

# Collecteur de type ( synchronizer-collector-linker ) - Import depuis un autre Synchronizer

## Sommaire

- Introduction
  - Fonctionnement
- Configuration
- Utilisation de la source
  - Etape 1: Définir une connexion aux Shinken distants
  - Etape 2: Lancer l'import
    - Liste des Synchronizers distants
- Précisions techniques
  - Clés de synchronisation
    - Propriétés par défaut utilisé pour la construction des clés de synchronisation

## Objectifs

Méthode POST de type READ qui permet de récupérer des données de supervision, comme sur l'interface de visualisation,

- Filtrées ( optionnel )
- Triées
- Rangées :
  - En arbres ( hôtes/clusters => checks )
  - Tous au même niveau

Contrairement à la route [V2 - \( READ \) /api/v2/inventory](#), cette route permet de choisir les propriétés obtenues sur le résultat en sortie.

Toutefois, attention au volume de données que l'ajout de propriétés supplémentaires à récupérer peut engendrer.

## Les paramètres

Cet appel utilise les 4 paramètres suivants :

- **filterX**
- **sort**
- **output\_format**
- **ouput\_field**

Voir la page [V2 - Les paramètres des API du broker-module-livedata](#), pour leur description complète.

## Réponse

### Codes de retour

Codes de retour	Explications
200	OK
400	Paramètre invalide
401	Accès nécessite une authentification ou un Token valide.
403	Authentification de l'utilisateur OK, mais droits non suffisant.
500	L'appel est valide, mais un problème d'exécution est rencontré.

### Retour du code 200

Les propriétés retournées doivent être choisies avec l'option **output\_field**

mais les propriétés suivantes seront au minimum automatiquement retournées :

- **nb\_element**
- **type** (si le paramètre **output\_format** vaut *elements\_on\_same\_level*)
- **father\_uuid**
- **father\_name**
- **par check** :
  - **check\_uuid**

- **check\_name**

Voir la page [V2 - Les propriétés présentes dans le retour 200 des API du broker-module-livedata](#) , pour la description complète de tous les propriétés pouvant être renvoyées.

Suivant le paramètre **output\_format** ( *checks\_attached\_to\_father* / *elements\_on\_same\_level* )

- ( présentation du format de retour ci-dessus, mais se référer à l'exemple pour le format exact )

**checks\_attached\_to\_father:**

- **request\_statistics** :
  - ...
- **elements\_found** :
  - **clusters** :
    - **cluster1**:
      - **cluster\_uuid** : *text*
      - **cluster\_name** : *text*
      - ...
      - **checks** :
        - **check\_uuid 1** : *text*, **check\_name1** : *text*, ...
        - **check\_uuid2** : *text*, **check\_name2** : *text*, ...
        - ...
    - ...
  - **hosts** :
    - **host 1**:
      - **father\_uuid** : *text*
      - **father\_name** : *text*
      - ...
      - **checks** :
        - **check\_uuid1** : *text*, **check\_name1** : *text*, ...
        - **check\_uuid2** : *text*, **check\_name2** : *text*, ...
        - ...
    - ...

**elements\_on\_same\_level :**

- **request\_statistics** :
  - ...
- **elements\_found**:
  - **elem1**:
    - **type**: *cluster*
    - **father\_name**: *text*
    - **father\_uuid** : *text*
    - ...
  - **elem2**:
    - **type**: *host*
    - **father\_name**: *text*
    - **father\_uuid** : *text*
    - ...
  - **elem3**:
    - **type**: *check*
    - **check\_uuid** : *text*
    - **check\_name**: *text*
    - **father\_name** : *text*
    - **father\_uuid** : *text*
    - ...
  - **elem4**:
    - **type**: *check*
    - **check\_uuid** : *text*
    - **check\_name**: *text*
    - **father\_name** : *text*
    - **father\_uuid** : *text*
    - ...
  - ...

```
curl -s -S -H 'x-api-token: XYZ' \
-d "output_format=checks_attached_to_father" \
-d
"output_field=raw_perf_data~father_templates~status
~context" \
-d "filter0=type:cluster^^host^^check_cluster" \
-d "filter1=type:check_host~father_name:
Bordeaux~check_name:CPU stats" \
-d "filter2=type:check_host~father_name:
bordeaux~check_name:Disks Stats" \
http://broker-module-livedata:50100/api/v2/all-
monitored-elements
```

```
curl -s -S -H 'x-api-token: XYZ' \
-d "output_format=elements_on_same_level" \
-d
"output_field=raw_perf_data~father_templates~status
~context" \
-d "filter0=type:cluster^^host^^check_cluster" \
-d "filter1=type:check_host~father_name:
Bordeaux~check_name:CPU stats" \
-d "filter2=type:check_host~father_name:
bordeaux~check_name:Disks Stats" \
http://broker-module-livedata:50100/api/v2/all-
monitored-elements
```

