "critical"
      put "&status=critical" into tOptServices
      break
    case "ok"
      put "&status=ok" into tOptServices
      break
    case "pending"
      put "&status=pending" into tOptServices
      break
    case "unknown"
      put "&status=unknown" into tOptServices
      break

  end switch
```

end menuPick

Il nous reste à rajouter le code dans l'objet button qui permettra d'afficher le temps réel.

global tOptHosts

global tOptServices

-- Sent when the mouse is released after clicking

-- pMouseButton specifies which mouse button was pressed

on mouseUp pMouseButton

-- empty fields

put empty into fld "TempsReelHost"

put empty into fld "TempsReelService"

put the fld "IpCentreon" into tIpCentreon

put the fld "User" into tUser

put the hiddenText of fld "Password" into tPassword

put "Accept: application/json, text/plain, text/event-stream" & cr into couchHeaders

put "Content-Type:application/x-www-form-urlencoded" after couchHeaders

set httpheaders to couchHeaders

put "username=" & tUser & "&password=" & tPassword into tData

put "http://" & tIpCentreon & "/centreon/api/index.php?action=authenticate" into tUrl

post tData to URL tUrl

put JSONToArray(the urlResponse) into tAuthenticate

put tAuthenticate["authToken"] into tToken

put "Content-Type: application/json" & cr into couchHeaders

put "centreon-auth-token: " after couchHeaders

put tToken after couchHeaders

set httpheaders to couchHeaders

put "http://" & tIpCentreon & "/centreon/api/index.php?

object=centreon_realtime_hosts&action=list" & tOptHosts into tUrlHost

put URL tUrlHost into fld "TempsReelHost"

put "http://" & tIpCentreon & "/centreon/api/index.php?

object=centreon_realtime_services&action=list" & tOptServices into tUrlService

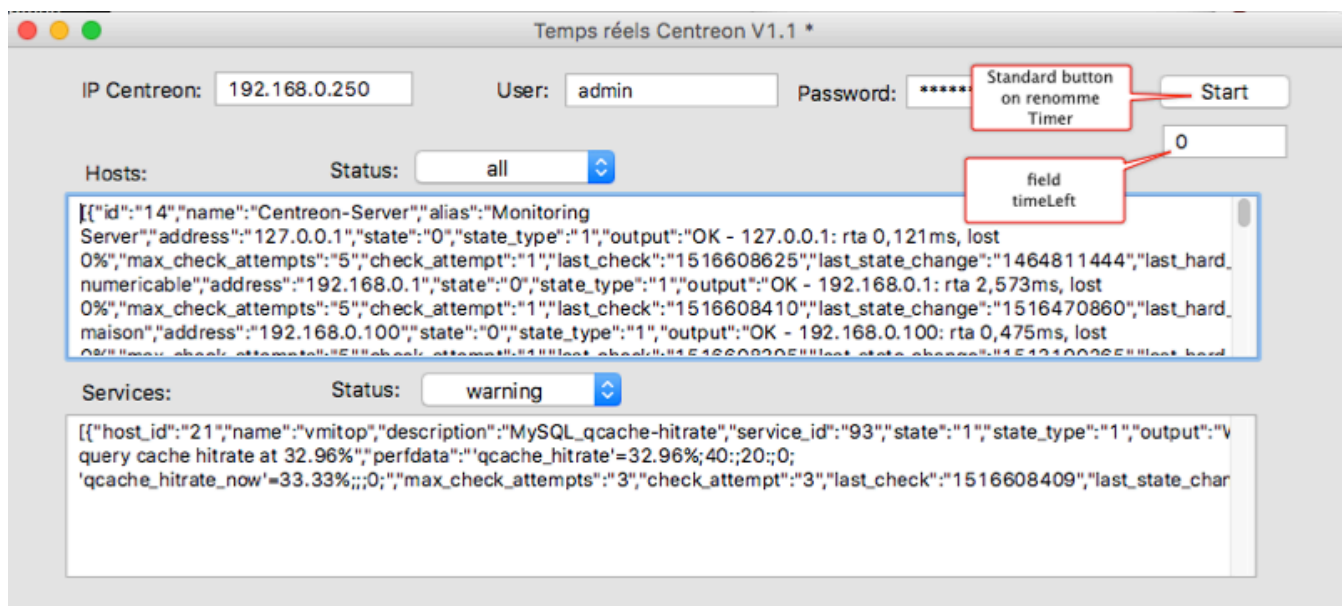
put URL tUrlService into fld "TempsReelService"

end mouseUp

Attention le nombre d'affichage d'objets sera limité à 30, pour augmenter cette limite, il faudra jouer avec la variable limit. Nous verrons bientôt d'autres fonctionnalités de l'API Rest Centreon. Vous pouvez télécharger la première version de mon programme (https://translate.googleusercontent.com/translate_c?depth=1&hl=en&prev=search&pto=aue&rurl=translate.google.com&sl=fr&sp=nmt4&u=https://wRest/Temps-re%25CC%2581els-Centreon-V1.livecode&usg=ALkJrhig-srS5aDFHCGTVGegwHksjB8a3g).

