

Module event-manager-reader

Sommaire

Contexte

Activation du module

Exemple d'activation du module nommé "event-manager-reader" sur le module nommé "WebUI" (configuration livrée par défaut par Shinken)

Créer un nouveau module de type event-manager-reader

Configuration

Exemple de fichier de configuration

Détails des sections composant le fichier de configuration

Identification du module

Accès à la base MongoDB

Configuration de l'URI de connexion et de l'authentification par mot de passe

Connexion directe au serveur MongoDB

Connexion par SSH au serveur MongoDB

Gestion de l'auto reconnexion

Configuration pour l'interface de visualisation

Options de notifications par défaut

Paramétrage des notifications audio

Changement du son joué

Paramétrage des notifications visuelles

Options Internes

Contexte

Le module de type **event-manager-reader** est un module qui permet au module WebUI (*l'interface de Visualisation*) du Broker d'afficher le bac à événement (voir la page [Vue - Les Événements](#)).

- Il récupère les événements en base de donnée des événements.
- Il doit donc être listé dans la liste des modules, des modules de type **WebUI** (voir la page [Module WebUI](#), propriété "modules").



Pour que la fonctionnalité "bac à événement" apparaisse dans l'interface, il faut absolument que ce module soit activé.



S'il y a plusieurs configurations de WebUI présentes dans l'architecture, n'oubliez pas d'activer le module dans chaque configuration.

Exemple :

- Plusieurs WebUI sur le même Broker.
- Une WebUI sur différents Brokers qui n'utilise pas la même définition.



Ce module fonctionne en paire avec le module : `event-manager-writer` qui a la charge d'écrire en base les informations (voir la page [Module event-manager-writer](#)).

Activation du module

Les modules de type "event_container" sont des modules qui doivent être activés sur un module de type "webui", qu'on appellera le **module parent**.

- L'activation du module s'effectue en ajoutant le **nom** du module dans la configuration du **module parent**.
 - Pour cela, il faut ouvrir le fichier de configuration du **module parent** (de type "webui"), et ajouter dans le paramètre **modules**, le nom du module de type "event-manager-reader".
- Il est possible de faire plusieurs modules de type "event_container".
 - Cela permet, **par exemple**, d'avoir des configurations différentes en fonction des royaumes.
- S'il y a plusieurs modules de type "webui" présents dans l'architecture, il ne faut pas oublier d'activer le module de type "event-manager-reader" dans la configuration de chacun d'eux.
- **Contraintes** :
 - Activable uniquement sur un module de type "webui" (voir la page [Module WebUI](#)).
 - Il ne peut y avoir qu'un seul module de type "event_container" sur un module de type "webui".

Pour prendre en compte le changement de configuration, il faut redémarrer l'Arbiter :

```
service-shinken-arbiter restart
```

Exemple d'activation du module nommé "event-manager-reader" sur le module nommé "WebUI" (configuration livrée par défaut par Shinken)

L'exemple suivant :

- active le module "event-manager-reader" ,
- sur le module "WebUI", dont la configuration est dans le fichier **/etc/shinken/module/webui.cfg**

```
define module {
    [...]
    modules          Module 1, Module 2, Module 3, event-manager-reader
    [...]
}
```

Puis redémarrage de l'Arbiter

```
service-shinken-arbiter restart
```

Créer un nouveau module de type event-manager-reader

Pour pouvoir configurer un module de type "event-manager-reader", il faut faire un nouveau fichier de configuration grâce au fichier d'exemple fourni par défaut.

