

V2 - (READ) /api/v2/check/{ID}

Sommaire

- Objectifs
- Paramètres
- Réponse
 - Codes de retour
 - Retour du code 200
 - Retour du code 400
 - Paramètre GET absent (ID)
 - Le paramètre attendu est un nom
 - Le paramètre attendu est un UUID
 - Paramètres POST incorrects
 - Paramètre inconnu
 - Paramètre désactivé sur cette route
 - Messages d'erreurs lors du paramétrage des propriétés présentes dans la sortie (output_field)
 - Propriété de sortie inexistante
 - Messages d'erreurs liés au paramètre is_name
 - Valeur incorrecte
- Erreurs communes lors de l'envoi de la requête
 - Messages d'erreurs liés au protocole HTTPS
 - Le certificat SSL a été refusé
 - Requête HTTP sur un serveur en HTTPS
 - Requête HTTPS sur un serveur en HTTP

Objectifs

Méthode POST de type READ qui permet de récupérer des données de supervision d'un check et éventuellement de son père comme sur l'Interface de Visualisation.

Paramètres

Le paramètre **ID** reçu dans l'URL identifie l'élément à rechercher.

Sa nature (nom ou UUID) est définie à l'aide du paramètre **is_name** (paramètre POST, détaillé ci-dessous).



Certains caractères sont interdits dans les URL, ils doivent être échappés (URL encodées) pour pouvoir être utilisés

caractère interdit	:	/	?	#	[]	@	!	\$	&	'	()	*	+	,	;	=	%	(espace)	
remplacement		%3A	%2F	%3F	%23	%5B	%5D	%40	%21	%24	%26	%27	%28	%29	%2A	%2B	%2C	%3B	%3D	%25	%20 ou +

Pour plus d'information <https://developer.mozilla.org/fr/docs/Glossary/percent-encoding> et [rfc3986](https://tools.ietf.org/html/rfc3986)

Exemple: pour utiliser le nom **ch@nged_name** il faudra utiliser l'url **http://broker-module-livedata:50100/api/v2/check/ch%40nged_name**

Les 2 paramètres suivants sont également disponibles :

- **output_field** (Voir la page [V2 - Les paramètres des API du broker-module-livedata](#), pour leur description complète)
- **is_name**: précisant la nature de l'**ID** mentionné dans l'URL

Nom	Type	Défaut	Commentaire
is_name	Booléen	true	Paramètre dans le corps de la requête POST : <ul style="list-style-type: none">• true, ID est considéré comme un nom de check pour la récupération du check• false, ID est considéré comme un UUID pour la récupération du check
ID	text		Paramètre dans l'URL de la requête POST : Si is_name est sur false : ID est composé de l'uuid de l'hôte suivi de l'uuid du check associé, séparés par un -. Exemple : 693693aa225011efbf01080027db8bdb-f77a15d8225011efa15d080027db8bdb

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 suffisants.
404	L'hôte demandé n'existe pas
405	L'élément correspondant à l'ID n'est pas un hôte
500	L'appel est valide, mais un problème d'exécution est survenu.

Retour du code 200

Les propriétés retournées doivent être choisies avec l'option **output_field**

Mais les propriétés suivantes seront au minimum automatiquement retournées :

- **type**,
- **father_uuid**,
- **father_name**,
- **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 toutes les propriétés pouvant être renvoyée.

```
curl -s -S -H 'x-api-token: XYZ' \  
http://broker-module-livedata:50100/api/v2/check/CPU%20Stats
```

Exemple de sortie attendue :