Exemple de sortie attendue :

<http://localhost:50100/api/v2/all-monitored-elements>

```
{
  "request_statistics": {
    "nb_elements_total": 9,
    "nb_hosts_total": 4,
    "nb_clusters_total": 1,
    "nb_checks_total": 4,
    "nb_elements_filtered": 8,
    "nb_hosts_filtered": 4,
```

Exemple de sortie attendue :

<http://localhost:50100/api/v2/all-monitored-elements>

```
{
  "request_statistics": {
    "nb_elements_total": 9,
    "nb_hosts_total": 4,
    "nb_clusters_total": 1,
    "nb_checks_total": 4,
    "nb_elements_filtered": 8,
    "nb_hosts_filtered": 4,
```

```

    "nb_clusters_filtered": 1,
    "nb_checks_filtered": 3
  },
  "elements_found": {
    "clusters": [
      {
        "status": 0,
        "father_templates": [],
        "father_name": "datacenter bdx",
        "context": "NOTHING",
        "raw_perf_data": "",
        "father_uuid":
"d6921ee8ba1511eba36c0800277faebe",
        "checks": [
          {
            "status": 0,
            "check_name": "System Uptime",
            "check_uuid":
"d6921ee8ba1511eba36c0800277faebe-
e6daad4cbal511eb95980800277faebe",
            "raw_perf_data": "",
            "context": "NOTHING"
          }
        ]
      }
    ],
    "hosts": [
      {
        "status": 0,
        "father_templates": [
          "linux"
        ],
        "father_name": "Bordeaux",
        "context": "NOTHING",
        "raw_perf_data": "rta=0.035000ms;
1000.000000;3000.000000;0.000000 pl=0%;100;100;0",
        "father_uuid":
"2c6dcflabal611ebaa7d0800277faebe",
        "checks": [
          {
            "status": 0,
            "check_name": "CPU Stats",
            "check_uuid":
"2c6dcflabal611ebaa7d0800277faebe-
c296d75e5ad911e58cc5080027f08538",
            "raw_perf_data": "cpu_l_sys=5.10%
cpu_l_usr=0.00% cpu_l_steal=0.00% cpu_l_soft=1.02%
cpu_l_irq=0.00% cpu_l_nice=7.14% cpu_l_idle=86.73%
cpu_l_iowait=0.00% cpu_l_guest=0.00% cpu_0_sys=4.
21% cpu_0_usr=0.00% cpu_0_steal=0.00% cpu_0_soft=0.
00% cpu_0_irq=0.00% cpu_0_nice=5.26% cpu_0_idle=90.
53% cpu_0_iowait=0.00% cpu_0_guest=0.00%
cpu_all_sys=4.69% cpu_all_usr=0.00%
cpu_all_steal=0.00% cpu_all_soft=0.52%
cpu_all_irq=0.00% cpu_all_nice=5.73%
cpu_all_idle=89.06% cpu_all_iowait=0.00%
cpu_all_guest=0.00%",
            "context": "NOTHING"
          }
        ],
        {
          "status": 0,
          "check_name": "Disks Stats",
          "check_uuid":
"2c6dcflabal611ebaa7d0800277faebe-
c29735965ad911e58cc5080027f08538",
          "raw_perf_data": "sda_r_by_sec=0r/s
sda_w_by_sec=13w/s sda_rKB_by_sec=0rKB/s
sda_wKB_by_sec=45wKB/s sda_util=0.28% dm-
0_r_by_sec=0r/s dm-0_w_by_sec=12w/s dm-
0_rKB_by_sec=0rKB/s dm-0_wKB_by_sec=45wKB/s dm-
0_util=0.28% dm-1_r_by_sec=0r/s dm-1_w_by_sec=0w/s
"
        }
      }
    ]
  }
}

