

\$KEY\$ Usage Switch SNMPv1v2

Introduction

Le module **broker-module-livodata** permet de mettre à disposition sur le Broker une API HTTP permettant d'accéder aux informations d'un hôte, d'un cluster ou d'un check.

Cette API peut être utilisée pour accéder rapidement aux informations d'éléments supervisés pour ensuite les intégrer dans des outils externes (outil de ticketing, récolte de données, etc...).

Activation du module

Le module **broker-module-livodata** est un module qui peut être activé seulement sur un démon Broker.

L'activation du module s'effectue en ajoutant le module dans le fichier de configuration du Broker concerné.

Pour ce faire, dans le fichier de configuration de votre Broker à l'emplacement **/etc/shinken/brokers/nom_de_votre_broker.cfg** à la ligne suivante, et ajouter "*broker-module-livodata*"

```
define broker {
    [...]
    modules          Simple-log, WebUI, Graphite-Perfdata, sla,
    broker-module-livodata
    [...]
}
```

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

```
service shinken-arbiter restart
```

Configuration

La configuration du module se trouve par défaut dans le fichier suivant: **/etc/shinken/modules/broker-module-livodata.cfg**

Configuration de l'interface et du port d'écoute

Par défaut, le port de l'API rendue disponible par le module "*broker-module-livodata*" est **50100**. Ce port peut être changé via le paramètre "*port*" dans le fichier de configuration du module: **/etc/shinken/modules/broker-module-livodata.cfg**.

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

Cette modification de configuration se fait via le paramètre "*host*" du fichier de configuration du module: **/etc/shinken/modules/broker-module-livodata.cfg**. Dans l'exemple suivant, l'API ne sera disponible que sur l'interface avec l'adresse **192.168.1.27**:

```
/etc/shinken/modules/broker-module-livodata.cfg

define module {
    [...]
    host          192.168.1.27
    [...]
}
```

Sommaire

- Contexte
- Paramétrage
 - Données utilisées provenant du modèle
 - Données utilisées provenant du check
 - Données globales
 - Propriétés de l'hôte
- Résultat
 - Exemple
 - Interprétation
- Métriques

L'API du module est par défaut mise à disposition sur toutes les interfaces: le paramètre "*host*" est à *0.0.0.0*

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

```
service shinken-arbiter restart
```

Activation du SSL

L'API du module est accessible via HTTP. Si pour des raisons de sécurité, cette API doit être accessible via HTTPS, il faut passer le paramètre "*use_ssl*" à 1 dans le fichier de configuration du module:

/etc/shinken/modules/broker-module-livedata.cfg

```
define module {
    [...]
    # HTTPs part, enable if you want to set the visualisation interface listen in HTTPs mode
    # disabled by default. Set your own certificates. Set your own token, it is usefull to get access to the
    API
    use_ssl                1
    ssl_cert               /etc/shinken/certs/server.cert
    ssl_key                /etc/shinken/certs/server.key

    [...]
}
```

Les paramètres "*ssl_cert*" et "*ssl_key*" permettent de spécifier la clé et le certificat SSL à utiliser pour la connexion.

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

```
service shinken-arbiter restart
```

Authentification

L'API du module *broker-module-livedata* est accessible à l'adresse suivante:

```
http://<ip-broker>:<port-module>/
```

Les appels effectués à l'API demandent l'utilisation d'un token d'identification. Ce token est spécifié dans le fichier de configuration du module (**/etc/shinken/modules/broker-module-livedata.cfg**).

Il est fortement recommandé de changer ce token.

Ce token doit être ajouté en paramètre à chaque requête sous peine de recevoir une erreur 401 de la part de l'API.

Par exemple:

```
http://192.168.1.59:50100/api/v1/host/408cd29ad37611e8810e0800277b7e16?token=ak5zv6t5s25r6g4
```

Chaque appel à l'API renvoie une réponse au format JSON.



Remarque

Avec le module configuré en HTTP, une simple écoute du réseau permet de trouver le token. Il est fortement recommandé de passer en HTTPS afin d'éviter une interception du token.

Utilisation

Les routes suivantes sont mise à disposition par l'API. Chaque appel doit contenir dans ses paramètres le token d'identification décrit dans la section précédente.

GET /api/v1/all-monitored-elements

Cet appel permet de recevoir la liste des uuids de tous les hôtes et clusters ainsi que de leurs checks.