- Pour commencer, il faut choisir le nom du nouveau module
 - Pour l'exemple, on l'appelle "Mon-Module-event-manager-reader"
 - Remplacer dans l'exemple le mot "Mon-Module-event-manager-reader" par le nom qui a été choisi.
- Puis il faut créer le fichier de configuration :
 - Copier le fichier de définition du module d'exemple : **/etc/shinken-user-example/configuration/daemons/brokers/modules/webui/modules/event_manager_reader/event_manager_reader-example.cfg** dans le répertoire de définition des modules **/etc/shinken/modules/** .
(*Exemple : /etc/shinken/modules/Mon-Module-event-manager-reader.cfg*)

```
cp /etc/shinken-user-example/configuration/daemons/brokers/modules/webui/modules
/event_manager_reader/event_manager_reader-example.cfg /etc/shinken/modules/Mon-Module-event-
manager-reader.cfg
```

- Ensuite, il faut modifier le fichier nouvellement créé pour configurer le nouveau module.
 - Il faut vérifier que le fichier appartienne à l'utilisateur shinken et qu'il possède le droit d'édition. Si ce n'est pas le cas, il faut effectuer les commandes suivante :

```
chown -R shinken:shinken /etc/shinken/modules/Mon-Module-event-manager-reader.cfg
chmod u+w /etc/shinken/modules/Mon-Module-event-manager-reader.cfg
```

- On change le nom du module en "Mon-Module-event-manager-reader" dans le fichier **/etc/shinken/modules/Mon-Module-event-manager-reader.cfg**

```
...
    # Module name [ Must be unique ]
[ MANDATORY ]
    #

    module_name          Mon-Module-event-manager-reader
...
```

- Ensuite il faut ajouter le nouveau module dans le module de type "WebUI" correspondant

- Dans notre exemple, on ajoute le module "Mon-Module-event-manager-reader" au module "WebUI" définie dans le fichier /etc/shinken/module/ma_webui.cfg

```
define module {
    [...]
    modules                               Module 1, Module 2, Module 3,
    Mon-Module-event-manager-reader
    [...]
}
```

- Puis pour finir il faut redémarrer l'Arbiter pour que le Broker puisse prendre en compte ce nouveau module.

```
service-shinken-arbiter restart
```

Configuration

La configuration du module que Shinken livre par défaut se trouve dans le fichier **/etc/shinken/modules/event_manager_reader.cfg**.

- un exemple dans **/etc/shinken-user-example/configuration/daemons/brokers/modules/event_manager_reader/event_manager_reader-example.cfg**.

Exemple de fichier de configuration

```
#####
# event manager
#####
# Modules that can load this module:
# - WebUI (to display events data to the users)
# This module read event from database to show you in the WebUI
#####

define module {

    # #
    #     MODULE IDENTITY     #
    # #

    # Module name [ Must be unique ]                               [ MANDATORY ]
    #
    module_name                                     event-manager-reader

    # Module type [ Do not edit ]                                   [ MANDATORY ]
    #
    module_type                                     event_container

    # #
    #     DATABASE CONNECTION     #
    # #

    # MongoDB parameters     #

    # MongoDB uri definition . You can find the mongodb uri syntax at
    # https://docs.mongodb.com/manual/reference/connection-string/
    #
    #     Default : mongodb://localhost/?w=1&fsync=false
    #
    # uri                                     mongodb://localhost/?w=1&fsync=false

    # Which database contains events data
    #
    #     Default : event_container
    #
    # database                                     event_container

    # username/password to authenticate to MongoDB.
```

```

# Both parameters must be provided for authentication to function correctly.
#
# broker__module_webui__module_event_manager_reader__database__username
#
# broker__module_webui__module_event_manager_reader__database__password

# SSH tunnel activation to secure your mongodb connection
# That will allow all mongodb to be encrypted & authenticated with SSH
#
#         ...      : Enable  => 1 ( enable ssh tunnel )
#         Default  : Disable => 0 ( disable ssh tunnel )
#
# use_ssh_tunnel          0

# If the SSH connection goes wrong, then retry use_ssh_retry_failure time before_shinken_inactive
#
#         Default : 1 ( try )
#
# use_ssh_retry_failure  1

# SSH user to connect to the mongodb server.
#
#         Default : shinken
#
# ssh_user              shinken

# SSH keyfile to connect to the mongodb server.
#
#         Default : ~shinken/.ssh/id_rsa
#
# ssh_keyfile           ~shinken/.ssh/id_rsa

# SSH Timeout used to test if the SSH tunnel is viable or not, in seconds.
#
#         Default : 10 ( seconds )
#
# ssh_tunnel_timeout    10

#   AutoReconnect Management   #

# When MongoDB require you to reconnect ( For example, It can occur when a new PRIMARY is elected
# in a MongoDB cluster ), it will raised the MongoDB AutoReconnect exception.
#
# How many try to reconnect before module go in error
#
#         Default : 4 ( try )
#
# auto_reconnect_max_try    4

# Time between each try
#
#         Default : 3 ( seconds )
#
# auto_reconnect_sleep_between_try    3

# NOTE: Change these values only if you have a MongoDB cluster and you change the
# heartbeatTimeoutSecs of your MongoDB replica set
# The value of auto_reconnect_max_try * auto_reconnect_sleep_between_try must be higher than
# heartbeatTimeoutSecs in the rs.conf(); of your MongoDB replica set.

# #
# VISUALISATION UI           #
# #

# Specify the max retries when the Visualisation UI can't communicate with the event_manager_reader
# during the Excel export.
# Increase this parameter when users encounter difficulties when generating the Excel export
# of events. This will make the export more tolerant to network loss or database slowness,
# or restart of the broker.
# The value cannot be 0.

