

# V2 - Les paramètres des API du broker-module-livedata

## Sommaire

### Les Paramètres

- filterX ( Filtres )
- sort ( tri des résultats )
- output\_format ( structure du retour )
- output\_field ( Informations présentes dans le retour de la requête )
  - Les propriétés utilisables pour les filtres
  - Le reste des propriétés apparaissant dans le retour
  - Exemple d'utilisation
- Exemple complet d'appel

## Les Paramètres

Globalement, il y a 4 types de paramètres :

Nom et format	Valeur par défaut	Description et Syntaxe
<b>filter0=expression~expression~...</b> ... <b>filter9=...</b>	Aucun filtre défini => tous les éléments sont retournés	<ul style="list-style-type: none"><li>~ ayant le sens de "et"</li><li><b>expression</b> de la forme : <b>critère:valeur0^^valeur1</b><ul style="list-style-type: none"><li>où ^^ a le sens de "ou"</li></ul></li></ul> <p>Plusieurs filtres peuvent être utilisés ( <b>filter0</b> , <b>filter1</b> , ...), chaque élément correspondant à <b>au moins un des filtres</b> sera retourné en résultat de la requête.</p> <p>S'il y a plusieurs filtres, les éléments correspondant au filtre0 seront retournés ainsi que les éléments correspondant au filtre1, ainsi que...</p>
<b>sort= propriete1:</b> <b>order1~propriete2:</b> <b>order2~...</b>	<b>father_name:asc</b>	<ul style="list-style-type: none"><li>~ ayant le sens de "et"</li><li><b>"propriete"</b> étant une des propriétés disponibles en sortie de la requête</li><li><b>order</b> valant <b>asc</b> ou <b>desc</b></li></ul>
<b>output_format=valeur</b>	<b>elements_on_same_level</b> (ou non utilisé pour certaines requêtes)	<ul style="list-style-type: none"><li><b>valeur</b> valant <b>checks_attached_to_father</b> ou <b>elements_on_same_level</b></li></ul>
<b>ouput_field=propriete1~propriete2~...</b>	spécifique à chaque requête	<ul style="list-style-type: none"><li>~ ayant le sens de "et"</li><li><b>"propriete"</b> étant une des propriétés disponibles en sortie de la requête</li></ul>

### ! IMPORTANT

Par défaut, les appels utilisent tous ces paramètres, mais certaines routes ont des exceptions

## filterX ( Filtres )

Les filtres ont pour formes :

- filterX=expression~expression**
  - ~ ayant le sens de "et"
  - expression** de la forme : **critère:valeur0^^valeur1**
    - où ^^ a le sens de "ou"
- X** vaut de **0 à 9**.
- Chaque élément correspondant à **au moins un des filtres** sera retourné.

Liste des **critères** et leurs **valeurs possibles**:

Nom	Matching	Type	Valeur	Exemple d'expression
<b>status</b>	correspond exactement	<i>Tableau</i>	<b>0</b> pour <b>OK</b> , <b>1</b> pour <b>ATTENTION</b> , <b>2</b> pour <b>CRITIQUE</b> , <b>3</b> pour <b>INCONNU</b>	<b>0^^1^^2</b>

