

Network Interfaces SNMP (linux)

Sommaire

- Contexte
- Exemple
 - Exemple de résultat
- Données et métriques
 - Données
 - Métriques

Objectifs

Méthode POST de type READ qui permet de récupérer des données d'hôtes et éventuellement de ses checks, comme sur l'interface de visualisation :

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

Paramètres

Cet appel utilise les 4 paramètres suivants :

- **filterX**
 - Pour la propriété **type**, le fait d'utiliser des valeurs différentes de **host** et **check_host** est interdit sur cette route.
- **sort**
- **output_format**
- **output_field**

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

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 :
 - 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: host
 - father_name : text
 - father_uuid : text
 - ...
 - elem2:
 - type: check
 - check_uuid : text
 - check_name : text
 - father_name : text
 - father_uuid : text
 - ...
 - elem3:
 - 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" \  
http://broker-module-livedata:50100/api/v2/hosts
```

```
curl -s -S -H 'x-api-token: XYZ' \  
-d "output_format=elements_on_same_level" \  
http://broker-module-livedata:50100/api/v2/hosts
```

http://localhost:50100/api/v2/hosts/

```
{
  "request_statistics": {
    "nb_elements_total": 9,
    "nb_hosts_total": 4,
    "nb_checks_total": 4,
    "nb_elements_filtered": 7,
    "nb_hosts_filtered": 4,
    "nb_checks_filtered": 3
  },
  "elements_found": {
    "clusters": [],
    "hosts": [
      {
        "father_uuid":
"2c6dcflaba1611ebaa7d0800277faebe",
        "father_name": "Bordeaux",
        "checks": [
          {
            "check_name": "CPU Stats",
            "check_uuid":
"2c6dcflaba1611ebaa7d0800277faebe-
c296d75e5ad911e58cc5080027f08538"
          },
          {
            "check_name": "Disks Stats",
            "check_uuid":
"2c6dcflaba1611ebaa7d0800277faebe-
c29735965ad911e58cc5080027f08538"
          }
        ]
      },
      {
        "father_uuid":
"8db8b5f0ba1611eba60d0800277faebe",
        "father_name": "Lyon",
        "checks": []
      },
      {
        "father_uuid":
"76f45d80bale11eba2670800277faebe",
        "father_name": "Nantes",
        "checks": [
          {
            "check_name": "CPU Stats",
            "check_uuid":
"76f45d80bale11eba2670800277faebe-
c296d75e5ad911e58cc5080027f08538"
          }
        ]
      },
      {
        "father_uuid":
"e5460dc2ba1611eb81580800277faebe",
        "father_name": "Rennes",
        "checks": []
      }
    ]
  }
}
```

http://localhost:50100/api/v2/hosts/

```
{
  "request_statistics": {
    "nb_elements_total": 9,
    "nb_hosts_total": 4,
    "nb_checks_total": 4,
    "nb_elements_filtered": 7,
    "nb_hosts_filtered": 4,
    "nb_checks_filtered": 3
  },
  "elements_found": [
    {
      "father_uuid":
"2c6dcflaba1611ebaa7d0800277faebe",
      "father_name": "Bordeaux",
      "type": "host"
    },
    {
      "check_name": "CPU Stats",
      "check_uuid":
"2c6dcflaba1611ebaa7d0800277faebe-
c296d75e5ad911e58cc5080027f08538",
      "father_uuid":
"2c6dcflaba1611ebaa7d0800277faebe",
      "father_name": "Bordeaux",
      "type": "check_host"
    },
    {
      "check_name": "Disks Stats",
      "check_uuid":
"2c6dcflaba1611ebaa7d0800277faebe-
c29735965ad911e58cc5080027f08538",
      "father_uuid":
"2c6dcflaba1611ebaa7d0800277faebe",
      "father_name": "Bordeaux",
      "type": "check_host"
    },
    {
      "father_uuid":
"8db8b5f0ba1611eba60d0800277faebe",
      "father_name": "Lyon",
      "type": "host"
    },
    {
      "father_uuid":
"76f45d80bale11eba2670800277faebe",
      "father_name": "Nantes",
      "type": "host"
    },
    {
      "check_name": "CPU Stats",
      "check_uuid":
"76f45d80bale11eba2670800277faebe-
c296d75e5ad911e58cc5080027f08538",
      "father_uuid":
"76f45d80bale11eba2670800277faebe",
      "father_name": "Nantes",
      "type": "check_host"
    },
    {
      "father_uuid":
"e5460dc2ba1611eb81580800277faebe",
      "father_name": "Rennes",
      "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-livodata:50100/api/v2/hosts
```

```
ERROR 400: POST parameter [ parametre_inconnu ] is unknown
```

Messages d'erreurs des filtres ([filterX](#))

Filtre inexistant

```
curl -s -S -H "x-api-token: XYZ" \  
-d "filter01=is_status_:true" \  
http://broker-module-livodata:50100/api/v2/hosts
```

```
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-livodata:50100/api/v2/hosts
```

```
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-livodata:50100/api/v2/hosts
```

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

Opérateur incorrect

```
curl -s -S -H "x-api-token: XYZ" \  
-d "filter0=next_check:avant" \  
http://broker-module-livodata:50100/api/v2/hosts
```

```
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-livodata:50100/api/v2/hosts
```

```
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](#))

Propriété inconnu

```
curl -s -S -H "x-api-token: XYZ" \  
-d "sort=host" \  
http://broker-module-livodata:50100/api/v2/hosts
```

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

Ordre de tri incorrect

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

```
curl -s -S -H "x-api-token: XYZ" \  
-d "sort=father_name:big" \  
http://broker-module-livodata:50100/api/v2/hosts
```

Messages d'erreurs liés au paramètre de format du résultat ([output_format](#))

Valeur incorrecte

```
curl -s -S -H "x-api-token: XYZ" \  
-d "output_format=vrai" \  
http://broker-module-livodata:50100/api/v2/hosts
```

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

Messages d'erreurs lors du paramétrage des propriétés présentes dans la sortie ([output_field](#))

Propriété de sortie inexistante

```
curl -s -S -H "x-api-token: XYZ" \  
-d "output_field=is_status_" \  
http://broker-module-livodata:50100/api/v2/hosts
```

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