```

```

#
#           Default : 2 ( try )
#
# broker__module_webui__module_event_manager_reader__events_export__max_retry 2

# #
#   DEFAULT NOTIFICATIONS OPTIONS           #
# #

# The parameters below allow users to change default configuration of notification for Events list

#   Sound Notification Parameters           #

# Sound notification on new event
#
#           Default : 0 => Disable ( Sound notification is not played )
#           ...      : 1 => Enable ( Sound notification is played )
#
# broker__module_webui__module_event_manager_reader__default_notifications_settings__sound_enabled 0

#   Visual Notification Parameters         #

# Visual notification on new event
#
#           Default : 0 => Disable ( Visual notification is not played )
#           ...      : 1 => Enable ( Visual notification is played )
#
# broker__module_webui__module_event_manager_reader__default_notifications_settings__visual_blink_enabled 0

# #
#   INTERNAL OPTIONS                       #
# #

# INTERNAL : DO NOT EDIT FOLLOWING PARAMETER WITHOUT YOUR DEDICATED SUPPORT
# Broker idle time before considering that Shinken is inactive.
# Use this if you have Broker loop time that exceeds 30 seconds
#
#           Default : 30 ( seconds )
#
# time_before_shinken_inactive                30
}

```

Détails des sections composant le fichier de configuration

Identification du module

Il est possible de définir plusieurs instances de module de type "event-manager-reader" dans l'architecture Shinken.

- Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Description
module_name	Texte	---	event-manager-reader	Shinken conseille de choisir un nom en fonction de l'utilisation du module pour que la configuration soit simple à maintenir. Doit être unique.
module_type	Texte	---	event_container	Ne dois pas être modifié.

Doit être unique

Accès à la base MongoDB

Cette configuration s'effectue dans le fichier de configuration du module.

Pour se connecter à la base MongoDB utilisé pour le stockage des données, deux méthodes sont disponibles :

- **Connexion directe** : Par défaut, mais non sécurisée.
- **Tunnel SSH** : Shinken se connecte à la base MongoDB au travers d'un module SSH pour plus de sécurité.

