

Module WebUI

Sommaire

[Exemple de fichier de configuration](#)
[Détails des sections composant le fichier de configuration](#)
 [Identification du module](#)
 [Configuration de l'interface et du port d'écoute](#)
 [Sécurisation de la communication avec le module](#)
 [Sécurité du module](#)
 [Langue](#)
 [Dossiers de fichiers partagés et de photos](#)
 [Modules du module WebUI](#)
 [Connexion à la base MongoDB](#)
 [Accès aux données de Métrologie](#)
[Récupération des brokers](#)
 [Les logs d'activité des utilisateurs \(authentification et session \)](#)
 [Surcharge de la configuration](#)

La configuration du module se trouve par défaut dans le fichier `/etc/shinken/modules/webui .cfg`

- Vous trouverez aussi systématiquement un exemple dans `/etc/shinken-user-example/configuration/daemons/brokers/modules/webui /webui-example.cfg`

Exemple de fichier de configuration

```
#####
# 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          THIS SHOULD BE CHANGED BEFORE PRODUCTION

# 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    0

# which HTTP header to get user name if remote_user_enable is 1
remote_user_variable  X_Remote_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          LANG

##### 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 #####
# This parameter defines the graphite server addresses for each realm used to create the metrology
data.
# The address preceded by the character * corresponds to the default graphite.
# You can specify an address for a specific realm. # To define multiple graphite backends addresses,
values must be separated by a comma
# Ex. : graphite_backends France:127.0.0.1, Bordeaux:192.168.1.100:8080
# Ex. : graphite_backends *:127.0.0.1, Paris:192.168.2.200:8080
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

# 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   0

# 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

##### Webui Authentication External Log #####
# Log the webui authentication history in a file

# Enable authentication log or not.
# by default: 0 (disabled)
#   /\ Need be change in the file [OVERLOAD_FROM] webui_cfg_overload.cfg
# webui__log_users__enabled=0

# File use for log authentication history
#   /\ Need be change in the file [OVERLOAD_FROM] webui_cfg_overload.cfg
# webui__log_users__file_path=/var/log/shinken/webui/log_users.log

# Add user name to log.
# by default: 0 (disabled)
#   /\ Need be change in the file [OVERLOAD_FROM] webui_cfg_overload.cfg
# webui__log_users__add_user_name=0

##### Extended configuration #####
# The line [OVERLOAD_FROM] /etc/shinken/_default/daemons/brokers/modules/webui.cfg must be present and
must not be changed !
# You can overload value in the webui_cfg_overload.cfg.
[OVERLOAD_FROM] /etc/shinken/_default/daemons/brokers/modules/webui.cfg
[OVERLOAD_FROM] /etc/shinken-user/configuration/daemons/brokers/modules/webui/webui_cfg_overload.cfg
}

```

Détails des sections composant le fichier de configuration

Identification du module

Il est possible de définir plusieurs instances de module de type "webui" dans votre architecture Shinken.

- Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Commentaire
-----	------	-------	--------	-------------

module_name	Texte	---	WebUI	Nous vous conseillons de choisir un nom en fonction de l'utilisation du module pour que votre configuration soit simple à maintenir. Doit être unique.
module_type	Texte	---	webui	Ne peut être modifié.

Configuration de l'interface et du port d'écoute

```

===== 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

```

Il est possible de configurer l'interface réseau sur laquelle est mise à disposition l'interface de visualisation. Si par exemple l'interface de visualisation ne doit être accessible seulement via un réseau local, il est possible de n'écouter que les requêtes sur cette interface réseau.

Les paramètres suivants permettent de configurer l'accès à l'interface de visualisation :

Nom	Type	Unité	Défaut	Commentaire
host	Texte	Adresse IPv4	0.0.0.0	L'interface réseau sur laquelle le module "WebUI" va écouter.
port	Texte	Port réseau	7767	Port réseau sur lequel le module "WebUI" va écouter.

Sécurisation de la communication avec le module

```

# 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

```

L'interface de visualisation est accessible via HTTP. Il est recommandé d'utiliser le protocole HTTPS pour chiffrer la communication avec le module.

Si pour des raisons de sécurité, cette interface de visualisation doit être accessible via HTTPS, il est possible de configurer les certificats avec les paramètres suivants :

Nom	Type	Unité	Défaut	Commentaire
use_ssl	Booléen	---	0	Permet d'activer ou non l'utilisation du protocole HTTPS. 0 : Désactivé (<i>utilise HTTP</i>) 1 : Activé (<i>utilise HTTPS</i>)
ssl_cert	Texte	Chemin	/etc/shinken/certs/server.cert	Chemin vers le certificat SSL utilisé par le protocole HTTPS.
ssl_key	Texte	Chemin	/etc/shinken/certs/server.key	Chemin vers la clé SSL utilisée par le protocole HTTPS.