Réponse

Codes de retour

Codes de retour	Explications
200	OK
401	Permission refusée (Mauvais token)

Retour du code 200

- **hosts :**
 - **host_uuid :** text
 - **checks :**
 - **check_uuid :** text
- **clusters :**
 - **cluster_uuid :** text
 - **checks :**
 - **check_uuid :** text

Exemple

<http://localhost:50100/api/v1/all-monitored-elements>

```
{
  "clusters": [
    {
      "cluster_uuid": "ac169cd1fb5744999eaadc5da595f88b",
      "checks": [
      ]
    },
    {
      "cluster_uuid": "0dcad2b1667642d28c933de70915331e",
      "checks": [
        "0dcad2b1667642d28c933de70915331e-905374c0f6ed11e78cd83cf862f4613a",
        "0dcad2b1667642d28c933de70915331e-714f6a1b90b318bbcfffb1a71d6258af",
        "0dcad2b1667642d28c933de70915331e-8537fb11f6ed11e7bf9f3cf862f4613a",
        "0dcad2b1667642d28c933de70915331e-8700ea097f3eb219b0e61d871936bfe4"
      ]
    }
  ],
  "hosts": [
    {
      "host_uuid": "ac169cd1fb5744999eaadc5da595f88b",
      "checks": [
      ]
    },
    {
      "host_uuid": "0dcad2b1667642d28c933de70915331e",
      "checks": [
        "0dcad2b1667642d28c933de70915331e-905374c0f6ed11e78cd83cf862f4613a",
        "0dcad2b1667642d28c933de70915331e-714f6a1b90b318bbcfffb1a71d6258af",
        "0dcad2b1667642d28c933de70915331e-8537fb11f6ed11e7bf9f3cf862f4613a",
        "0dcad2b1667642d28c933de70915331e-8700ea097f3eb219b0e61d871936bfe4"
      ]
    }
  ]
}
```

GET /api/v1/host

Cet appel permet de recevoir la liste des uuids de tous les hôtes ainsi que de leurs checks.

Réponse

Codes de retour

Codes de retour	Explications
200	OK
401	Permission refusée (Mauvais token)

Retour du code 200

- **hosts** :
 - **host_uuid** : text
 - **checks** :
 - **check_uuid** : text

Exemple

<http://localhost:50100/api/v1/host?token=ak5zv6t5s25r6g4>

```
{
  "hosts": [
    {
      "host_uuid": "ac169cd1fb5744999eaadc5da595f88b",
      "checks": [
      ]
    },
    {
      "host_uuid": "0dcad2b1667642d28c933de70915331e",
      "checks": [
        "0dcad2b1667642d28c933de70915331e-905374c0f6ed11e78cd83cf862f4613a",
        "0dcad2b1667642d28c933de70915331e-714f6a1b90b318bbcfffb1a71d6258af",
        "0dcad2b1667642d28c933de70915331e-8537fb11f6ed11e7bf9f3cf862f4613a",
        "0dcad2b1667642d28c933de70915331e-8700ea097f3eb219b0e61d871936bfe4"
      ]
    }
  ]
}
```

GET /api/v1/host/{uuid}

Cet appel permet de recevoir des informations sur un hôte.

Paramètres

Paramètre	Description
uuid	UUID de l'hôte

Réponse

Codes de retour

Codes de retour	Explications
200	OK
401	Permission refusée (Mauvais token)
404	Hôte introuvable

Retour du code 200

- **host_uuid** : *text*
- **host_name** : *text*
- **host_display_name** : *text*
- **status** : *text*
 - OK
 - WARNING
 - CRITICAL
 - UNKNOWN
 - MISSING-DATA
 - SHINKEN-INACTIVE
- **status_since** : *epoch (en secondes)*
- **context** : *text*
 - NOTHING
 - ACKNOWLEDGED
 - PARTIAL-ACKNOWLEDGED
 - INHERITED-ACKNOWLEDGED
 - DOWNTIME
 - PARTIAL-DOWNTIME
 - INHERITED-DOWNTIME
 - FLAPPING
 - PARTIAL-FLAPPING
 - DISABLED
- **business_impact** : 0, 1, 2, 3, 4, 5
- **output** : *text*
- **long_output** : *text*

Exemple

<http://localhost:50100/api/v1/host/408cd29ad37611e8810e0800277b7e16?token=ak5zv6t5s25r6g4>

```
{
  "status": "OK",
  "host_display_name": "localhost",
  "long_output": "",
  "host_name": "localhost",
  "context": "NOTHING",
  "status_since": 1540457802,
  "business_impact": 2,
  "output": "PING OK - Packet loss = 0%, RTA = 0.02 ms",
  "host_uuid": "408cd29ad37611e8810e0800277b7e16"
}
```

GET /api/v1/check-on-host/{uuid}

Cet appel permet de recevoir des informations sur un check attaché à un hôte.

