

API REST: Gérer des hôtes

Sommaire

- [Introduction](#)
- [Format des URL](#)
- [Sécuriser l'écouteur](#)
 - [Authentification](#)
 - [Chiffrement des appels](#)
- [Description des actions](#)
 - [Obtenir la liste des hôtes créés par l'écouteur](#)
 - [Définition](#)
 - [Réponse](#)
 - [Exemple](#)
 - [Créer un hôte](#)
 - [Définition](#)
 - [Réponse](#)
 - [Exemple](#)
 - [Obtenir les données d'un hôte](#)
 - [Définition](#)
 - [Réponse](#)
 - [Exemple](#)
 - [Modifier un hôte](#)
 - [Définition](#)
 - [Réponse](#)
 - [Exemple](#)
 - [Supprimer un hôte créé par l'écouteur](#)
 - [Définition](#)
 - [Réponse](#)
 - [Exemple](#)
- [Questions fréquentes sur l'import d'éléments via l'API REST](#)
 - [Forcer l'utilisation d'un modèle](#)

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 en format JSON.

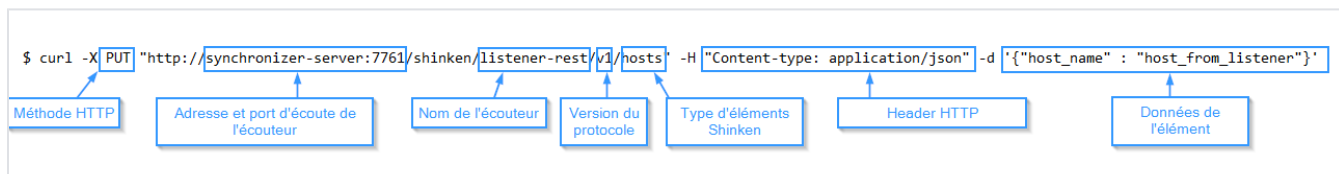


Notez que cet écouteur ne permet de travailler que sur les hôtes.

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 l'outil de votre choix.



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 Entreprise 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 Entreprise est **listener-rest**
- Version du protocole : dans cette version de Shinken Entreprise, 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, nous spécifions que nous envoyons du contenu 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 de la même manière que dans les fichiers CFG (voir page [Syntaxe des fichiers d'imports](#)).

Sécuriser l'écouteur

Par défaut, l'écouteur transmet et reçoit les requêtes en clair (non chiffrée) et sans authentification. Vous pouvez 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 saisir un **identifiant** et un **mot de passe**, qui ne seront valables que pour cet écouteur.

Validez ensuite cette configuration en cliquant 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 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 Socket Layer).

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

i 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 faire son propre certificat (voir la page [Génération d'un certificat SSL gratuit](#)).

Validez ensuite cette configuration en cliquant 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

Authentication

Identifiant

Mot de passe

Utiliser SSL

Clé SSL

Certificat SSL

Envoyer

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"
```

i Si vous utilisez le certificat fourni par Shinken ou un certificat auto signé (non émis et vérifiés par une autorité de certification), vous aurez besoin de spécifier l'option `-k` ou `--insecure` dans la ligne de commande curl

i Si vous avez le certificat de l'autorité de certification vous pouvez 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, nous allons utiliser des des requêtes :

- 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 utilisé sera donc

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

Pensez à adapter cette URL en fonction de votre configuration

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

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

Obtenir la liste des hôtes créés par l'écouteur

Définition

URL à utiliser	/shinken/listener-rest/v1/hosts
Méthodes possibles	GET

Cette URL, associée à la méthode GET, permet d'obtenir la liste de tous les éléments présents dans l'écouteur. Le format de retour est un JSON

Les éléments présents dans l'écouteur sont ceux qui ont été créés sur cet écouteur (et qui n'ont pas été supprimés).

Les propriétés et données Shinken affichées sont celles reçues par le collecteur.

i Cet appel correspond à l'onglet "Liste des éléments enregistrés dans la base de données" sur la page de description de votre source.

Réponse

Codes de retour	Contenus de retour	Explications
200	Liste des éléments au format JSON	L'appel s'est déroulé avec succès
500	Internal Server Error	L'appel est valide, mais un problème d'exécution est survenu.

Exemple

```
$ curl -H "Content-Type: application/json" "http://localhost:7761/shinken/listener-rest/v1/hosts/" -X GET |  
python -m json.tool  
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
           % Done      0         0     100k      0  ---:--:--  ---:--:--  101k  
[  
  {  
    "_SE_UUID": "core-hosts-a0af58fafdfb4db5b0fe07c704caf937",  
    "_SYNC_KEYS": "core-hosts-a0af58fafdfb4db5b0fe07c704caf937,host_example_01",  
    "_id": "a0af58fafdfb4db5b0fe07c704caf937",  
    "address": "192.168.1.48",  
    "host_name": "host_example_01",  
    "imported_from": "listener-rest envoy\u00e9 depuis l'adresse 127.0.0.1",  
    "update_date": 1625570835.448688,  
    "use": "linux"  
  },  
  {  
    "_SE_UUID": "core-hosts-3941f2a530194d149f18b82f84bcef51",  
    "_SYNC_KEYS": "core-hosts-3941f2a530194d149f18b82f84bcef51,host_example_02",  
    "_id": "3941f2a530194d149f18b82f84bcef51",  
    "address": "192.168.1.54",
```

