

shinken-synchronizer-database-restore -- Restaurer une sauvegarde automatique de la base de données du Synchronizer

v

Sommaire

- Concept
 - Utilisation de la commande
 - Options de de paramétrage des sauvegardes à charger
 - Options de connexion à la base MongoDB
 - Options génériques
 - Options de connexion SSH
 - Options d'authentification
 - Options SSL/TLS
- Exemple

Concept

Le module **webui--module-report-handler** permet au module **WebUI** (*l'interface de visualisation*) du **Broker** de générer des rapports (voir la page [Les rapports](#)) ou de déporter leur gestion.

- Ce module n'est disponible que depuis la version 02.08.02-RC018.
- Auparavant, c'est le module WebUI qui générait les rapports.
- À partir de la 02.08.02-RC018, le module **webui--module-report-handler** est indispensable pour la génération des rapports, il rend disponible la section **Rapports** dans le menu **Les Synthèses** de l'Interface de Visualisation..

Activation du module

Les modules de type "webui__module_report_handler" 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 "webui__module_report_handler".
- Il est possible de faire plusieurs modules de type "webui__module_report_handler".
 - 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 "webui__module_report_handler" dans la configuration de chacun d'eux.
- Contraintes :
 - Activable uniquement sur un module de type "webui" (voir la page [Configuration du module WebUI](#)).
 - Il ne peut y avoir qu'un seul module de type "webui__module_report_handler" 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 "webui--module-report-handler" sur le module nommé "WebUI" (configuration livrée par défaut par Shinken)

L'exemple suivant

- active le module "webui--module-report-handler" ,
- sur le module "WebUI" , dont la configuration est dans le fichier **/etc/shinken/modules/webui.cfg**.

Modification dans le fichier du module **/etc/shinken/modules/webui.cfg** :

```

define module {
    [...]
    modules          Module 1, Module 2, Module 3, webui--module-report-handler
    [...]
}

```

Puis redémarrage de l'Arbiter

```
service shinken-arbiter restart
```

Créer un nouveau module de type `webui__module_report_handler`

Définir plusieurs modules de type `webui__module_report_handler` permet d'avoir une configuration différente pour chaque module de type `webui`.

En fonction du besoin, il n'est pas nécessaire de créer plusieurs modules de type `webui__module_report_handler`.

Exemple :

- **Les modules de type `webui` génèrent leurs rapports directement** : Il vaut mieux définir un seul module de type `webui__module_report_handler` puis le déclarer dans tous les modules de type `webui`.
- **Les modules de type `webui` dans des royaumes différents génèrent leurs rapports en passant par des modules externes** : Il vaut mieux définir plusieurs modules de type `webui__module_report_handler` (*par royaume*) afin d'avoir des configurations adaptées pour chaque module de type `webui` (*voir la page [Partage de charge lors de la génération des rapports SLA depuis le module WebUI](#)*).

Pour pouvoir configurer un module de type "`webui--module-report-handler`", 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-webui--module-report-handler`".
 - Remplacer dans l'exemple le mot "`Mon-Module-webui--module-report-handler`" 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/webui__module_report_handler/webui--module-report-handler-example.cfg` dans le répertoire de définition des modules `/etc/shinken/modules/`.
(Exemple : `/etc/shinken/modules/webui__Mon-Module-webui--module-report-handler.cfg`)

```

cp /etc/shinken-user-example/configuration/daemons/brokers/modules/webui/modules
/webui__module_report_handler/webui--module-report-handler-example.cfg /etc/shinken/modules
/webui__Mon-Module-webui--module-report-handler.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 suivantes :

```

chown shinken:shinken /etc/shinken/modules/webui__Mon-Module-webui--module-report-handler.cfg
chmod +w /etc/shinken/modules/webui__Mon-Module-webui--module-report-handler.cfg

```

- On change le nom du module en "`Mon-Module-webui--module-report-handler`" dans le fichier `/etc/shinken/modules/webui__Mon-Module-webui--module-report-handler.cfg`

```

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

    module_name          Mon-Module-webui--module-report-handler
...

```

- Ensuite, il faut ajouter le nouveau module dans le module de type "webui" correspondant.
 - Dans notre exemple, on ajoute le module "Mon-Module-webui--module-report-handler" au module "webui" défini dans le fichier /etc/shinken/modules/webui.cfg

```
define module {
    [...]
    modules                               Module 1, Module 2, Module 3,
    Mon-Module-webui--module-report-handler
    [...]
}
```

- Puis pour finir, il faut redémarrer l'Arbiter pour que les modifications apportées à la configuration soient appliquées.

```
service shinken-arbiter restart
```