<b>context</b>	correspond exactement	Tableau	<b>NOTHING, ACKNOWLEDGED, PARTIAL-ACKNOWLEDGED, INHERITED-ACKNOWLEDGED, DOWNTIME, PARTIAL-DOWNTIME, INHERITED-DOWNTIME, FLAPPING, PARTIAL-FLAPPING, DISABLED</b>	<b>NOTHING^^FLAPPING</b>
<b>is_status_confirmed</b>	correspond exactement ( insensible à la casse )	Booléen	<b>True / False</b>	<b>True</b>
<b>type</b>	correspond exactement	Tableau	<b>HOST, CLUSTER, CHECK, CHECK_HOST, CHECK_CLUSTER</b>	<b>CHECK^^HOST</b>
<b>father_name</b>	correspond exactement ( insensible à la casse )	Tableau	Permet de lister les noms exacts des hôtes ou des clusters.  Si vous désirez filtrer des hôtes avec une chaîne et des clusters avec un autre, il vous faut faire 2 filtres avec type=HOST et type=CLUSTER.	<b>bordeaux^^rennes</b>
<b>father_name_contains</b>	contient ( insensible à la casse )	Tableau	Permet de lister des chaînes que l'on cherche dans les hôtes ou des clusters.  Si vous désirez filtrer des hôtes avec une chaîne et des clusters avec un autre, il vous faut faire 2 filtres avec type=HOST et type=CLUSTER.	<b>bor^^renn</b>
<b>father_uuid</b>	correspond exactement	Tableau	UUID des hôtes / clusters	<b>4a893fbcb0047b8a1922bce91e3dfd9</b>
<b>check_name</b>	correspond exactement ( insensible à la casse )	Tableau	Permet de lister les noms de checks exacts recherchés.	<b>CPU Usage^^Disk Usage</b>
<b>check_name_contains</b>	contient ( insensible à la casse )	Tableau	Permet de lister les chaînes que l'on cherche dans les noms de check.	<b>CPU^^Disk</b>
<b>check_uuid</b>	correspond exactement	Tableau	UUID des checks.	<b>4a893fbcb0047b8a1922bce91e3decf</b>
<b>description</b>	contient ( insensible à la casse )	Texte		<b>Bordeaux</b>
<b>address</b>	contient ( insensible à la casse )	Texte		<b>192.168.1.20</b>
<b>realm</b>	correspond exactement	Tableau	nom complet des royaumes	<b>Paris^^Bordeaux^^CORSE</b>
<b>notification_contacts</b>	correspond exactement	Tableau	nom complet de modèle de l'utilisateur	<b>user1^^user2</b>
<b>notification_contact_groups</b>	correspond exactement	Tableau	nom complet de groupe d'utilisateur	
<b>father_templates</b>	correspond exactement	Tableau	nom complet de modèle d'hôte ou de cluster	<b>linux^^http</b>
<b>host_groups</b>	correspond exactement	Tableau	nom complet de groupe d'hôte	<b>ERP_bordeaux</b>
<b>business_impact</b>	correspond exactement	Entier	0, 1, 2, 3, 4, 5	<b>4</b>
<b>status_since</b>	correspond à la contrainte de date	Contrainte de date	<b>LATER-THAN nb secondes</b> <b>OLDER-THAN nb secondes</b>	<b>LATER-THAN 60</b>  Plus récent que 60 secondes
<b>status_confirmed_since</b>	correspond à la contrainte de date	Contrainte de date	<b>LATER-THAN nb secondes</b> <b>OLDER-THAN nb secondes</b>	<b>LATER-THAN 60</b>  Plus récent que 60 secondes
<b>last_check</b>	correspond à la contrainte de date	Contrainte de date	<b>LATER-THAN nb secondes</b> <b>OLDER-THAN nb secondes</b>	<b>OLDER-THAN 60</b>  Plus ancien que 60 secondes
<b>next_check</b>	correspond à la contrainte de date	Contrainte de date	<b>IN-MORE-THAN nb secondes</b> <b>IN-LESS-THAN nb secondes</b>	<b>IN-LESS-THAN 300</b>  Dans moins de 5 minutes

Exemple filtrant tous les éléments qui sont de statut CRITIQUE, et dont le contexte est soit pris en compte ( **ACKNOWLEDGED** ) ou en maintenance ( **DOWNTIME** )

#### Exemple de définition du paramètre filter

```
"filter0=context:DOWNTIME^^ACKNOWLEDGED~status:2~type:check^^cluster^^host~father_name:datacenter bordeaux"
```

## sort ( tri des résultats )

Le tri peut se faire:

- sur n'importe quel critère de filtre,
- par ordre ascendant ( **asc** ) ou descendant ( **desc** ).

Exemple triant la réponse par nom de "père" ( hôte ou cluster ) en ordre alphabétique descendant et par statut ascendant, soit dans le sens 0 => 3

#### Exemple de définition du paramètre sort

```
"sort=father_name:desc~status:asc"
```

## output\_format ( structure du retour )

Ce paramètre permet d'indiquer où sont situés les checks dans la réponse de retour:

- **checks\_attached\_to\_father** : les checks sont accrochés à leurs hôtes / clusters ( *forme d'arbre* )
- **elements\_on\_same\_level** : les checks sont listés au niveau des hôtes / clusters ( *une liste* )

REMARQUE: Dans le cas où le filtre vaut uniquement **type=check** ( donc pas d'hôtes ou clusters )

- Mais où **output\_format=checks\_attached\_to\_father**, les hôtes / clusters seront quand même présents pour les checks correspondant à ce filtre.
- Si le **output\_format=elements\_on\_same\_level**, les hôtes et/ou clusters ne sont pas présents.

## output\_field ( Informations présentes dans le retour de la requête )

Ce paramètre permet de lister les propriétés qui seront affichées sur le résultat en sortie.

