

# V1 - Les API de écouteur de type listener-rest

## Sommaire

- [Introduction](#)
- [Format des URL](#)
- [Sécuriser l'écouteur](#)
  - [Authentification](#)
  - [Chiffrement des appels](#)
- [Description des actions](#)
  - [Obtenir la liste des hôtes créés par l'écouteur](#)
  - [Créer un hôte](#)
  - [Obtenir les données d'un hôte](#)
  - [Modifier un hôte](#)
  - [Supprimer un hôte créé par l'écouteur](#)
- [Forcer la valeur des propriétés](#)
  - [Liste des clés d'import pouvant être forcées](#)

## Introduction

L'API fournie par les écouteurs est de type REST ; elle fonctionne donc par l'intermédiaire des différentes méthodes du protocole HTTP :


- GET : Obtenir des informations sur un élément, ou la liste des éléments.
- PUT : Créer ou modifier un élément.
- POST : Modifier un élément.
- PATCH : Modifier un élément.
- DELETE : Supprimer un élément.

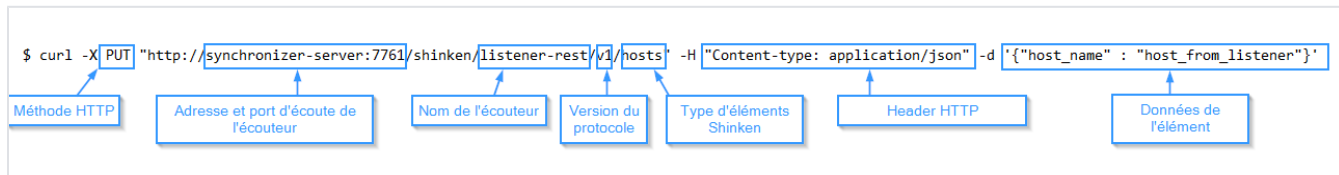
L'identifiant des éléments est fourni dans l'URL et les données supplémentaires, quand cela est nécessaire, sont fournies dans le corps de la requête HTTP au format JSON.

Cette page décrit le format des URL ainsi que les différentes actions réalisables.

## Format des URL


Les exemples utilisent la commande `curl`, disponible dans la plupart des distributions Linux ; cependant la requête HTTP peut être réalisée avec n'importe quel outil.

 Cet écouteur ne permet de travailler que sur les hôtes.



Une requête est composée de plusieurs éléments :

- Méthode HTTP : l'action que l'écouteur doit effectuer ; les actions disponibles sont décrites en détail dans les sections suivantes.
- Adresse et port de l'écouteur : l'adresse est celle du Synchronizer, le port du listener REST fourni avec Shinken Enterprise est le **7761**.
- Nom de l'écouteur : il s'agit du nom de l'écouteur tel que défini dans la propriété `listener_name` du fichier de définition de l'écouteur ; le listener fourni avec Shinken Enterprise est `listener-rest`.
- Version du protocole : dans cette version de Shinken Enterprise, il s'agit de la **v1** ; si le protocole évolue dans de futures versions, le numéro de version évoluera également, ce qui permettra à l'écouteur de conserver la compatibilité avec la version actuelle.
- Type d'éléments Shinken : le type d'élément Shinken traité par la requête.


 Actuellement, seuls les hôtes sont gérés.

- Header HTTP : La liste des en-têtes HTTP ; dans le cas de l'écouteur, il faut spécifier que le contenu sera du JSON.
- Données de l'élément : Les données décrivant l'élément à importer respectent le format JSON. Il s'agit d'un dictionnaire décrivant toutes les propriétés et données de l'élément.
  - Elles sont définies à l'aide des noms de propriétés décrites dans la page [Syntaxe des fichiers d'imports](#).
  - Des exemples de requêtes sont disponibles plus bas dans la page.

