

Module broker-module-livodata

Sommaire

- [Introduction](#)
- [Activation du module](#)
 - [Mécanisme des données manquantes](#)
- [Configuration](#)
 - [Configuration de l'interface et du port d'écoute](#)
 - [Activation du SSL](#)
 - [Absorption des broks \(éléments de supervision\)](#)
- [Authentification](#)
- [Utilisation](#)
 - [GET /api/v1/all-monitored-elements](#)
 - [GET /api/v1/host](#)
 - [GET /api/v1/host/{uuid}](#)
 - [GET /api/v1/check-on-host/{uuid}](#)
 - [GET /api/v1/cluster](#)
 - [GET /api/v1/cluster/{uuid}](#)
 - [GET /api/v1/check-on-cluster/UUID](#)
- [Obtenir un UUID](#)
 - [Hôtes et Clusters](#)
 - [Checks attachés](#)
 - [Donnée Shinken](#)

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
```

Mécanisme des données manquantes



Voir : [Détermination de l'état de l'hôte](#) (pour plus d'information sur les cas particuliers)

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-livdata*" est **50100**. Ce port peut être changé via le paramètre "*port*" dans le fichier de configuration du module: ***/etc/shinken/modules/broker-module-livdata.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-livdata.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-livdata.cfg

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

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-livdata.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
```

Absorption des broks (éléments de supervision)

Le fonctionnement du processus de récupération des **broks** de ce module peut être configuré via certains paramètres, afin de modifier son "agressivité".

Pendant la mise à jour des données de supervision, le module ne peut pas répondre aux requêtes via son API.



Une mauvaise configuration de ces paramètres peut compromettre le bon fonctionnement du module. En cas de doute, rapprochez-vous de votre support.

Principe de l'algorithme d'absorption des broks :

1. Attente de broks à traiter
2. Récupération de broks en retard (fonctionnalité de rattrapage)
3. Désérialisation des broks
4. Entrée en session critique (les requêtes à l'API sont bloquées)
5. Traitement des broks
6. Libérer la session critique et attendre de nouveaux broks, **ou** continuer l'absorption de broks (en cas de retard important, on repart à l'étape 1. en restant sur la session critique)

Clé	Type	Valeur par défaut	Description
<code>broker_module_livedata__broks_getter__activate_late_set_catchup</code>	Booléen	1	Utilisation de la fonctionnalité de rattrapage pour absorber des broks en retard : • 1 : Activer • 0 : Désactiver
<code>broker_module_livedata__broks_getter__nb_late_set_allowed_before_catchup</code>	Entier	10	Nombre de brok set en attente tolérés. Au-dessus de ce nombre, les brok set sont immédiatement récupérés par l'algorithme de rattrapage pour être traités maintenant
<code>broker_module_livedata__broks_getter__catchup_broks_managed_by_module_in_a_catchup_loop</code>	Entier	200000	Nombre maximal de broks que l'algorithme de rattrapage récupère avant de lancer le traitement Ce paramètre permet de borner la consommation mémoire et le temps d'exécution d'un tour de boucle de traitement
<code>broker_module_livedata__broks_getter__catchup_run_endless_until_nb_late_set_allowed_reached</code>	Booléen	1	Après traitement des broks , si le nombre de brok set en retard est trop élevé, • 1 : continuer le rattrapage et absorber des broks en retard (en restant sur la session critique ("avec le lock")) • 0 : arrêter l'absorption de brok et libérer la session critique (rendre le lock)
<code>broker_module_livedata__broks_getter__include_deserialisation_and_catchup_in_lock</code>	Booléen	0	Dans le cas où vous voulez disposer d'un maximum de temps CPU pour traiter les broks en retard, vous pouvez inclure la phase 2 (Récupération de broks en retard) et Phase 3 (Désérialisation des broks) dans la phase Critique (Phase 4) La récupération des broks en retard, et la désérialisation se font alors dans la session critique (Locké) pour • 1 : Activer • 0 : Désactiver

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 mises 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 les 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)*
- **status_confirmed_since** : *epoch (en secondes, ou vide en état SOFT)*
- **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*

- **raw_perf_data** : text, version texte de perf_data
- **perf_data** : liste de métriques au format :
 - **value** : number
 - **uom** : Unit Of Mesure / Unité de mesure
 - **warning** : valeur d'alerte
 - **critical** : valeur d'alerte critique
 - **min** : valeur minimale possible
 - **max** : valeur maximale possible