```

```

    "nb_clusters_filtered": 1,
    "nb_checks_filtered": 3
  },
  "elements_found": [
    {
      "status": 0,
      "father_templates": [
        "linux"
      ],
      "father_name": "Bordeaux",
      "raw_perf_data": "rta=0.011000ms;1000.000000;
3000.000000;0.000000 pl=0%;100;100;0",
      "context": "NOTHING",
      "father_uuid":
"2c6dcflabal611ebaa7d0800277faebe",
      "type": "host"
    },
    {
      "status": 0,
      "father_name": "Bordeaux",
      "raw_perf_data": "cpu_l_sys=3.16%
cpu_l_usr=1.05% cpu_l_steal=0.00% cpu_l_soft=0.00%
cpu_l_irq=0.00% cpu_l_nice=5.26% cpu_l_idle=90.53%
cpu_l_iowait=0.00% cpu_l_guest=0.00% cpu_0_sys=4.
30% cpu_0_usr=0.00% cpu_0_steal=0.00% cpu_0_soft=1.
08% cpu_0_irq=0.00% cpu_0_nice=6.45% cpu_0_idle=88.
17% cpu_0_iowait=0.00% cpu_0_guest=0.00%
cpu_all_sys=4.26% cpu_all_usr=0.00%
cpu_all_steal=0.00% cpu_all_soft=0.53%
cpu_all_irq=0.00% cpu_all_nice=6.38%
cpu_all_idle=88.83% cpu_all_iowait=0.00%
cpu_all_guest=0.00%",
      "context": "NOTHING",
      "check_name": "CPU Stats",
      "type": "check_host",
      "father_uuid":
"2c6dcflabal611ebaa7d0800277faebe",
      "check_uuid":
"2c6dcflabal611ebaa7d0800277faebe-
c296d75e5ad911e58cc5080027f08538"
    },
    {
      "status": 0,
      "father_name": "Bordeaux",
      "raw_perf_data": "sda_r_by_sec=0r/s
sda_w_by_sec=13w/s sda_rKB_by_sec=0rKB/s
sda_wKB_by_sec=44wKB/s sda_util=0.26% dm-
0_r_by_sec=0r/s dm-0_w_by_sec=12w/s dm-
0_rKB_by_sec=0rKB/s dm-0_wKB_by_sec=44wKB/s dm-
0_util=0.26% dm-1_r_by_sec=0r/s dm-1_w_by_sec=0w/s
dm-1_rKB_by_sec=0rKB/s dm-1_wKB_by_sec=0wKB/s dm-
1_util=0.00%",
      "context": "NOTHING",
      "check_name": "Disks Stats",
      "type": "check_host",
      "father_uuid":
"2c6dcflabal611ebaa7d0800277faebe",
      "check_uuid":
"2c6dcflabal611ebaa7d0800277faebe-
c29735965ad911e58cc5080027f08538"
    },
    {
      "status": 0,
      "father_templates": [],
      "father_name": "datacenter bdx",
      "raw_perf_data": "",
      "context": "NOTHING",
      "father_uuid":
"d6921ee8ba1511eba36c0800277faebe",
      "type": "cluster"
    }
  ],

```

```

dm-1_rKB_by_sec=0rKB/s dm-1_wKB_by_sec=0wKB/s dm-
l_util=0.00%",
  "context": "NOTHING"
}
]
},
{
  "status": 0,
  "father_templates": [],
  "father_name": "Lyon",
  "context": "NOTHING",
  "raw_perf_data": "rta=0.013000ms;
1000.000000;3000.000000;0.000000 pl=0%;100;100;0",
  "father_uuid":
"8db8b5f0ba1611eba60d0800277faebe",
  "checks": []
},
{
  "status": 2,
  "father_templates": [
    "shinken",
    "shinken-daemon",
    "shinken-synchronizer-db",
    "shinken-broker-db",
    "linux"
  ],
  "father_name": "Nantes",
  "context": "NOTHING",
  "raw_perf_data": " -6] ",
  "father_uuid":
"76f45d80bale11eba2670800277faebe",
  "checks": []
},
{
  "status": 0,
  "father_templates": [],
  "father_name": "Rennes",
  "context": "NOTHING",
  "raw_perf_data": "rta=0.017000ms;
1000.000000;3000.000000;0.000000 pl=0%;100;100;0",
  "father_uuid":
"e5460dc2ba1611eb81580800277faebe",
  "checks": []
}
]
}
}

```

```

{
  "status": 0,
  "father_name": "datacenter bdx",
  "raw_perf_data": "",
  "context": "NOTHING",
  "check_name": "System Uptime",
  "type": "check_cluster",
  "father_uuid":
"d6921ee8ba1511eba36c0800277faebe",
  "check_uuid":
"d6921ee8ba1511eba36c0800277faebe-
e6daad4c1511eb95980800277faebe"
},
{
  "status": 0,
  "father_templates": [],
  "father_name": "Lyon",
  "raw_perf_data": "rta=0.013000ms;1000.000000;
3000.000000;0.000000 pl=0%;100;100;0",
  "context": "NOTHING",
  "father_uuid":
"8db8b5f0ba1611eba60d0800277faebe",
  "type": "host"
},
{
  "status": 2,
  "father_templates": [
    "shinken",
    "shinken-daemon",
    "shinken-synchronizer-db",
    "shinken-broker-db",
    "linux"
  ],
  "father_name": "Nantes",
  "raw_perf_data": " -6] ",
  "context": "NOTHING",
  "father_uuid":
"76f45d80bale11eba2670800277faebe",
  "type": "host"
},
{
  "status": 0,
  "father_templates": [],
  "father_name": "Rennes",
  "raw_perf_data": "rta=0.025000ms;1000.000000;
3000.000000;0.000000 pl=0%;100;100;0",
  "context": "NOTHING",
  "father_uuid":
"e5460dc2ba1611eb81580800277faebe",
  "type": "host"
}
]
}