Les propriétés présentes peuvent être les suivants :

### Les propriétés utilisables pour les filtres

Nom	Sortie	Valeurs possibles	Exemple d'expression
<b>status</b>	1 valeur	<b>0</b> pour <b>OK</b> , <b>1</b> pour <b>ATTENTION</b> , <b>2</b> pour <b>CRITIQUE</b> , <b>3</b> pour <b>INCONNU</b>	0
<b>context</b>	1 valeur	<b>NOTHING, ACKNOWLEDGED, PARTIAL-ACKNOWLEDGED, INHERITED-ACKNOWLEDGED, DOWNTIME, PARTIAL-DOWNTIME, INHERITED-DOWNTIME, FLAPPING, PARTIAL-FLAPPING, DISABLED</b>	"FLAPPING"
<b>is_status_confirmed</b>	1 valeur	<b>true / false</b>	true
<b>type</b>	1 valeur	<b>HOST, CLUSTER, CHECK_HOST, CHECK_CLUSTER</b>	"check_host"
<b>father_name</b>	1 valeur		"bordeaux-storage"
<b>check_name</b>	1 valeur		"CPU Stats"
<b>description</b>	1 valeur		"Bordeaux"
<b>address</b>	1 valeur		"192.168.1.20"
<b>realm</b>	1 valeur		"Paris"

<b>notification_contacts</b>	valeurs séparées par des virgules	<i>Noms des contacts</i>	[ "user1", "user2" ]
<b>notification_contact_groups</b>	valeurs séparées par des virgules	<i>Noms des groupes de contacts</i>	[ "groupe1", "groupe2" ]
<b>father_templates</b>	valeurs séparées par des virgules	<i>Noms des modèles d'hôtes ou de clusters</i>	[ "linux", "http" ]
<b>host_groups</b>	valeurs séparées par des virgules	<i>Noms des groupes d'hôtes</i>	[ "ERP_bordeaux", "Aquitaine" ]
<b>business_impact</b>	1 valeur	<b>0</b> pour *, <b>1</b> pour **, <b>2</b> pour ***, <b>3</b> pour \$, <b>4</b> pour \$\$, <b>5</b> pour \$\$\$	4
<b>status_since</b>	1 valeur	<i>epoch ( en secondes )</i>	1606155095
<b>status_confirmed_since</b>	1 valeur	<i>epoch ( en secondes ) (0 si état non confirmé)</i>	1606155095
<b>last_check</b>	1 valeur	<i>epoch ( en secondes )</i>	1606991424
<b>next_check</b>	1 valeur	<i>epoch ( en secondes )</i>	1606991444

### Le reste des propriétés apparaissant dans le retour

Nom	Sortie	Valeur possible	Exemple d'expression
<b>attempts</b>	1 valeur	<i>entier</i>	2
<b>max_attempts</b>	1 valeur	<i>entier</i>	2
<b>output</b>	1 valeur	<i>texte</i>	"PING OK - Packet loss = 0%, RTA = 0.01 ms"
<b>long_output</b>	1 valeur	<i>texte</i>	"<div class=\"skn-ich\">Module info:</div> <div class=\"skn-tbl-57069\">Some DATA</div>"
<b>perf_data</b>	1 valeur	<i>tableau JSON</i>	[ { "name": "rta", "min": 0, "max": null, "value": 0.013, "warning": 1000, "critical": 3000, "uom": "ms" }, { "name": "pl", "min": 0, "max": 100, "value": 0, "warning": 100, "critical": 100, "uom": "%" } ]
<b>raw_perf_data</b>	1 valeur	<i>texte</i>	"rta=0.013000ms;1000.000000;3000.000000;0.000000 pl=0%;100;100;0"
<b>notification_options</b>	1 valeur	<i>tableau JSON</i>	[ "w", "u", "c", "r", "f" ]
<b>notes_url</b>	1 valeur	<i>texte</i>	"http://localhost/"
<b>notes_multi_url</b>	1 valeur	<i>texte</i>	"UI_Configuration~http://localhost:7766~write~NO~#~UI_Visualisation~http://localhost:7767~stats~OPEN_IN_NEW_TAB"
<b>notification_period</b>	1 valeur	<i>texte</i>	"24x7"
<b>thresholds_display</b>	1 valeur	<i>texte</i>	"Warning 70%\nCritique 90%"

### Exemple d'utilisation

Exemple de définition du paramètre <code>ouput_field</code>
"ouput_field=status-context~father_name~check_name~output~perf_data~status_since~last_check"

Note : l'ordre dans lequel sont cités les propriétés ne change pas le format de sortie.

### Exemple complet d'appel

### Exemple par Appel curl

```
curl -s -S -H 'x-api-token: XYZ' -d 'filter0=check_name:CPU Stats' -d 'filter1=check_name:Disks' -d  
"sort=father_name:desc-status:asc" -d "output_format=elements_on_same_level" http://broker-server:50100/api  
/v2/inventory
```

-s, alias de --silent, ne pas afficher les barres de progression, n'affiche que les données récupérées  
-S, alias de --show-error, afficher quand même les messages d'erreurs  
-H, alias de --header, inclure ce header à la requête HTTP  
-d, alias de --data, envoie les données spécifiées en requête POST au serveur HTTP