Sécurité du module

```

===== 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

# 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

```

Nom	Type	Unité	Défaut	Commentaire
auth_secret	Texte	---	---	Les cookies sont chiffrés par l'interface afin que les utilisateurs ne puissent pas les créer et récupérer le compte d'un autre utilisateur. Cette clé de chiffrement est une chaîne de caractères aléatoire. Il vous est possible de créer votre propre clé en mettant des caractères aléatoires les uns à la suite des autres. Plus votre clé sera longue, plus votre cookie sera sécurisé. Pour cela, il utilise la clé de chiffrement contenue dans ce paramètre .  Doit être la même que celle de l'interface de configuration. Voir la page Synchronizer.cfg (Paramètres globaux)  Ce paramètre est généré lors de l'installation et n'a pas à être modifié par les administrateurs.
manage_acl	Booléan	---	1	Gestion des droits d'exécution des utilisateurs sur l'exécution d'actions sur l'interface de visualisation. (ex: <i>Forcer le statut, définir une période de maintenance ...</i>) Selon les droits définis sur l'utilisateur, certains boutons d'actions lui seront cachés.
remote_user_enable	Booléan	---	0	Authentification de l'application à distance. C'est une authentification unique (<i>SSO</i>). Elle est souvent activée pour permettre de se connecter de manière centralisée à des services. Pour plus de détail sur l'utilisation de l'authentification unique, référez-vous à la page WebUI - Authentification unique (SSO).
remote_user_variable	Texte		X-REMOTE-USER	En-tête HTTP qui permettra d'obtenir le nom de l'utilisateur si le paramètre "remote_user_enable" est activé. Pour plus de détail sur l'utilisation de l'authentification unique, référez-vous à la page WebUI - Authentification unique (SSO).
remote_user_case_sensitive	Booléan		1	Vérification de la casse sur le login de l'utilisateur distant seulement si le paramètre remote_user_enable est activé 1 : Active la vérification 0 : Désactive la vérification Pour plus de détail sur l'utilisation de l'authentification unique, référez-vous à la page WebUI - Authentification unique (SSO).

Langue

```

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

```

Nom	Type	Unité	Défaut	Commentaire
lang	Texte	---	en	Modification de la langue utilisée dans l'interface de visualisation. À ce jour, deux langues sont disponibles - en (Anglais) - fr (Français)

Dossiers de fichiers partagés et de photos

```

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

```

Nom	Type	Unité	Défaut	Commentaire
share_dir	Texte	---	/var/lib/shinken /share	Emplacement du dossier des fichiers partagés (contiens les icônes affichées dans les listes de l'interface de configuration)
photo_dir	Texte	---	/var/lib/shinken /photos	Emplacement du dossier des photos

Modules du module WebUI

```

#===== 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

```

Plusieurs modules sont disponibles pour le module WebUI

Nom	Type	Unité	Défaut	Commentaire
modules	Texte	---		Liste de noms de sous modules, séparés par des virgules. Par défaut, aucun sous module n'est activé.

Connexion à la base MongoDB

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

Accès aux données de Métrologie

```

===== Metrology access =====
# This parameter defines the graphite server addresses for each realm used to create the metrology data.
# The address preceded by the character * corresponds to the default graphite.
# You can specify an address for a specific realm.
# To define multiple graphite backends addresses, values must be separated by a comma
# Ex. : graphite_backends France:127.0.0.1, Bordeaux:192.168.1.100:8080, Paris:http://192.168.2.200:200
# Ex. : graphite_backends *:127.0.0.1, Paris:192.168.2.200:8080
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

```

Clé	Type	Défaut	Description
graphite_backends	Liste	---	Ce paramètre définit les adresses des serveurs graphite pour chaque royaume utilisé pour créer les données de métrologie. La syntaxe est la suivante : REALM:HOSTNAME:PORT REALM : Nom du royaume : - permet de définir un royaume précis, - la valeur par défaut (*) regroupe les royaumes non définis explicitement. HOSTNAME : Nom de l'hôte ou adresse IP du serveur Graphite ; PORT : Port du service Graphite, configurable dans le fichier /etc/httpd/conf.d/graphite.conf. Si le port n'est pas précisé, c'est le port 80 qui sera utilisé (<i>par défaut</i>). Pour définir plusieurs <i>graphite_backend</i> sur une WebUI, les adresses doivent être séparées par des virgules Exemple : <code>grapite_backends *:127.0.0.1, France:192.168.100:8080</code> Dans cet exemple, le royaume France utilise l'adresse 192.168.100:8080 et tous les autres royaumes utilisent l'adresse 127.0.0.1 avec le port 80.
metrology__ping_timeout	Entier	10	Avant qu'une requête graphite soit effectuée, une vérification est faite pour s'assurer du bon fonctionnement du serveur graphite. Ce paramètre définit le temps maximum d'exécution de la requête de vérification.
metrology__query_timeout	Entier	20	Temps maximum d'exécution des requêtes graphites.
metrology__after_error_wait_before_retry	Entier	120	Si la commande de vérification du bon fonctionnement du serveur graphite échoue, le module sera suspendu avant de relancer la vérification. Ce paramètre correspond au temps d'attente avant de relancer la vérification.

