

Module de source de type synchronizer-collector-linker (pour lier 2 Shinkens)

Explication

Le module WebUI permet de générer l'interface de visualisation. Il est possible de modifier les paramètres (Langue de l'interface, port d'écoute) via le fichier de configuration ci-dessous.

Mécanisme des données manquantes



Voir : [Détermination de l'état de l'hôte](#) (pour plus d'information sur les cas particuliers)

Configuration

MongoDB

Le module WebUI a besoin de se connecter à une base MongoDB pour stocker les portails, tableaux de bords et favoris des utilisateurs. Le paramétrage pour la connexion à cette base Mongo est décrit dans la page du [Module MongoDB](#)

Récupération des broks

Le fonctionnement du thread de récupération des **broks** peut être configuré via certains paramètres, afin de modifier son "agressivité".

Pendant la mise à jour des données de supervision, le module ne peut pas répondre aux requêtes HTTP qu'il reçoit.



Une mauvaise configuration de ces paramètres peut compromettre le bon fonctionnement du module, se rapprocher du support si vous avez le moindre doute

Principe de l'algorithme d'absorption des broks :

1. Attente de broks à traiter
2. Récupération de broks en retard (fonctionnalité de rattrapage)
3. Dé-sérialisation des broks
4. Entrée en session critique (les requêtes HTTP sont bloquées)
5. Traitement des broks
6. Libérer la session critique et attendre de nouveaux broks, **ou** continuer l'absorption de broks (en cas de retard important, on repart à l'étape 1. en restant sur la session critique)

Clé	Type	Valeur par défaut	Description
<code>webui_broks_getter_include_deserialization_and_catchup_in_lock</code>	Booléen	0	Entrer en session critique après la récupération d'un premier brok set . La récupération des broks en retard, et la dé-sérialisation se font alors dans la session critique (Locké) pour disposer d'un maximum de temps CPU <ul style="list-style-type: none">• 1 : Activer• 0 Désactiver
<code>webui_broks_getter_activate_late_set_catchup</code>	Booléen	1	Utilisation de la fonctionnalité de rattrapage pour absorber des broks en retard : <ul style="list-style-type: none">• 1 : Activer• 0 Désactiver
<code>webui_broks_getter_nb_late_set_allowed_before_catchup</code>	Entier	10	Nombre de brok set en attente tolérés. Au dessus de ce nombre, les brok set sont immédiatement récupérés par l'algorithme de rattrapage pour être traités maintenant
<code>webui_broks_getter_catchup_broks_managed_by_module_in_a_catchup_loop</code>	Entier	200000	Nombre maximal de broks que l'algorithme de rattrapage récupère avant de lancer le traitement Ce paramètre permet de borner la consommation mémoire et le temps d'exécution d'un tour de boucle de traitement

webui__broks_getter__catchup_run_endless_until_nb_late_set_allowed_reached	Booléen	1	Après traitement des broks , si le nombre de brok set en retard est trop élevé, <ul style="list-style-type: none"> • 1 continuer le rattrapage et absorber des broks en retard (en restant sur la session critique, ou "avec le lock") • 0 arrêter l'absorption de brok et libérer la session critique (rendre le lock)
--	---------	---	---

Fichier de configuration

Voici le fichier CFG de configuration présent dans : **/etc/shinken/modules/webui.cfg**

```

#####
# WebUI
#####
# Daemons that can load this module:
# - broker
# This module is the visualization interface
#####

define module {

# Shinken Enterprise. Lines added by import core. Do not remove it, it's used by Shinken Enterprise to
update your objects if you re-import them.
    _SE_UUID          core-module-2e0f08c45adc11e59a90080027f08538
    _SE_UUID_HASH     99215795c7e21b658e7c34989e872fcc
# End of Shinken Enterprise part

    ##### Module identity #####
    # Module name. Must be unique
    module_name       WebUI

    # Module type (to load module code). Do not edit.
    module_type       webui

    ##### Listening address #####
    # host: IP address to listen to.
    #   note: 0.0.0.0 = all interfaces.
    host               0.0.0.0
    # port to listen
    port               7767

    # HTTPs part, enable if you want to set the visualisation interface listen in HTTPs mode
    # disabled by default. Set your own certificates.
    use_ssl             0
    ssl_cert            /etc/shinken/certs/server.cert
    ssl_key             /etc/shinken/certs/server.key

    ##### Security #####
    # auth_secret: secret key used to crypt cookie.
    # note: must be the same as the configuration UI
    auth_secret        AUTH-SECRET-22316-4472-3256

    # Allow or not HTML chars in plugins output
    allow_html_output  0

    # Use users ACL or enable all for everyone
    manage_acl         1

    # Remote application authentication
    # if 1: allow the user to be load from a HTTP Header
    remote_user_enable  1

    # which HTTP header to get user name if remote_user_enable is 1
    remote_user_variable X-Forwarded-User

    # if remote_user_enable is 1,

```

```

# http_remote_user_case_sensitive to 1 enable case check on remote user login
# http_remote_user_case_sensitive to 0 disable case check on remote user login
# default value : 1, login is case sensitive
remote_user_case_sensitive 1

# Maximum output length for plugin output, 0=no limit
max_output_length 0

##### Lang #####
# Select the lang that will be used by default on the UIs
# Currently managed:
# -en (english)
# -fr (français)
lang fr

##### Paths #####
share_dir /var/lib/shinken/share
photo_dir /var/lib/shinken/photos

##### Modules #####
# Modules loaded by the Visualisation interface
# Available:
# - Cfg_password : check password from the user configuration
# - auth-active-directory : check password from active directory
# - MongoDB : [mandatory] use to save user data (hive, favorites, ...)
# - webui-enterprise : [mandatory]
# - sla : [mandatory] read sla from this module definition
modules Cfg_password, MongoDB,webui-enterprise, sla

##### Metrology access #####
# Multi-realm graphite parameter
graphite_backends *:127.0.0.1

# Before a graphite query is done, the graphite server is tested
# Timeout for the alive timeout
# default: 10 (seconds)
metrology_ping_timeout 10

# Timeout for graphite queries
# default: 20 (seconds)
metrology_query_timeout 20

# If the test does fail, the graphite server will be exclude during this time
# to avoid to lock query for timeouts
# default: 120 (seconds)
metrology_after_error_wait_before_retry 120

##### Broks getter in WebUI #####
# These parameters allow some internal tuning in broks management in WebUI

# Take the lock as soon as getter thread has some broks to manage
# in order to attempt to reduce concurrent usage of CPU
# webui_broks_getter_include_deserialisation_and_catchup_in_lock 0

# Enable or disable late broks sets catchup
#
webui_broks_getter_activate_late_set_catchup
1

# Take extra broks sets to manage if more than this parameter sets are waiting
# webui_broks_getter_nb_late_set_allowed_before_catchup 10

# Stop taking extra broks sets in catchup when we reach this number of broks
# webui_broks_getter_catchup_broks_managed_by_module_in_a_catchup_loop 200000

```

```
# Continue catchup if too much late broks sets remains after
# webui_broks_getter_catchup_run_endless_until_nb_late_set_allowed_reached      1

#===== Extended configuration =====
[OVERLOAD_FROM]          /etc/shinken/_default/daemons/brokers/modules/webui.cfg
[OVERLOAD_FROM]          /etc/shinken-user/configuration/daemons/brokers/modules/webui
/webui_cfg_overload.cfg

}
```