Paramètres

Paramètre	Description
uuid	UUID du check, sachant qu'il correspond à {uuid_de_l'hôte}-{uuid_du_check}

Réponse

Codes de retour

Codes de retour	Explications
200	OK
401	Permission refusée (Mauvais token)
404	Check introuvable

Retour du code 200

- **check_uuid** : *text*
- **check_name** : *text*
- **check_display_name** : *text*
- **status** : *text*
 - OK
 - WARNING
 - CRITICAL
 - UNKNOWN
 - MISSING-DATA
 - SHINKEN-INACTIVE
- **status_since** : *epoch (en secondes)*
- **context** : *text*
 - NOTHING
 - ACKNOWLEDGED
 - PARTIAL-ACKNOWLEDGED
 - INHERITED-ACKNOWLEDGED
 - DOWNTIME
 - PARTIAL-DOWNTIME
 - INHERITED-DOWNTIME
 - FLAPPING
 - PARTIAL-FLAPPING
 - DISABLED
- **output** : *text*
- **business_impact** : 0, 1, 2, 3, 4, 5
- **long_output** : *text*
- **host_uuid** : *text*
- **host_name** : *text*
- **host_display_name** : *text*

Exemple

<http://localhost:50100/api/v1/check-on-host/408cd29ad37611e8810e0800277b7e16-022a505c849f840e110f2aba5e27597f?>

[token=ak5zv6t5s25r6g4](#)

```
{
  "status": "OK",
  "host_display_name": "localhost",
  "check_name": "Broker Daemon Livedata - broker-master",
  "host_uuid": "408cd29ad37611e8810e0800277b7e16",
  "long_output": "",
  "host_name": "localhost",
  "context": "NOTHING",
  "output": "<span style=\"color:#2A9A3D;font-weight:bold;\">[OK]</span> The example works fine.<br/>",
  "business_impact": 2,
  "status_since": 1540457781.797493,
  "check_uuid": "022a505c849f840e110f2aba5e27597f",
  "check_display_name": "Broker Daemon Livedata - broker-master"
}
```

GET /api/v1/cluster

Cet appel permet de recevoir la liste des uuids de tous les clusters ainsi que de leurs checks.

Réponse

Codes de retour

Codes de retour	Explications
200	OK
401	Permission refusée (Mauvais token)

Retour du code 200

clusters :

- cluster_uuid : text
- checks :
 - check_uuid : text

Exemple

<http://localhost:50100/api/v1/cluster?token=ak5zv6t5s25r6g4>

```
{
  "clusters": [
    {
      "cluster_uuid": "ac169cd1fb5744999eaadc5da595f88b",
      "checks": [
      ]
    },
    {
      "cluster_uuid": "0dcad2b1667642d28c933de70915331e",
      "checks": [
        "0dcad2b1667642d28c933de70915331e-905374c0f6ed11e78cd83cf862f4613a",
        "0dcad2b1667642d28c933de70915331e-714f6a1b90b318bbcfffb1a71d6258af",
        "0dcad2b1667642d28c933de70915331e-8537fb11f6ed11e7bf9f3cf862f4613a",
        "0dcad2b1667642d28c933de70915331e-8700ea097f3eb219b0e61d871936bfe4"
      ]
    }
  ]
}
```

GET /api/v1/cluster/{uuid}

Cet appel permet de recevoir des informations sur un cluster.