Récupération des broks

```

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

# 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

# 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_deserialization_and_catchup_in_lock 0

```

Le fonctionnement du mécanisme 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
webui__broks_getter__include_deserialization_and_catchup_in_lock	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 • 1 : Activer • 0 Désactiver
webui__broks_getter__activate_late_set_catchup	Booléen	1	Utilisation de la fonctionnalité de rattrapage pour absorber des broks en retard : • 1 : Activer • 0 Désactiver
webui__broks_getter__nb_late_set_allowed_before_catchup	Entier	10	Nombre de brok set en attente toléré. Au dessus de ce nombre, les brok set sont immédiatement récupérés par l'algorithme de rattrapage pour être traités maintenant
webui__broks_getter__catchup_broks_managed_by_module_in_a_catchup_loop	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, • 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)
webui__broks_getter__include_deserialisation_and_catchup_in_lock			Dans le cas où vous voulez disposer d'un maximum de temps CPU pour traiter les broks en retard, vous pouvez inclure la phase 2 (Récupération de broks en retard) et Phase 3 (Dé-sérialisation des broks) dans la phase Critique (Phase 4) La récupération des broks en retard, et la dé-sérialisation se font alors dans la session critique (Locké) pour • 1 : Activer • 0 Désactiver

Log de la connexion des utilisateurs

Il est possible d'activer le log des utilisateurs dans un fichier à part du fichier de log habituel avec ces paramètres :

```

===== Webui Authentication External Log =====
# Log the webui authentication history in a file

# Enable authentication log or not.
# by default: 0 (disabled)
# /\ Need be change in the file [OVERLOAD_FROM] webui_cfg_overload.cfg
# webui__log_users__enabled=0

# File use for log authentication history
# /\ Need be change in the file [OVERLOAD_FROM] webui_cfg_overload.cfg
# webui__log_users__file_path=/var/log/shinken/webui/log_users.log

# Add user name to log.
# by default: 0 (disabled)
# /\ Need be change in the file [OVERLOAD_FROM] webui_cfg_overload.cfg
# webui__log_users__add_user_name=0

```

Clé	Type	Valeur par défaut	Description
webui__log_users__enabled	Booléen	0	Active ou non le log de la connexion des utilisateurs dans un fichier de logs à part : • 1 : Activé • 0 : Désactivé
webui__log_users__file_path	Booléen	/var/log/shinken/webui/log_users.log	Précise le chemin vers le quel les logs vont être écrits.
webui__log_users__add_user_name	Booléen	0	Ajoute le nom des utilisateur dans les logs. Par défaut, seul un identifiant anonyme sera présent à la place. • 1 : Nom d'utilisateurs visibles • 0 : Nom d'utilisateurs remplacés par des identifiants anonymes

Surcharge de la configuration

```

===== Extended configuration =====
# The line [OVERLOAD_FROM] /etc/shinken/_default/daemons/brokers/modules/webui.cfg must be present and must not be changed !
# You can overload value in the webui_cfg_overload.cfg.
[OVERLOAD_FROM] /etc/shinken/_default/daemons/brokers/modules/webui.cfg
[OVERLOAD_FROM] /etc/shinken-user/configuration/daemons/brokers/modules/webui/webui_cfg_overload.cfg

```

Nom	Type	Unité	Défaut	Commentaire
-----	------	-------	--------	-------------

[OVERLOAD _FROM]	Texte	---	Ce paramètre permet de surcharger les paramètres de la webui. Il contient les paramètres de l'interface, qui peuvent être commun avec d'autres WebUI. Le premier fichier contient les valeurs livrées par défaut par Shinken. Il sera écrasé et mis à jour à chaque mise à jour et installation de patch. Il doit être présent et non modifié. Ce paramètre peut être utilisé plusieurs fois pour ajouter plusieurs fichiers.
---	-------	-----	---