

API REST: Gérer des hôtes ou des clusters

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 objet, ou la liste des objets
- PUT : Créer ou modifier un objet
- POST : Modifier un objet
- DELETE : Supprimer un objet

L'identifiant des objets est fournie dans l'URL, et les données supplémentaires, quand cela est nécessaire, sont fournies dans le corps de la requête HTTP sous la forme d'un objet JSON.

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

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

Description des actions

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.

Description générale d'une requête

```
$ curl -X PUT 'http://synchronizer-server:7761/shinken/listener-rest/v1/hosts/' -H "Content-type: application/json" -d '{"host_name": "host_from_listener"}
```

Le diagramme ci-dessus illustre la structure d'une requête curl avec des annotations pointant vers les différents composants de la commande :

- Action HTTP** : Pointe vers `-X PUT`
- Adresse et port d'écoute du Listener** : Pointe vers `'http://synchronizer-server:7761'`
- Nom du listener** : Pointe vers `/shinken/listener-rest`
- Versión du protocole** : Pointe vers `/v1`
- Type d'objet Shinken** : Pointe vers `/hosts/`
- Header HTTP : On envoie du contenu JSON** : Pointe vers `-H "Content-type: application/json"`
- Données de l'objet en JSON** : Pointe vers `-d '{"host_name": "host_from_listener"}'`

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 d'écoute de l'écouteur : l'adresse est celle du Synchronizer, le port du listener REST fourni avec Shinken Entreprise est le **7761**
- Nom du listener : 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'objet Shinken : le type d'objet 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'objet : Les données décrivant l'objet à importer respectent le format JSON. Il s'agit d'un dictionnaire décrivant toutes les propriétés de l'objet ; Elles sont définies de la même manière que dans les fichiers CFG et décrites [sur cette page](#).

Si l'authentification HTTP est configurée dans le listener, vous devez fournir les informations d'authentification lors de l'exécution de la requête HTTP. Avec curl, il s'agit de l'option **-u** :

```
$ curl -u 'root:root' -X PUT ....
```

Si l'authentification SSL est configurée dans le listener, vous devez fournir le certificat correspondant lors de la requête. Avec curl, il s'agit de l'option **--cacert** :

```
$ curl --cacert certificate.crt -X PUT ...
```

Sources > listener-rest **OK**

Configuration Résumé des dernières exécutions Liste des éléments

Clié

Ordre

Dernière synchronisation

Intervalle d'import

Modules

Description

Interface d'écoute

Port d'écoute

Authentication HTTP

Configuration

Authentication

Identifiant

Mot de passe

Utiliser SSL

Clié SSL

Certificat SSL

Envoyer

Créer un hôte / cluster

Pour mener à bien cette action, vous pouvez utiliser les méthodes **PUT** ou **POST** qui permettent d'envoyer à l'écouteur un nouvel objet

- Lors de la création d'un nouvel objet, l'écouteur renvoie l'identifiant de l'objet créé sous la forme "core-hosts-<ID>". Cet identifiant devra être utilisé lors d'autres opérations manipulant l'objet :

```
$ curl -u 'root:root' -X PUT http://localhost:7761/shinken/listener-rest/v1/hosts -d '{ "host_name": "hote-example", "use":"linux", "_SSH_USER":"root", "_SSH_KEY_PASSPHRASE":"root" }'
"core-hosts-22b48e7408a549f6895badada2fd3690"
```

Modifier un hôte / cluster

- Pour modifier un objet existant, utilisez la méthode **POST** en indiquant son identifiant dans l'URL. Les propriétés présentes dans l'objet JSON transmis dans la requête remplaceront ou compléteront alors celles se trouvant dans l'objet. Les autres propriétés ne sont pas modifiées.

```
$ curl -u 'root:root' -X POST http://localhost:7761/shinken/listener-rest/v1/hosts/core-hosts-22b48e7408a549f6895badada2fd3690 -d '{ "address": "192.168.1.42" }'
done
```

Obtenir les données d'un hôte / cluster

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

```
$ curl -u 'root:root' http://localhost:7761/shinken/listener-rest/v1/hosts/core-hosts-22b48e7408a549f6895badada2fd3690 | python -m json.tool
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100    381    100    381     0     0  72668      0 --:--:-- --:--:-- --:--:--  76200
{
  "_SE_UUID": "core-hosts-22b48e7408a549f6895badada2fd3690",
  "_SSH_KEY_PASSPHRASE": "root",
  "_SSH_USER": "root",
  "_SYNC_KEYS": "core-hosts-22b48e7408a549f6895badada2fd3690,hote-example",
  "_id": "22b48e7408a549f6895badada2fd3690",
  "host_name": "hote-example",
  "imported_from": "listener-rest envoy\u00e9 depuis l'adresse 127.0.0.1",
  "update_date": 1527839180.6257291,
  "use": "linux"
}
```

Obtenir la liste des hôtes / clusters créés par l'écouteur

La méthode **GET**, lorsque la requête ne contient pas d'identifiant, renvoie la liste de tous les objets créés par l'écouteur, au format JSON.

```
$ curl -u 'root:root' http://localhost:7761/shinken/listener-rest/v1/hosts/ | python -m json.tool
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100    785    100    785     0     0  104k      0 --:--:-- --:--:-- --:--:--  109k
[
  {
    "_SE_UUID": "core-hosts-22b48e7408a549f6895badada2fd3690",
    "_SSH_KEY_PASSPHRASE": "root",
    "_SSH_USER": "root",
    "_SYNC_KEYS": "core-hosts-22b48e7408a549f6895badada2fd3690,hote-example",
    "_id": "22b48e7408a549f6895badada2fd3690",
    "host_name": "hote-example",
    "imported_from": "listener-rest envoy\u00e9 depuis l'adresse 127.0.0.1",
    "update_date": 1527839180.6257291,
    "use": "linux"
  },
  {
    "_SE_UUID": "core-hosts-bafe4fae3fc447b992daa3f10ee260f2",
    "_SYNC_KEYS": "core-hosts-bafe4fae3fc447b992daa3f10ee260f2,hote-example-2",
    "_id": "bafe4fae3fc447b992daa3f10ee260f2",
    "address": "192.168.0.42",
    "host_name": "hote-example-2",
    "imported_from": "listener-rest envoy\u00e9 depuis l'adresse 127.0.0.1",
    "update_date": 1527839954.469449,
    "use": "shinken-full"
  }
]
```

Supprimer un hôte / cluster créé par l'écouteur

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



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

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