Exemple

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

{
  "status": "OK",
  "host_display_name": "localhost",
  "raw_perf_data": "rta=0.021000ms;1000.000000;3000.000000;0.000000 pl=0%;100;100;0",
  "perf_data": [{"name": "rta", "min": 0, "max": null, "value": 0.021, "warning": 1000, "critical": 3000, "uom": "ms"}, {"name": "pl", "min": 0, "max": 100, "value": 0, "warning": 100, "critical": 100, "uom": "%"}],
  "long_output": "",
  "host_name": "localhost",
  "context": "NOTHING",
  "status_since": 1540457802,
  "status_confirmed_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 les 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)
- **status_confirmed_since** : epoch (en secondes, ou vide en état SOFT)
- **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*
- **raw_perf_data** : *text*, version texte de perf_data
- **perf_data** : liste de métriques au format :
 - **value** : *number*
 - **uom** : Unit Of Measure / Unité de mesure
 - **warning** : valeur d'alerte
 - **critical** : valeur d'alerte critique
 - **min** : valeur minimale possible
 - **max** : valeur maximale possible

Exemple

[http://localhost:50100/api/v1/check-on-host/408cd29ad37611e8810e0800277b7e16-022a505c849f840e110f2aba5e27597f?](http://localhost:50100/api/v1/check-on-host/408cd29ad37611e8810e0800277b7e16-022a505c849f840e110f2aba5e27597f?token=ak5zv6t5s25r6g4)

[token=ak5zv6t5s25r6g4](#)

```
{
  "status": "OK",
  "host_display_name": "localhost",
  "raw_perf_data": "livedata_nb_error_last_hour=0 livedata_average_response_time=0.000852610383715
livedata_nb_request_last_hour=84 livedata_error_percent=0.0",
  "perf_data": [{"name": "livedata_nb_error_last_hour", "min": null, "max": null, "value": 0, "warning":
null, "critical": null, "uom": null}, {"name": "livedata_nb_request_last_hour", "min": null, "max": null,
"value": 84, "warning": null, "critical": null, "uom": null},
  "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,
  "status_confirmed_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)*
- **status_confirmed_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*
- **raw_perf_data** : text, version texte de perf_data
- **perf_data** : liste de métriques au format :
 - **value** : number
 - **uom** : Unit Of Mesure / Unité de mesure
 - **warning** : valeur d'alerte
 - **critical** : valeur d'alerte critique
 - **min** : valeur minimale possible
 - **max** : valeur maximale possible

Exemple

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

```
{
  "status": "OK",
  "cluster_display_name": "localhost",
  "raw_perf_data": "rta=0.021000ms;1000.000000;3000.000000;0.000000 pl=0%;100;100;0",
  "perf_data": [{"name": "rta", "min": 0, "max": null, "value": 0.021, "warning": 1000, "critical": 3000, "uom": "ms"}, {"name": "pl", "min": 0, "max": 100, "value": 0, "warning": 100, "critical": 100, "uom": "%"}],
  "long_output": "",
  "cluster_name": "localhost",
  "context": "NOTHING",
  "status_since": 1540457802,
  "status_confirmed_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)*
- **status_confirmed_since** : *epoch (en secondes, ou vide en état SOFT)*
- **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*
- **raw_perf_data** : *text, version texte de perf_data*
- **perf_data** : *liste de métriques au format :*
 - **value** : number
 - **uom** : Unit Of Mesure / Unité de mesure
 - **warning** : valeur d'alerte
 - **critical** : valeur d'alerte critique
 - **min** : valeur minimale possible
 - **max** : valeur maximale possible

Exemple

<http://localhost:50100/api/v1/check-on-cluster/408gfhr7611e8810e0800277b7e16-044a505c849f840e110f2aba5e27597f?>

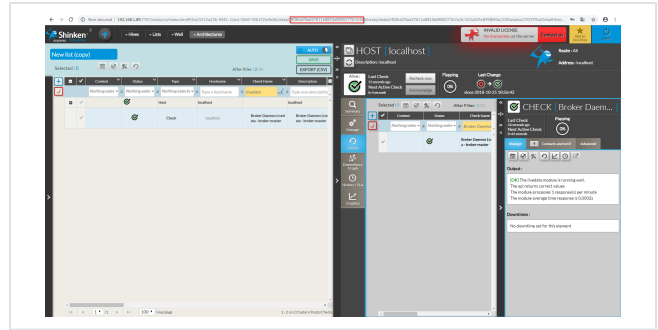
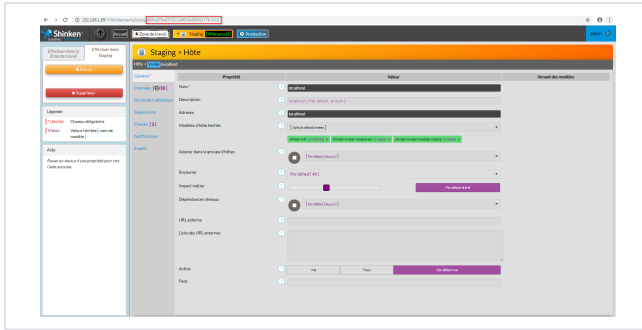
[token=ak5zv6t5s25r6g4](#)

```
{
  "status": "OK",
  "cluster_display_name": "localhost",
  "raw_perf_data": "livedata_nb_error_last_hour=0 livedata_average_response_time=0.000852610383715
livedata_nb_request_last_hour=84 livedata_error_percent=0.0",
  "perf_data": [{"name": "livedata_nb_error_last_hour", "min": null, "max": null, "value": 0, "warning":
null, "critical": null, "uom": null}, {"name": "livedata_nb_request_last_hour", "min": null, "max": null,
"value": 84, "warning": null, "critical": null, "uom": null},
  "check_name": "Broker Daemon Livedata - broker-master",
  "cluster_uuid": "408gfhr7611e8810e0800277b7e16",
  "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,
  "status_confirmed_since": 1540457781.797493,
  "check_uuid": "044a505c849f840e110f2aba5e27597f",
  "check_display_name": "Example cluster check"
}
```