## Sécuriser l'écouteur

Par défaut, l'écouteur transmet et reçoit les requêtes en clair ( *non chiffrées* ) et sans authentification. Il est possible de sécuriser ces communications avec :

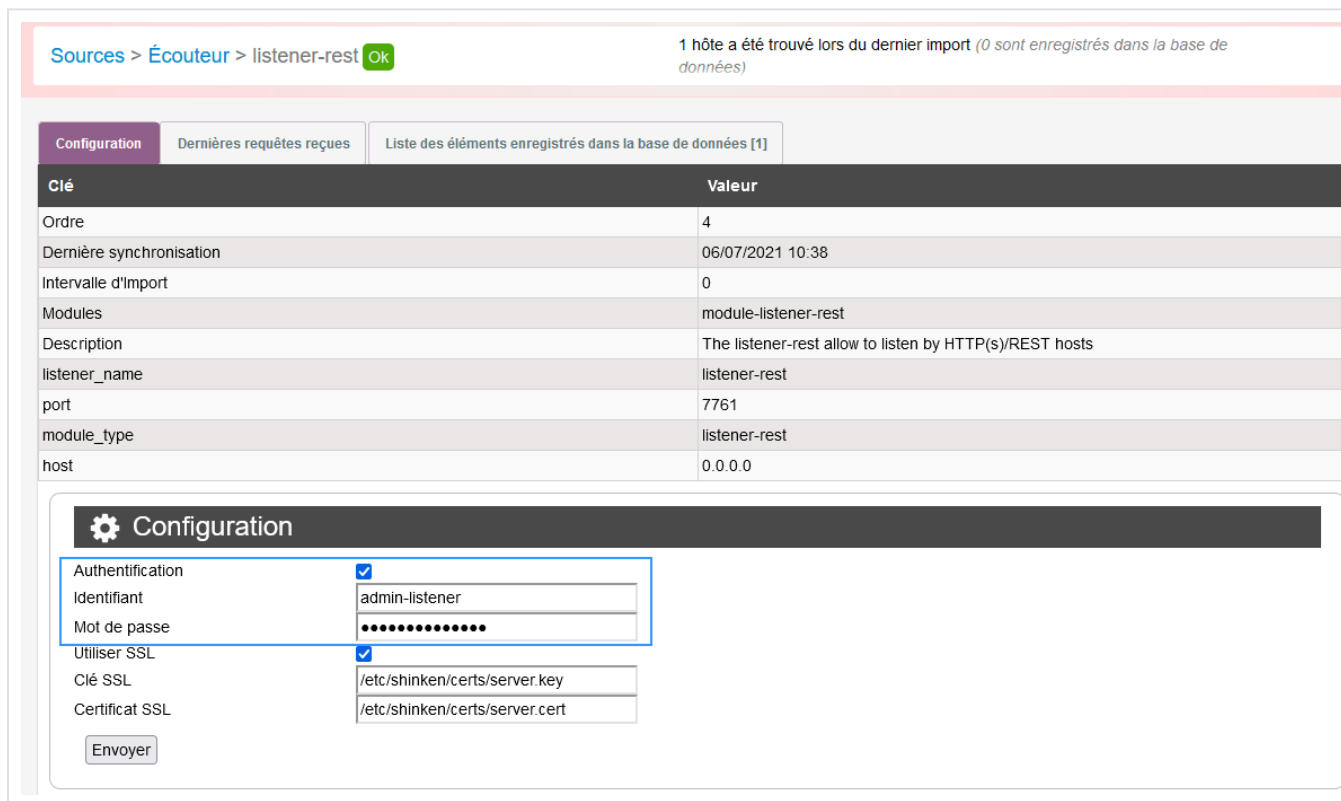
- Une authentification par identifiant/mot de passe ;
- Un chiffrement grâce à SSL ;

 Il est fortement conseillé d'activer ces paramètres de sécurité en même temps.

## Authentification

Il est possible d'authentifier les requêtes grâce à un couple identifiant/mot de passe, configurable uniquement sur la page de l'écouteur, dans l'onglet Configuration.

- Il suffit de cocher la case **Authentification** et de saisir un **identifiant** et un **mot de passe**, qui ne seront valables que pour cet écouteur.
- Pour valider cette configuration, il faudra cliquer sur le bouton **Envoyer**.
  - L'écouteur va se relancer avec cette nouvelle configuration ( *ou lors de l'activation de la source si celle-ci est désactivée au moment de l'édition* ).