```

## Retour du code 400

### Paramètres POST incorrects

#### Paramètre inconnu

```

$ curl -s -S -H "x-api-token: XYZ" \
-d "parametre_inconnu=is_status:true" \
http://broker-module-livedata:50100/api/v2/all-
monitored-elements
ERROR 400: POST parameter [ parametre_inconnu ] is
unknown

```

```

ERROR 400: POST parameter [ parametre_inconnu ] is unknown

```

## Messages d'erreurs des filtres ( filterX )

( voir la page [V2 - Les paramètres des API du broker-module-livedata](#) )

### Filtre inexistant

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "filter01=is_status_:true" \  
http://broker-module-livedata:50100/api/v2/all-  
monitored-elements  
ERROR 400: filtering[0]: invalid field name [  
is_status_ ]
```

```
ERROR 400: filtering[0]: invalid field name [ is_status_ ]
```

### Filtre incomplet

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "filter0=next_check" \  
http://broker-module-livedata:50100/api/v2/all-  
monitored-elements  
ERROR 400: filtering[0]: missing value for field [  
next_check ]
```

```
ERROR 400: filtering[0]: missing value for field [ next_check ]
```

### Filtre incorrect

Valeur incorrecte pour ce type de filtre

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "filter0=status:9" \  
http://broker-module-livedata:50100/api/v2/all-  
monitored-elements  
ERROR 400: filtering[0]: field [ status ] => wrong  
value ['9']
```

```
ERROR 400: filtering[0]: field [ status ] => wrong value ['9']
```

### Opérateur incorrect

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "filter0=next_check:avant" \  
http://broker-module-livedata:50100/api/v2/all-  
monitored-elements  
ERROR 400: filtering[0]: field [ next_check ]  
unknown date constraint [ avant ]
```

```
ERROR 400: filtering[0]: field [ next_check ] unknown date constraint  
[ avant ]
```

### Argument incorrect

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "filter0=next_check:in-less-than|hier" \  
http://broker-module-livedata:50100/api/v2/all-  
monitored-elements  
ERROR 400: filtering[0]: field [ next_check ] => [  
in-less-than ] => invalid literal for int() with  
base 10: 'hier'
```

```
ERROR 400: filtering[0]: field [ next_check ] => [ in-less-than ] =>  
invalid literal for int() with base 10: 'hier'
```

## Messages d'erreurs liés aux paramètres de tri ( sort )

( voir la page [V2 - Les paramètres des API du broker-module-livedata](#) )

### Propriété inconnu

```
ERROR 400: sort: invalid field name [ host ]
```

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "sort=host" \  
http://broker-module-livodata:50100/api/v2/all-  
monitored-elements  
ERROR 400: sort: invalid field name [ host ]
```

#### Ordre de tri incorrect

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "sort=father_name:big" \  
http://broker-module-livodata:50100/api/v2/all-  
monitored-elements  
ERROR 400: sort: invalid sort direction [ big ]  
for field [ father_name ]
```

```
ERROR 400: sort: invalid sort direction [ big ] for field [ father_name ]
```

#### Messages d'erreurs liés au paramètre de format du résultat ( output\_format )

( voir la page [V2 - Les paramètres des API du broker-module-livodata](#) )

##### Valeur incorrecte

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "output_format=vrai" \  
http://broker-module-livodata:50100/api/v2/all-  
monitored-elements  
ERROR 400: output_format: invalid value [ vrai ]
```

```
ERROR 400: output_format: invalid value [ vrai ]
```

#### Messages d'erreurs lors du paramétrage des propriétés présentes dans la sortie ( output\_field )

( voir la page [V2 - Les paramètres des API du broker-module-livodata](#) )

##### Propriété de sortie inexistante

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "output_field=is_status_" \  
http://broker-module-livodata:50100/api/v2/all-  
monitored-elements  
ERROR 400: output_field: invalid field name [  
is_status_ ]
```

```
ERROR 400: output_field: invalid field name [ is_status_ ]
```

#### Error rendering macro 'excerpt-include'

No link could be created for 'V2 - Les erreurs communes lors de l'envoi de la requête'.