<http://localhost:50100/api/v2/check/2c6dcf1aba1611ebaa7d0800277faebe-c296d75e5ad911e58cc5080027f08538>

```
{  
  "request_statistics": {  
    "nb_elements_total": 9,  
    "nb_checks_total": 4,  
    "nb_elements_filtered": 2,  
    "nb_checks_filtered": 2  
  },  
  "elements_found": [  
    {  
      "check_name": "CPU Stats",  
      "check_uuid": "2c6dcf1aba1611ebaa7d0800277faebe-c296d75e5ad911e58cc5080027f08538",  
      "father_name": "Bordeaux",  
      "father_uuid": "2c6dcf1aba1611ebaa7d0800277faebe",  
      "type": "check_host"  
    },  
    {  
      "check_name": "CPU Stats",  
      "check_uuid": "76f45d80ba1e11eba2670800277faebe-c296d75e5ad911e58cc5080027f08538",  
      "father_name": "Nantes",  
      "father_uuid": "76f45d80ba1e11eba2670800277faebe",  
      "type": "check_host"  
    }  
  ]  
}
```

Retour du code 400

Paramètre GET absent (**ID**)

Le paramètre attendu est un nom

```
$ curl -s -S -H "x-api-token: XYZ" \  
http://broker-module-livodata:50100/api/v2/check  
ERROR 400: mandatory parameter [ name ] is missing
```

```
ERROR 400: mandatory parameter [ name ] is missing
```

Le paramètre attendu est un UUID

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "is_name=false" \  
http://broker-module-livodata:50100/api/v2/check  
ERROR 400: mandatory parameter [ uuid ] is missing
```

```
ERROR 400: mandatory parameter [ uuid ] is missing
```

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/check/ID  
ERROR 400: POST parameter [ parametre_inconnu ] is  
unknown
```

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

Paramètre désactivé sur cette route

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "output_format=checks_attached_to_father" \  
http://broker-module-livodata:50100/api/v2/check/ID  
ERROR 400: POST parameter [ output_format ] is not  
available for this route
```

```
ERROR 400: POST parameter [ output_format ] is not available for  
this route
```

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/check/ID  
ERROR 400: output_field: invalid field name [ is_status_ ]
```

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

Messages d'erreurs liés au paramètre [is_name](#)

Valeur incorrecte

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "is_name=cluster" \  
http://broker-module-livodata:50100/api/v2/check/ID  
ERROR 400: is_name: invalid value [cluster]
```

```
ERROR 400: is_name: invalid value [cluster]
```

Erreurs communes lors de l'envoi de la requête

Messages d'erreurs liés au protocole HTTPS

Le certificat SSL a été refusé

```
$ curl -s -S -H "x-api-token: XYZ" \  
https://broker-module-livedata:50100/api/v2/inventory  
curl: (60) SSL certificate problem: unable to get  
local issuer certificate  
...
```



Cela peut se produire si le certificat du serveur est auto-signé.

Pour contourner la validité de l'émetteur du certificat, il faut utiliser l'option **--insecure** (*option courte : -k*).

```
curl --insecure -s -S -H "x-api-token: XYZ" \  
https://broker-module-livedata:50100/api/v2  
/inventory
```

Avec curl v7.29.0 :

```
curl: (60) Peer's Certificate issuer is not recognized.  
...
```

Avec curl v7.60 et supérieur :

```
curl: (60) SSL certificate problem: unable to get local issuer certificate  
...
```

Requête HTTP sur un serveur en HTTPS

```
$ curl -s -S -H "x-api-token: XYZ" \  
http://broker-module-livedata:50100/api/v2/inventory  
The client IP_SERVEUR:PORT_CLIENT sent a plain HTTP  
request,  
but this server only speaks HTTPS on this port.
```

```
The client IP_SERVEUR:PORT_CLIENT sent a plain HTTP request,  
but this server only speaks HTTPS on this port.
```

Requête HTTPS sur un serveur en HTTP

```
$ curl -s -S -H "x-api-token: XYZ" \  
https://broker-module-livedata:50100/api/v2/inventory  
curl: (35) error:1408F10B:SSL routines:  
ssl3_get_record:wrong version number
```

Avec curl v7.29.0 :

```
curl: (35) SSL received a record that exceeded the maximum  
permissible length.
```

Avec curl v7.60 et supérieur :

```
curl: (35) error:1408F10B:SSL routines:ssl3_get_record:wrong version  
number
```