Configuration de l'URI de connexion et de l'authentification par mot de passe

```
# #
# DATABASE CONNECTION #
# #

# MongoDB parameters #

# MongoDB uri definition . You can find the mongodb uri syntax at
# https://docs.mongodb.com/manual/reference/connection-string/
#
# Default : mongodb://localhost/?w=1&fsync=false
#
# uri mongodb://localhost/?w=1&fsync=false

# Which database contains events data
#
# Default : event_container
#
# database event_container

# username/password to authenticate to MongoDB.
# Both parameters must be provided for authentication to function correctly.
#
# broker__module_webui__module_event_manager_reader__database__username

#
# broker__module_webui__module_event_manager_reader__database__password
```

Nom	Type	Unité	Défaut	Description
uri	Texte	URL	mongodb://localhost/?safe=true	Trouver la syntaxe de l'uri de MongoDB à l'adresse https://docs.mongodb.com/manual/reference/connection-string/ .
database	Texte	---	shinken	Nom de la base de données où sont stockées les données événements.
broker__module_webui__module_event_manager_reader__database__username	Texte	---		Utilisateur pour l'authentification avec mot de passe à la base MongoDB. Utile uniquement si l'activation par mot de passe a été activé (voir la page MongoDB - activation de l'authentification par mot de passe)
broker__module_webui__module_event_manager_reader__database__password	Texte	---		Mot de passe de l'utilisateur utilisé pour l'authentification avec mot de passe à la base MongoDB. Utile uniquement si l'activation par mot de passe a été activé (voir la page MongoDB - activation de l'authentification par mot de passe)

Connexion directe au serveur MongoDB

Par défaut, le module se connecte de manière directe à la base MongoDB pour y lire et écrire les données.

Dans la configuration du module, ceci correspond au paramètre "use_ssh_tunnel" à 0.

Cette méthode de connexion a pour avantage d'être facile à configurer au niveau de Shinken. Par contre, elle oblige à permettre l'accès à la base MongoDB au monde extérieur, et donc s'exposer à des problèmes de sécurité.

La sécurisation de la base MongoDB est bien sûr toujours possible (voir la page [Sécurisation des connexions aux bases MongoDB](#)) mais bien plus complexe à mettre en place. La méthode de connexion par SSH est ainsi préférable pour des raisons pratiques et de sécurité.

Connexion par SSH au serveur MongoDB

```
# SSH tunnel activation to secure your mongodb connection
# That will allow all mongodb to be encrypted & authenticated with SSH
#
#     ...      : Enable => 1 ( enable ssh tunnel )
#     Default  : Disable => 0 ( disable ssh tunnel )
#
# use_ssh_tunnel                                0

# If the SSH connection goes wrong, then retry use_ssh_retry_failure time before_shinken_inactive
#
#     Default  : 1 ( try )
#
# use_ssh_retry_failure                          1

# SSH user to connect to the mongodb server.
#
#     Default  : shinken
#
# ssh_user                                       shinken

# SSH keyfile to connect to the mongodb server.
#
#     Default  : ~shinken/.ssh/id_rsa
#
# ssh_keyfile                                   ~shinken/.ssh/id_rsa

# SSH Timeout used to test if the SSH tunnel is viable or not, in seconds.
#
#     Default  : 10 ( seconds )
#
# ssh_tunnel_timeout                            10
```

Le module peut également se connecter par tunnel SSH à la base MongoDB, pour des raisons de sécurité.

En effet, le paramétrage de MongoDB permet de définir sur quelle interface réseau ce dernier écoute les requêtes. En n'autorisant seulement interface réseau avec l'adresse 127.0.0.1, cela évite d'ouvrir la base au monde extérieur.

Dans la configuration de la base MongoDB (*/etc/mongod.conf*), il faut que le paramètre " *bind_ip* " est positionné pour n'écouter que sur l'interface locale :

- `bind_ip= 127.0 . 0.1`

Dans cette configuration, la base MongoDB écoute que sur l'interface réseau local, pour que le module se connecte, il faut passer par un tunnel SSH. Pour ce faire, activer les options suivantes :

Nom	Type	Unité	Défaut	Description
use_ssh_tunnel	Booléen	---	0	<ul style="list-style-type: none">• 1 : Connexion par tunnel SSH.• 0 : Connexion directe.
use_ssh_retry_failure	Entier	---	1	Spécifie le nombre supplémentaire de tentatives lors de l'établissement du tunnel SSH si ce dernier n'arrive pas à être établi.

ssh_user	Texte	Utilisateur unix	shinken	L'utilisateur avec lequel le tunnel sera établi.
ssh_keyfile	Texte	Chemin de fichier	~shinken/.ssh/id_rsa	La clé ssh privée présente sur le serveur Shinken qui sera utilisé pour établir le tunnel.
ssh_tunnel_timeout	Entier	Seconde	10	Spécifie le timeout en secondes de la vérification du tunnel SSH avant que la connexion vers MongoDB soit effectuée.