Obtenir un UUID

Hôtes et Clusters

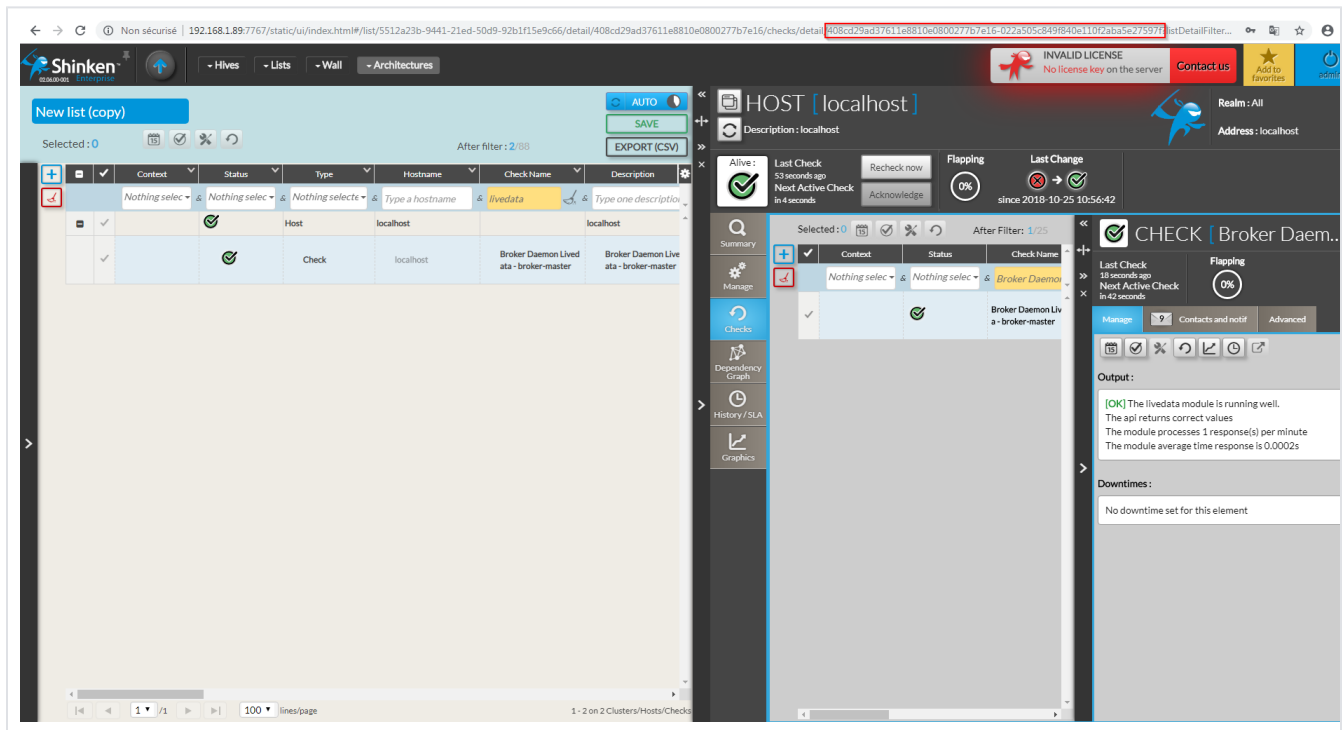
Pour récupérer l'UUID d'un hôte ou d'un cluster, il suffit de se rendre dans sa page de configuration sur l'UI de Configuration, ou de le sélectionner dans l'UI de Visualisation. Il se trouve alors dans l'URL de la page comme sur les images suivantes:



Il est aussi possible de récupérer les uuid directement par le biais des requêtes apis : all-monitored-elements, host et cluster.

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:



Donnée Shinken

Des données 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 données peuvent être utilisées par exemple dans les URL externes. Par exemple:

