

# Module receiver-module-webservice

## Sommaire

- Concept
- Activation du module
- Configuration
  - 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
    - Authentification HTTP
- Utilisation du module

## Concept

Le module **receiver-module-webservice** expose une API permettant de transmettre des résultats à Shinken via des requêtes HTTP. Il offre les fonctionnalités suivantes :

- **Transmission des résultats de check** pour un hôte ou un check dont le mode passif est activé ( voir la page [Mode actif et mode passif](#) ).
- **Gestion des contextes** :
  - Définir une période de maintenance.
  - Indiquer une prise en compte.
- **Forcer la vérification du statut d'un élément.**

Ce module met à disposition un serveur HTTP qui écoute les requêtes POST et réalise des actions sur les hôtes, clusters et checks.

- Le service écoute par défaut sur le port TCP **7760**.
- Il supporte à la fois les accès authentifiés et anonymes.

## Activation du module

Le module `receiver-module-webservice` est un module qui peut être activé seulement sur un démon Receiver.

- L'activation du module s'effectue en ajoutant le nom de ce module dans le fichier de configuration du démon Receiver.
- Pour cela, ouvrir le fichier de configuration du Receiver à l'emplacement `/etc/shinken/receivers/nom_du_receiver.cfg`, et ajouter le nom du module "receiver-module-webservice" dans la liste des modules.

Exemple : par défaut, Shinken livre un module dont le nom est "receiver-module-webservice" :

```
define receiver {
    [...]
    modules          Module 1, Module 2, Module 3, receiver-module-webservice
    [...]
}
```

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

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

## Configuration

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

- Un exemple est systématiquement disponible dans `/etc/shinken-user-example/configuration/daemons/receiver/receiver-module-webservice/receiver-module-webservice-example.cfg`

## Exemple de fichier de configuration

```

#=====
# Webservice-Receiver (webservice)
#=====
# Daemons that can load this module:
# - receiver
# This module is a webservice that can be used to send checks to Shinken Enterprise
# as POST HTTP(s)
#=====

define module {

    #===== Module identity =====
    # Module name. Must be unique
    module_name      receiver-module-webservice
    # Module type (to load module code). Do not edit.
    module_type      ws_arbiter

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

    # port to listen
    port              7760

    # HTTPs part, enable if you want to set the listening for HTTPS instead of default HTTP.
    # disabled by default. Set your own certificates.
    use_ssl           0
    ssl_cert          /etc/shinken/certs/server.cert
    ssl_key           /etc/shinken/certs/server.key

    #===== HTTP authentication =====
    # You can use HTTP basic authentication method for this module.
    # If username is set to anonymous and password is commented, then
    # no authentication will be required.
    username          anonymous
    #password         secret

}

```

## Détails des sections composant le fichier de configuration

### Identification du module

Il est possible de définir plusieurs instances de module `receiver-module-webservice` dans l'architecture Shinken.

- Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Description
<code>module_name</code>	Texte	---	<code>receiver-module-webservice</code>	Shinken conseille de choisir un nom en fonction de l'utilisation du module pour que la configuration soit simple à maintenir. Doit être unique.
<code>module_type</code>	Texte	---	<code>ws_arbiter</code>	Ne peut être modifié.

### Configuration de l'interface et du port d'écoute

```

#==== Listening address =====
# host: IPv4 address to listen to.
# note: 0.0.0.0 = all interfaces.
host 0.0.0.0

# port to listen
port 7760

```

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

Les paramètres suivants permettent de configurer l'accès à l'API :

Nom	Type	Unité	Défaut	Description
host	Texte	Adresse IPv4	0.0.0.0	L'interface réseau sur lequel le module "receiver-module-webservice" va écouter. La valeur 0.0.0.0 correspond à toutes les interfaces.
port	Texte	Port réseau	7760	Port réseau sur lequel le module "receiver-module-webservice" va écouter.

## Sécurisation de la communication avec le module

```

# HTTPs part, enable if you want to set the listening for HTTPS instead of default HTTP.
# 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'API du module 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 API doit être accessible via HTTPS, il est possible de configurer les certificats avec les paramètres suivants :

Nom	Type	Unité	Défaut	Description
use_ssl	Booléen	---	0	Permet d'activer ou non l'utilisation du protocole HTTPS. <ul style="list-style-type: none"> <li>0 : Désactivé ( <i>utilise HTTP</i> )</li> <li>1 : Activé ( <i>utilise HTTPS</i> )</li> </ul>
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.

## Authentification HTTP

```

#==== HTTP authentication =====
# You can use HTTP basic authentication method for this module.
# If username is set to anonymous and password is commented, then
# no authentication will be required.
username anonymous
#password secret

```

Afin d'être authentifié auprès du module, il est possible de configurer les identifiants à utiliser dans les requêtes, dans la configuration du module grâce aux paramètres suivants :

Nom	Type	Unité	Défaut	Description
username	Texte	---	anonymous	Nom d'utilisateur.
password	Texte	---	secret	Mot de passe.



Lors d'une requête cURL, les identifiants se renseignent grâce à l'option `-u`

```
curl -u anonymous:secret -X POST ...
```

## Utilisation du module

Parmi les cas d'utilisation documentés par Shinken, ce module peut notamment être employé pour convertir des traps SNMP en retours interprétables par Shinken ( voir la page [Script d'interprétation des traps avec le module receiver-module-webservice](#) )