The screenshot shows the configuration page for the 'listener-rest' source in the Shinken web interface. The breadcrumb navigation is 'Sources > Écouteur > listener-rest'. A status message indicates '1 hôte a été trouvé lors du dernier import (0 sont enregistrés dans la base de données)'. The configuration table lists various parameters such as 'Ordre', 'Dernière synchronisation', 'Intervalle d'Import', 'Modules', 'Description', 'listener\_name', 'port', 'module\_type', and 'host'. Below the table, the 'Configuration' section is expanded, showing the 'Authentification' checkbox checked, the 'Identifiant' field set to 'admin-listener', and the 'Mot de passe' field masked with dots. Other options like 'Utiliser SSL' are also checked, with fields for 'Clé SSL' and 'Certificat SSL'.

Clé	Valeur
Ordre	4
Dernière synchronisation	06/07/2021 10:38
Intervalle d'Import	0
Modules	module-listener-rest
Description	The listener-rest allow to listen by HTTP(s)/REST hosts
listener_name	listener-rest
port	7761
module_type	listener-rest
host	0.0.0.0

**Configuration**

Authentification

Identifiant

Mot de passe

Utiliser SSL

Clé SSL

Certificat SSL

Lors des appels à cet écouteur, il faudra spécifier cet identifiant et ce mot de passe pour que les requêtes s'exécutent.

Par exemple, avec **curl**, il faut utiliser l'option **-u** :

```
$ curl -u 'admin-listener:mot_de_passe' -X PUT ...
```

## Chiffrement des appels

Afin de chiffrer les communications entre un écouteur et un client, il suffit d'activer l'option SSL ( *Secure Sockets Layer* ).

Il suffit de cocher la case **Utiliser SSL** et de saisir une **clé SSL** et un **certificat SSL** qui ne seront valables que pour cet écouteur.



Shinken fournit un certificat SSL qui permet de chiffrer les communications pour tous les démons et/ou modules. Il se situe dans le répertoire `/etc/shinken/certs/`.

Il est aussi possible de créer son propre certificat, voir : [Création d'un certificat SSL gratuit](#).

Pour valider cette configuration, il faut cliquer sur le bouton **Envoyer**.

L'écouteur va se relancer avec cette nouvelle configuration ( *ou lors de l'activation de la source si celle-ci est désactivée au moment de l'édition* ).

Sources > Écouteur > listener-rest OK 1 hôte a été trouvé lors du dernier import (0 sont enregistrés dans la base de données)

Configuration | Dernières requêtes reçues | Liste des éléments enregistrés dans la base de données [1]

Clé	Valeur
Ordre	4
Dernière synchronisation	06/07/2021 10:38
Intervalle d'import	0
Modules	module-listener-rest
Description	The listener-rest allow to listen by HTTP(s)/REST hosts
listener_name	listener-rest
port	7761
module_type	listener-rest
host	0.0.0.0

### Configuration

Authentification

Identifiant

Mot de passe

Utiliser SSL

Clé SSL

Certificat SSL

Lors des appels à cet écouteur, il faudra penser à modifier le protocole HTTP en HTTPS.

```
$ curl -X PUT "https://synchronizer-server:7761/shinken/listener-rest/v1/hosts"
```



En utilisant le certificat fourni par Shinken ou un certificat auto-signé ( *non émis et vérifié par une autorité de certification* ), il sera nécessaire de spécifier l'option `-k` ou `--insecure` dans la ligne de commande curl.



En utilisant un certificat d'une autorité de certification, il est nécessaire de le spécifier dans la ligne de commande curl avec l'option `--cacert` `<chemin du certificat de l'autorité>`.

## Description des actions

Dans ces différents exemples, les requêtes seront :

- Non authentifiées.
- Sans le chiffrement ( `SSL` ).
- Lancées depuis le serveur hébergeant le Synchronizer : localhost.
- Utilisant le collecteur par défaut : listener-rest.
- Utilisant le port par défaut : 7761.

L'URL utilisée sera donc :

```
http://localhost:7761/shinken/listener-rest/v1/hosts/
```



Il faudra adapter cette URL en fonction de la configuration.