Pour configurer les clés SSH à utiliser, voir la page [Création automatique et gestion de la clé SSH de l'utilisateur shinken](#).

Gestion de l'auto reconnexion

```
# AutoReconnect Management

# When MongoDB require you to reconnect ( For example, It can occur when a new PRIMARY is elected
# in a MongoDB cluster ), it will raised the MongoDB AutoReconnect exception.
#
# How many try to reconnect before module go in error
#
#     Default : 4 ( try )
#
# auto_reconnect_max_try                               4

# Time between each try
#
#     Default : 3 ( seconds )
#
# auto_reconnect_sleep_between_try                     3

# NOTE: Change these values only if you have a MongoDB cluster and you change the
#       heartbeatTimeoutSecs of your MongoDB replica set
#       The value of auto_reconnect_max_try * auto_reconnect_sleep_between_try must be higher than
#       heartbeatTimeoutSecs in the rs.conf(); of your MongoDB replica set.
```

Définitions

- **Primaire** : nom de MongoDB pour désigner un serveur maître, le serveur sur lequel il est possible de faire des requêtes d'écriture dans la base.
- **Election** : processus de MongoDB pour choisir un nouveau membre Primaire si le membre Primaire devient inaccessible.

Voir : [Haute disponibilité de la base MongoDB \(mise en place d'un cluster\)](#)

Dans le cas de l'utilisation d'un cluster MongoDB, lorsque le membre Primaire devient inaccessible, une nouvelle élection est déclenchée, ce qui provoque une coupure temporaire de l'accès à la base.

Dans le but de ne pas interrompre le service, le module SLA va se reconnecter automatiquement au cluster MongoDB.

Pour ce faire, il va faire un nombre d'essais égaux au paramètre " auto_reconnect_max_try " avec une pause de X secondes entre chaque essai (correspondant au paramètre " auto_reconnect_sleep_between_try ").

Par défaut pour MongoDB, le temps maximum avant qu'un membre Primaire soit considéré comme indisponible et qu'une nouvelle élection ait lieu est de 10 secondes.

Voir : " heartbeatTimeoutSecs " donné par la commande rs . conf (); dans un shell de MongoDB.

Nom	Type	Unité	Défaut	Description
auto_reconnect_max_try	Entier	---	4	Nombre d'essais de reconnexion à la base.

auto_reconnect_sleep_between_try	Entier	Seconde	3	Temps entre chaque essai en seconde.
----------------------------------	--------	---------	---	--------------------------------------

Les valeurs par défauts du fichier laissent 12 secondes, ce qui est amplement suffisant avec la configuration par défaut de MongoDB.



Il est conseillé de ne pas modifier ces valeurs.

Configuration pour l'interface de visualisation

```
# #
#     VISUALISATION UI     #
# #

# Specify the max retries when the Visualisation UI can't communicate with the event_manager_reader
# during the Excel export.
# Increase this parameter when users encounter difficulties when generating the Excel export
# of events. This will make the export more tolerant to network loss or database slowness,
# or restart of the broker.
# The value cannot be 0.
#
#     Default : 2 ( try )
#
# broker__module_webui__module_event_manager_reader__events_export__max_retry 2
```

Le paramètre "broker__module_webui__module_event_manager_reader__events_export__max_retry" permet de configurer le nombre maximal de tentatives lorsque l'interface de visualisation ne peut pas communiquer avec le module event_manager_reader pendant l'export des données vers Excel.

