

V2 - (READ) /api/v2/sla -- OPTIONNEL --

Sommaire

Objectifs

Paramètres

- filterX (Filtres)
 - output_format (Format de retour de la requête)
 - output_field (Informations présentes dans le retour de la requête)
 - period (entre quelles dates de début et de fin, prendre les données SLA)
 - page_settings (combien d'éléments par page et quelle page retourner)
- Exemple

Réponse

Codes de retour

Retour du code 200

- Exemple 1 : output_format=checks_attached_to_father
- Exemple 2 : output_format=elements_on_same_level
- Exemple 3 : output_format=list_of_sla

Retour du code 400

Paramètres POST incorrects

- Paramètre inconnu
- Paramètre désactivé sur cette route

Messages d'erreurs des filtres (filterX)

- Filtre inexistant
- Filtre incomplet
- Filtre existant, mais non autorisé
- Valeur incorrecte pour ce type de filtre

Messages d'erreurs du format de retour de la requête (output_format)

- Valeur invalide

Messages d'erreurs des propriétés présentes dans la sortie (output_field)

- Propriété de sortie existante, mais non autorisé
- Propriété de sortie inexistante

Messages d'erreurs de la période sur laquelle récupérer les données SLA (period)

- Date de début inférieure à la date de la plus ancienne donnée SLA enregistrée dans la base MongoDB
- Date de fin supérieur à la date de la veille
- Date de fin inférieur à la date de début saisie

Messages d'erreurs de la pagination (page_settings)

- Mauvaise valeur pour le paramètre page
- Mauvaise valeur pour le paramètre size
- Pas de valeur pour le paramètre size
- Paramètre size manquant

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

Le module `livedata_module_sla_provider` permet par le biais d'une URL (*Méthode POST de type READ*) de recevoir la liste des données SLA de tous les éléments (*hôtes, clusters et checks*) :

- Filtrées (*optionnel*),
- Rangées,
 - En arbres (*hôtes/clusters checks*),
 - Tous au même niveau,
 - Seulement les données SLA des éléments demandés.
- En choisissant :
 - Les informations présentes dans le retour de la requête (*optionnel*),
 - La période sur laquelle les données SLA seront récupérées (*optionnel*),
 - Les données SLA sont calculées à la fin de la journée, donc **la dernière donnée disponible est celle d'hier**,
 - Le nombre d'éléments par page (*optionnel*).
- Les données SLA récupérées sont triées dans l'ordre chronologie (du plus récent au plus vieux).



Pré-requis

Le module de type `livedata_module_sla_provider` doit être activé sur le `broker-module-livedata` pour que la route `/api/v2/sla/` soit accessible.

La configuration du module se trouve par défaut dans le fichier suivant : `/etc/shinken/modules/livedata-module-sla-provider.cfg` (voir la page [Module livedata-module-sla-provider](#)).

Paramètres

Pour définir l'appel, 5 paramètres sont disponibles :

- Standards :
 - `filterX`
 - `output_format`
 - `output_field`
- Spécifiques :
 - `period`
 - `page_settings`

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.

Les **critères** suivants sont utilisables :

- `type`
- `father_name`
- `father_uuid`
- `father_templates`
- `check_name`
- `check_uuid`
- `address`
- `realm`
- `host_groups`
- `notification_contacts`
- `notification_contact_groups`
- `business_impact`

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

output_format (Format de retour de la requête)

Ce paramètre permet de définir quel format de retour est utilisé (Il en existe 3):

- **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*)
- **list_of_sla** : Seules les données SLA sont listées (*une liste*)

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

- Si le **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** ou **output_format = list_of_sla**, les hôtes et clusters ne sont pas présents.



Par défaut, la valeur est `"elements_on_same_level"`

output_field (Informations présentes dans le retour de la requête)

Ce paramètre permet de lister les propriétés qui seront affichées dans le résultat.


- Les propriétés présentes par défaut sont :
 - **father_name**
 - **father_uuid**
 - **check_name**
 - **check_uuid**

- Les propriétés suivantes sont propres à cette route et sont présentes par défaut :

Nom	Format	Description
sla_total	Secondes	Temps total de SLA (86400 secondes étant 1 journée complète)
sla_missing	Secondes	Temps en statut Données manquantes
sla_ok	Secondes	Temps en statut OK
sla_inactive	Secondes	Temps en statut Shinken Inactive
sla_unknown	Secondes	Temps en statut Inconnu
sla_crit	Secondes	Temps en statut Critique
sla_warn	Secondes	Temps en statut Attention
sla_thresholds	Liste de pourcentages	Deux pourcentages : <ul style="list-style-type: none"> • La première valeur est le seuil d'avertissement • La deuxième valeur est le seuil de critique Les pourcentages ont une précision à 3 chiffres (<i>ex: 90.001</i>)
sla_date	Chaîne de caractères	au format aaaa_mm_jj (<i>ex: 2021_05_12</i>)

- Les propriétés présentes peuvent être les suivants :
 - **type**

 La propriété **type** fait partie des propriétés présentes par défaut pour les formats de retour "*elements_on_same_level*" et "*list_of_sla*".

 Les propriétés ci-dessus sont décrites dans la page [V2 - Les paramètres de l'API du module livedata-module-sla-provider du broker-module-livedata](#)

period (entre quelles dates de début et de fin, prendre les données SLA)

Nom	Valeur par défaut	Description et syntaxe
period=start:date~end:date	La date de la veille	Défini la période où collecter les données SLA : <ul style="list-style-type: none"> • Les dates sont au format aaaa_mm_jj (<i>ex: 2021_05_12</i>) • Si start n'est pas précisé, cela signifie que le début de la période est la date de la veille. • Si end n'est pas précisé, la fin de la période est égal à la date de début.

Trois règles devront être respectées :

- La date de départ ne peut pas être antérieure à la date de la première donnée SLA disponible.
- La date de départ ne peut pas être supérieure à la date de la veille.
- La date de fin ne peut pas être supérieure à la date de début.

page_settings (combien d'éléments par page et quelle page retourner)

L'API peut, grâce à ce paramètre, définir le nombre d'éléments par page et le numéro de la page retournée, ce qui permet de contrôler le volume d'échange de données. Ceci est possible vu que les données SLA sont figées en base de données.

Le champ **has_next_page** dans la partie **pagination** du retour permet de savoir s'il y a une page suivante.

Nom	Valeur par défaut	Info
page_settings=page:page_index~nb_element:size	Le nombre d'éléments par défaut d'une page est 100	<ul style="list-style-type: none">• nb_element étant la taille de la page• page étant l'index de page demandée. Les indexes de page commencent à 0



Si **output_format** est à *checks_attached_to_father*, le nombre d'éléments par page correspondra aux hôtes / clusters.

Si **output_format** est à *elements_on_same_level*, le nombre d'éléments par page correspondra aux hôtes / clusters / checks.

Si **output_format** est à *list_of_sla*, le nombre d'éléments par page correspondra aux données SLA des hôtes / clusters / checks.

Exemple

Exemple permettant d'obtenir la première page d'une requête renvoyant 100 éléments avec leurs données SLA, du début de l'année 2021 au 1er mai 2021.

```
curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:2021_01_01~end:2021_05_01" \  
-d "page_settings=page:0~nb_element:100" \  
http://broker-module-livodata:50100/api/v2/sla
```

Exemple permettant d'obtenir la quatrième page d'une requête renvoyant 100 éléments avec leurs données SLA, du début de l'année 2021 au 1er mai 2021.

```
curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:2021_01_01~end:2021_05_01" \  
-d "page_settings=page:4~nb_element:100" \  
http://broker-module-livodata:50100/api/v2/sla
```

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 survenu.

Retour du code 200

En premier apparaîtra des informations donnant le nombre d'éléments :

- **request_statistics**
 - **nb_elements_total**
 - **nb_hosts_total**
 - **nb_clusters_total**
 - **nb_checks_total**
 - **nb_elements_filtered**
 - **nb_hosts_filtered**
 - **nb_clusters_filtered**
 - **nb_checks_filtered**
 - **nb_elements_in_page**
 - **nb_hosts_in_page**
 - **nb_clusters_in_page**

- nb_checks_in_page

Avec une information supplémentaire propre à la route :

Nom	Format	Description
nb_sla_in_page	Entier	Nombre de données SLA présentes dans la page

En deuxième, les données de pagination vont être retournées dans le format suivant :

- pagination
 - has_next_page
 - nb_total_page
 - page
 - page_size

Les propriétés présentes pour chaque élément retourné possédant des données SLA doivent être choisis avec l'option **output_field**, mais les propriétés suivantes sont au minimum automatiquement retournées :

Output_format à checks_attached_to_father :

- elements_found
 - hosts
 - father_name
 - father_uuid
 - checks
 - check_name
 - check_uuid
 - sla
 - sla_total
 - sla_missing
 - sla_ok
 - sla_inactive
 - sla_unknown
 - sla_crit
 - sla_warn
 - sla_thresholds
 - sla_date

Output_format à elements_on_same_level :