Paramètres

Paramètre	Description
uuid	UUID du cluster

Réponse

Codes de retour

Codes de retour	Explications
200	OK
401	Permission refusée (Mauvais token)
404	Cluster introuvable

Retour du code 200

- **cluster_uuid** : *text*
- **cluster_name** : *text*
- **cluster_display_name** : *text*
- **status** : *text*
 - OK
 - WARNING
 - CRITICAL
 - UNKNOWN
 - MISSING-DATA
 - SHINKEN-INACTIVE
- **status_since** : *epoch (en secondes)*
- **context** : *text*
 - NOTHING
 - ACKNOWLEDGED
 - PARTIAL-ACKNOWLEDGED
 - INHERITED-ACKNOWLEDGED
 - DOWNTIME
 - PARTIAL-DOWNTIME
 - INHERITED-DOWNTIME
 - FLAPPING
 - PARTIAL-FLAPPING
 - DISABLED
- **business_impact** : 0, 1, 2, 3, 4, 5
- **output** : *text*
- **long_output** : *text*

Exemple

<http://localhost:50100/api/v1/cluster/408gfhfr7611e8810e0800277b7e16?token=ak5zv6t5s25r6g4>

```
{
  "status": "OK",
  "cluster_display_name": "localhost",
  "long_output": "",
  "cluster_name": "localhost",
  "context": "NOTHING",
  "status_since": 1540457802,
  "business_impact": 2,
  "output": "PING OK - Packet loss = 0%, RTA = 0.02 ms",
  "cluster_uuid": "408gfhfr7611e8810e0800277b7e16"
}
```

GET /api/v1/check-on-cluster/UUID

Cet appel permet de recevoir des informations sur un check attaché à un cluster.

Paramètres

Paramètre	Description
uuid	UUID du check, sachant qu'il correspond à {uuid_du_cluster}-{uuid_du_check}

Réponse

Codes de retour

Codes de retour	Explications
200	OK
401	Permission refusée (Mauvais token)
404	Check introuvable

Retour du code 200

- **check_uuid** : *text*
- **check_name** : *text*
- **check_display_name** : *text*
- **status** : *text*
 - OK
 - WARNING
 - CRITICAL
 - UNKNOWN
 - MISSING-DATA
 - SHINKEN-INACTIVE
- **status_since** : *epoch (en secondes)*
- **context** : *text*
 - NOTHING
 - ACKNOWLEDGED
 - PARTIAL-ACKNOWLEDGED
 - INHERITED-ACKNOWLEDGED
 - DOWNTIME
 - PARTIAL-DOWNTIME
 - INHERITED-DOWNTIME
 - FLAPPING
 - PARTIAL-FLAPPING
 - DISABLED
- **output** : *text*
- **business_impact** : 0, 1, 2, 3, 4, 5
- **long_output** : *text*
- **cluster_uuid** : *text*
- **cluster_name** : *text*
- **cluster_display_name** : *text*

Exemple

<http://localhost:50100/api/v1/check-on-cluster/408gfhfr7611e8810e0800277b7e16-044a505c849f840e110f2aba5e27597f?>

[token=ak5zv6t5s25r6g4](#)

```
{
  "status": "OK",
  "cluster_display_name": "localhost",
  "check_name": "Broker Daemon Livedata - broker-master",
  "cluster_uuid": "408gfhfr7611e8810e0800277b7e16",
  "long_output": "",
  "cluster_name": "localhost",
  "context": "NOTHING",
  "output": "<span style='color:#2A9A3D;font-weight:bold;'>[OK]</span> The example works fine.<br/>",
  "business_impact": 2,
  "status_since": 1540457781.797493,
  "check_uuid": "044a505c849f840e110f2aba5e27597f",
  "check_display_name": "Example cluster check"
}
```

Obtenir un UUID

Hôtes et Clusters

Pour récupérer l'UUID d'un hôte ou d'un cluster il suffit de se rendre dans sa page de configuration sur l'UI de Configuration, ou de le sélectionner dans l'UI de Visualisation. Il se trouve alors dans l'URL de la page comme sur les screens suivants:

? Unknown Attachment

? Unknown Attachment

Checks attachés

Pour récupérer l'UUID d'un check attaché à un hôte ou à un cluster, il suffit de se rendre dans sa page de configuration sur l'UI de Configuration en y rajoutant l'UUID de l'hôte ou du cluster sur lequel il est attaché, suivi d'un tiret. Dans l'interface de Visualisation, il peut être trouvé directement dans l'URL en ouvrant le détail du check:

? Unknown Attachment

Macros Shinken

Des macros sont également disponibles sur Shinken pour utiliser l'UUID d'un hôte/cluster/check.

Macro	Description
\$HOSTUUID\$	Renvoie l'UUID de l'hôte ou du cluster
\$SERVICEUUID\$	Renvoie l'UUID du check

Exemple d'utilisation

Ces macros peuvent être utilisées par exemple dans les URL externes. Par exemple:

? Unknown Attachment

L'URL externe de cet hôte amène directement sur le volet Graphiques de l'hôte en question.