Cela rendra les exports plus tolérants aux pertes de réseau ou aux lenteurs de la base de données, ou aux redémarrages du Broker.

Nom	Type	Unité	Défaut	Description
broker__module_webui__module_event_manager_reader__events_export__max_retry	Entier	---	2	Nombre d'essais d'appel pour l'export des données vers Excel. Ne peut pas valoir 0.

Options de notifications par défaut

Il est possible de choisir une notification audio ainsi qu'une notification visuelle lorsqu'il y a un nouvel événement.

Les valeurs de ces paramètres impacteront seulement les valeurs par défaut de la liste d'événements.

```

# #
#     DEFAULT NOTIFICATIONS OPTIONS     #
# #

# The parameters below allow users to change default configuration of notification for Events list

# Sound Notification Parameters  #

# Sound notification on new event
#
#     Default : 0 => Disable ( Sound notification is not played )
#     ...     : 1 => Enable ( Sound notification is played )
#
# broker_module_webui_module_event_manager_reader_default_notifications_settings_sound_enabled 0

# Visual Notification Parameters  #

# Visual notification on new event
#
#     Default : 0 => Disable ( Visual notification is not played )
#     ...     : 1 => Enable ( Visual notification is played )
#
#
# broker_module_webui_module_event_manager_reader_default_notifications_settings_visual_blink_enabled 0

```

Paramétrage des notifications audio

```

# Sound Notification Parameters  #

# Sound notification on new event
#
#     Default : 0 => Disable ( Sound notification is not played )
#     ...     : 1 => Enable ( Sound notification is played )
#
# broker_module_webui_module_event_manager_reader_default_notifications_settings_sound_enabled 0

```

Nom	Type	Unité	Défaut	Description
broker_module_webui_module_event_manager_reader_default_notifications_settings_sound_enabled	Texte	---	0	Permet d'activer/désactiver la notification audio qui sera jouée lorsqu'un nouvel événement apparaît. Valeur possible : <ul style="list-style-type: none"> • 1 : Activé (La notification sonore est jouée). • 0 : Désactivé (La notification sonore n'est pas jouée).

Changement du son joué

Pour changer le son joué lors des notifications de changements (voir la page [Changer le son des notifications utilisé par l'Interface de visualisation](#))

Paramétrage des notifications visuelles

```

# Visual Notification Parameters  #

# Visual notification on new event
#
#     Default : 0 => Disable ( Visual notification is not played )
#     ...     : 1 => Enable ( Visual notification is played )
#
#
# broker_module_webui_module_event_manager_reader_default_notifications_settings_visual_blink_enabled 0

```

Nom	Type	Unité	Défaut	Description
broker__module_webui__module_event_manager_reader__default_notifications_settings_visual_blink_enabled	Texte	---	0	Permet d'activer/désactiver la notification visuelle qui sera jouée lorsqu'un nouvel événement apparaît. Valeur possible : <ul style="list-style-type: none"> 1 : Activé (<i>La notification visuelle est jouée</i>). 0 : Désactivé (<i>La notification visuelle n'est pas jouée</i>).

Options Internes

```
# #
# INTERNAL OPTIONS #
# #

# INTERNAL : DO NOT EDIT FOLLOWING PARAMETER WITHOUT YOUR DEDICATED SUPPORT
# Broker idle time before considering that Shinken is inactive.
# Use this if you have Broker loop time that exceeds 30 seconds
#
# Default : 30 ( seconds )
#
# time_before_shinken_inactive 30
```



Ces paramètres sont dédiés au fonctionnement interne au module, il est fortement recommandé de ne pas les modifier sans le support dédié.

Nom	Type	Unité	Défaut	Description
time_before_shinken_inactive	Entier	Seconde	30	Temps d'inactivité du Broker avant de considérer que Shinken est inactif. Utiliser cette option si le temps de boucle du Broker dépasse 30 secondes.