4 Ajout d'un timer pour les requêtes

Améliorons notre programme en rajoutant un timer pour exécuter les requêtes automatiquement toutes les minutes. Nous rajouterons un objet field nommé timeLeft. le bouton start sera renommé timer



ajout d'un timer

Toute la partie du code réalisant la requête API Rest va être copié dans une fonction appelé RequeteAPIRest. On ajoutons une autre fonction pour la temporisation d'une minute, cette dernière s'appellera timerCountDown

global countdownValue

global tOptHosts

global tOptServices

on timerCountDown

subtract 1 from CountdownValue

if countdownValue > 0 **then**

put countdownValue into field "timeLeft"

send "timerCountDown" to me in 1 sec

else

if the label of button "Timer" = "Stop" **then**

RequeteAPIRest

put 60 into countdownValue

put countdownValue into field "timeLeft"

send "timerCountDown" to me in 1 sec

else

put 0 into countdownValue

put countdownValue into field "timeLeft"

end if

end if

end timerCountDown

on RequeteAPIRest

-- empty fields

put empty into fld "TempsReelHost"

put empty into fld "TempsReelService"

put the fld "IpCentreon" into tIpCentreon

put the fld "User" into tUser

put the hiddenText of fld "Password" into tPassword

put "Accept: application/json, text/plain, text/event-stream" & cr into couchHeaders

put "Content-Type:application/x-www-form-urlencoded" after couchHeaders

set httpheaders to couchHeaders

put "username=" & tUser & "&password=" & tPassword into tData

put "http://" & tIpCentreon & "/centreon/api/index.php?action=authenticate" into tUrl

post tData to URL tUrl

put JSONToArray(the urlResponse) into tAuthenticate

put tAuthenticate["authToken"] into tToken

put "Content-Type: application/json" & cr into couchHeaders

put "centreon-auth-token: " after couchHeaders

put tToken after couchHeaders

set httpheaders to couchHeaders

put "http://" & tlpCentreon & "/centreon/api/index.php?"

object=centreon_realtime_hosts&action=list" & tOptHosts into tUrlHost

put URL tUrlHost into fld "TempsReelHost"

put "http://" & tlpCentreon & "/centreon/api/index.php?"

object=centreon_realtime_services&action=list" & tOptServices into tUrlService

put URL tUrlService into fld "TempsReelService"

end RequeteAPIRest

Modifions le code du bouton timer

global tOptHosts

global tOptServices

global countDownValue

-- Sent when the mouse is released after clicking

-- pMouseButton specifies which mouse button was pressed

on mouseUp pMouseButton

if the label of button "Timer" = "Start" **then**

set the label of button "Timer" to "Stop"

put 60 into countDownValue

put countDownValue into field "timeLeft"

send "timerCountDown" to me in 1 sec

RequeteAPIRest

else

set the label of button "Timer" to "Start"

put 0 into countDownValue




put countDownValue into field "timeLeft"

end if

end mouseUp

Le programme est maintenant modifié, lors de la première requête on déclenche un timer qui exécutera une requête API Rest automatiquement toutes les minutes. Pour arrêter les requêtes automatiques, il suffira de cliquer à nouveau sur le bouton. Vous pouvez télécharger la version 1.1 de mon programme (https://translate.googleusercontent.com/translate_c?depth=1&hl=en&prev=search&pto=aue&rurl=translate.google.com&sl=fr&sp=nmt4&u=https://vRest/Temps-re%25CC%2581els-Centreon-V1.1.livecode&usg=ALkJrhixZkad73HAIE0wy8OsSsGbBxUFBg).

0 Commentaires Sugarbug  Règles de confidentialité de Disqus  1 S'identifier ▼

 Recommander  Tweet  Partager Les meilleurs ▼



Commencer la discussion...

S'IDENTIFIER AVEC

OU INSCRIVEZ-VOUS SUR DISQUS 

Nom

Soyez le premier à commenter.

 S'abonner  Ajoutez Disqus à votre site web !Ajouter DisqusAjouter

© 2020 Eric Coquard [Contact me \(mailto:eric.coquard@sugarbug.fr\)](mailto:eric.coquard@sugarbug.fr) Last modified: 06/30/2020