Afin d'avoir un affichage plus esthétique des résultats de la méthode **GET**, il faut rediriger la sortie vers Python et utiliser le module `json.tool` grâce à la commande suivante à ajouter en fin de ligne :

```
curl ... | python -m json.tool
```

## Obtenir la liste des hôtes créés par l'écouteur

Pour obtenir la liste des hôtes présents dans l'écouteur via l'appel `/shinken/listener-rest/v1/hosts`, et la méthode **GET** :

- Voir la page [V1 - \( GET \) /shinken/listener-rest/v1/hosts](#) ( liste des hôtes présents dans l'écouteur ).

## Créer un hôte

Pour créer un hôte via l'appel `/shinken/listener-rest/v1/hosts`, et la méthode **PUT** :

- Voir la page [V1 - \( PUT \) /shinken/listener-rest/v1/hosts](#) ( création d'hôte ).

## Obtenir les données d'un hôte

Pour obtenir les informations définissant un hôte via l'appel `/shinken/listener-rest/v1/hosts/SE_UUID`, et la méthode **GET** :

- Voir la page [V1 - \( GET \) /shinken/listener-rest/v1/hosts/SE\\_UUID](#) ( Obtenir les données d'un hôte ).

## Modifier un hôte

Pour modifier un hôte via l'appel `/shinken/listener-rest/v1/hosts/SE_UUID`, et l'une des méthodes **POST** ou **PUT** ou **PATCH** :

- Voir la page [V1 - \( POST, PUT, PATCH \) /shinken/listener-rest/v1/hosts/SE\\_UUID](#) ( modification de l'hôte ).

## Supprimer un hôte créé par l'écouteur

Pour supprimer un hôte via l'appel `/shinken/listener-rest/v1/hosts/SE_UUID`, et la méthode **DELETE** :

- Voir la page [V1 - \( DELETE \) /shinken/listener-rest/v1/hosts/SE\\_UUID](#) ( Supprimer un hôte créé par l'écouteur ).

## Forcer la valeur des propriétés

Il est possible d'ajouter le suffixe `[FORCE]` ( *sans espace* ) à la **clé d'import** d'un élément Shinken afin de forcer la valeur associée ( voir la page [Forcer la valeur des noms des éléments et des propriétés de type liste](#) (comme la propriété des modèles hérités) ).

Exemple :

```
# Modifie l'hôte avec le SE_UUID core-hosts-a476af2aae9d4cdab85e0c439ee93bee en forçant l'adresse et les
modèles qui seront appliqués à l'hôte
curl -H "Content-Type: application/json" "http://localhost:7761/shinken/listener-rest/v1/hosts/core-hosts-
a476af2aae9d4cdab85e0c439ee93bee" -X POST -d '{ "host_name": "host_example", "address[FORCE]": "127.0.0.1",
"use[FORCE]": "linux, mongodb" }'
```

Lorsqu'une propriété est marquée avec `[FORCE]`, elle est traitée de la manière suivante par le mélange des sources :

- La valeur est considérée comme prioritaire par rapport aux autres sources.
- Si une autre source importe le même objet avec une valeur différente, cette dernière est ignorée.
- En cas de conflit entre plusieurs valeurs forcées, la priorité de la source détermine la valeur conservée.

Ajouter `[FORCE]` à une clé d'import contenant une liste ( comme la propriété `"members"` servant à définir les membres d'un groupe d'utilisateurs ) aura pour effet de remplacer complètement la liste des autres sources par celle fournie avec l'option `[FORCE]`.

## Liste des clés d'import pouvant être forcées

- Les clés d'import définissant les noms des éléments
  - Exemple : "host\_name" pour les Hôtes.
- Et toutes les propriétés pouvant avoir des valeurs multiples : ( voir la page [Éditer les Eléments](#) ( hôte, clusters, checks, utilisateurs ... ) )
  - Exemple : "view\_contacts" pour la propriété "Les utilisateurs qui voient l'hôte".

La liste exhaustive de toutes les clé est disponible ici : [Liste des clés pouvant utiliser \[FORCE\]](#)