```
    "host_name": "host_example_02",
    "imported_from": "listener-rest envoy\u00e9 depuis l'adresse 127.0.0.1",
    "update_date": 1625570975.763321,
    "use": "windows"
  }
}
```



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

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

Créer un hôte

Définition

URL à utiliser	/shinken/listener-rest/v1/hosts
Méthodes possibles	PUT

Cette route permet d'envoyer à l'écouteur un nouvel élément.

- Si le nom entré correspond à un hôte présent dans l'écouteur, dans la Zone de Travail, ou dans Staging, l'élément sera envoyé avec le même SE_UUID que l'hôte trouvé.
- L'écouteur renvoie l'identifiant de l'élément créé sous la forme "core-hosts-<ID>" .

Réponse

Codes de retour	Contenus de retour	Explications
201	SE_UUID de l'élément créé	La création s'est déroulée avec succès
500	Internal Server Error	L'appel est valide, mais un problème d'exécution est survenu.

Exemple

```
$ curl -H "Content-Type: application/json" "http://localhost:7761/shinken/listener-rest/v1/hosts/" -X PUT -d '{ "host_name": "host_example_01", "address": "192.168.1.48", "use": "linux" }'
"core-hosts-a0af58fafdfb4db5b0fe07c704caf937"
```

Obtenir les données d'un hôte

Définition

URL à utiliser	/shinken/listener-rest/v1/hosts/SE_UUID
Méthodes possibles	GET

La méthode **GET**, lorsqu'elle est accompagnée d'un identifiant, l'élément est renvoyé au format JSON.

Réponse

Codes de retour	Contenus de retour	Explications
200	Propriétés et Données Shinken de l'élément reçues sur le collecteur au format JSON	L'appel s'est déroulé avec succès

404	"Aucun hôte avec le _SE_UUID SE_UUID "	L'hôte demandé n'existe pas
500	Internal Server Error	L'appel est valide, mais un problème d'exécution est survenu.

Exemple

 Le **_SE_UUID** présent dans l'URL est clef de l'hôte créé dans le premier appel de création. Pensez à mettre le votre !


```
$ curl -H "Content-Type: application/json" "http://localhost:7761/shinken/listener-rest/v1/hosts/core-hosts-a0af58fafdfb4db5b0fe07c704caf937" -X GET | python -m json.tool
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100  361  100  361    0    0  51046      0  --:--:-- --:--:-- --:--:-- 60166
{
  "_SE_UUID": "core-hosts-a0af58fafdfb4db5b0fe07c704caf937",
  "_SYNC_KEYS": "core-hosts-a0af58fafdfb4db5b0fe07c704caf937,host_example_01",
  "_id": "a0af58fafdfb4db5b0fe07c704caf937",
  "address": "192.168.1.48",
  "host_name": "host_example_01",
  "imported_from": "listener-rest envoy\u00e9 depuis l'adresse 127.0.0.1",
  "update_date": 1625570835.448688,
  "use": "linux"
}
```

Modifier un hôte

Définition

URL à utiliser	/shinken/listener-rest/v1/hosts/ SE_UUID
Méthodes possibles	POST, PUT, PATCH

Vous pouvez renseigner le **_SE_UUID** d'un hôte de l'écouteur, de la Zone de Travail ou bien de Staging.

 Si l'hôte trouvé est que dans la Zone de Travail ou dans Staging, le nom de l'hôte sera ajouté aux données envoyées si le nom n'était pas déjà présent.

Réponse

Codes de retour	Contenus de retour	Explications
200	Le mot clé "done"	La modification s'est déroulée avec succès
404	"Aucun hôte avec le _SE_UUID SE_UUID "	L'hôte demandé n'existe pas
500	Internal Server Error	L'appel est valide, mais un problème d'exécution est survenu.

Exemple

On a un hôte avec le **_SE_UUID** "core-hosts-a0af58fafdfb4db5b0fe07c704caf937" et on exécute cette commande :

```
$ curl -H "Content-Type: application/json" "http://localhost:7761/shinken/listener-rest/v1/hosts/core-hosts-a0af58fafdfb4db5b0fe07c704caf937" -X POST -d '{"business_impact": "4"}'
done
```

Supprimer un hôte créé par l'écouteur

Définition

URL à utiliser	/shinken/listener-rest/v1/hosts/ SE_UUID
-----------------------	-------------------------------------------------

Méthodes possibles DELETE

Utilisez l'action **DELETE** pour supprimer un élément créé par l'écouteur, en fournissant l'identifiant de l'élément à supprimer.



Notez que cette requête permet de supprimer les éléments dans la base de l'écouteur ; elle ne permet pas de supprimer un élément dans le Synchronizer lui-même, pour des raisons de sécurité.

Réponse

Codes de retour	Contenus de retour	Explications
200	Le mot clé "done"	La suppression s'est déroulée avec succès
404	"Aucun hôte avec le _SE_UUID SE_UUID"	L'hôte demandé n'existe pas
500	Internal Server Error	L'appel est valide, mais un problème d'exécution est survenu.

Exemple

```
$ curl -X DELETE http://localhost:7761/shinken/listener-rest/v1/hosts/core-hosts-22b48e7408a549f6895badada2fd3690
done
```

Questions fréquentes sur l'import d'éléments via l'API REST

Forcer l'utilisation d'un modèle

Lorsqu'on importe un hôte existant qui contient la propriété "use", la liste des modèle décrits dans la propriété "use" est ajoutée à la liste des modèles actuellement utilisée par l'hôte.

Comme lors de l'import d'hôtes via fichiers Cfg, il est possible de forcer l'utilisation d'une liste de modèles (remplacement et non ajout de modèles) en utilisant la propriété "use[FORCE]" au lieu de "use".

Plus de précisions sur l'utilisation du FORCE pour les modèles sont disponibles dans la page de documentation (voir la page [Syntaxe des fichiers d'imports](#))