- elements_found
 - type
 - father_name
 - father_uuid
 - check_name
 - check_uuid
 - sla
 - sla_total
 - sla_missing
 - sla_ok
 - sla_inactive
 - sla_unknown
 - sla_crit
 - sla_warn
 - sla_thresholds
 - sla_date
- elements_not_found

Output_format à list_of_sla :

- elements_found
 - type
 - father_name
 - father_uuid
 - check_name
 - check_uuid
 - sla_total
 - sla_missing
 - sla_ok
 - sla_inactive
 - sla_unknown
 - sla_crit
 - sla_warn
 - sla_thresholds
 - sla_date
- elements_not_found

d
ate

- elements_not_found

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 renvoyées. Pour les éléments non trouvés, les critères utilisés dans les filtres seront renvoyés dans la description des l'élément.

Exemple 1 : output_format=checks_attached_to_father

```
curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:2021_05_24~end:2021_05_25" \  
-d "output_format=checks_attached_to_father" \  
-d "filter01=type:check" \  
-d "page_settings=page:0~size:2" \  
http://broker-module-livedata:50100/api/v2/sla
```

Exemple de sortie attendue :

```
{  
  "request_statistics": {  
    "nb_elements_total": 34,  
    "nb_hosts_total": 3,  
    "nb_clusters_total": 1,  
    "nb_checks_total": 30,  
    "nb_elements_filtered": 4,  
    "nb_hosts_filtered": 1,  
    "nb_clusters_filtered": 1,  
    "nb_checks_filtered": 2,  
    "nb_elements_in_page": 4,  
    "nb_hosts_in_page": 1,  
    "nb_clusters_in_page": 1,  
    "nb_checks_in_page": 2,  
    "nb_sla_in_page": 4  
  },  
  "pagination": {  
    "has_next_page": true,  
    "nb_total_page": 2,  
    "page": 0,  
    "page_size": 2  
  },  
  "elements_found": {  
    "clusters": [{  
      "father_uuid": "12760f56bc6d11eb85a3080027c44e8f",  
      "father_name": "Cluster 01",  
      "checks": [{  
        "check_name": "Check Cluster 01",  
        "check_uuid": "12760f56bc6d11eb85a3080027c44e8f-9d86c522bd3511ebb58c080027c44e8f",  
        "sla": [{  
          "sla_date": "2021_05_24",  
          "sla_total": 86400,  
          "sla_warn": 0,  
          "sla_unknown": 0,  
          "sla_thresholds": [99.0, 97.0],  
          "sla_missing": 0,  
          "sla_ok": 0,  
          "sla_inactive": 86400,  
          "sla_crit": 0  
        }, {  
          "sla_date": "2021_05_25",  
          "sla_total": 47616,  
          "sla_warn": 0,  
          "sla_unknown": 0,  
          "sla_thresholds": [99.0, 97.0],  
          "sla_missing": 0,  
          "sla_ok": 0,  
          "sla_inactive": 47616,  
          "sla_crit": 0  
        }  
      ]  
    }  
  ]  
}
```



```
    "nb_checks_in_page": 2,
    "nb_sla_in_page": 4
  },
  "pagination": {
    "has_next_page": true,
    "nb_total_page": 15,
    "page": 0,
    "page_size": 2
  },
  "elements_found": [{
    "check_name": "Check Cluster 01",
    "type": "check_cluster",
    "father_uuid": "12760f56bc6d11eb85a3080027c44e8f",
    "father_name": "Cluster 01",
    "check_uuid": "12760f56bc6d11eb85a3080027c44e8f-9d86c522bd3511ebb58c080027c44e8f",
    "sla": [{
      "sla_date": "2021_05_24",
      "sla_total": 86400,
      "sla_warn": 0,
      "sla_unknown": 0,
      "sla_thresholds": [99.0, 97.0],
      "sla_missing": 0,
      "sla_ok": 0,
      "sla_inactive": 86400,
      "sla_crit": 0
    }, {
      "sla_date": "2021_05_25",
      "sla_total": 51527,
      "sla_warn": 0,
      "sla_unknown": 24183,
      "sla_thresholds": [99.0, 97.0],
      "sla_missing": 3916,
      "sla_ok": 0,
      "sla_inactive": 23428,
      "sla_crit": 0
    }
  ]
}, {
  "check_name": "Check 01",
  "type": "check_host",
  "father_uuid": "f87c2e56b94b11ebaf7e080027c44e8f",
  "father_name": "Host 01",
  "check_uuid": "f87c2e56b94b11ebaf7e080027c44e8f-fdd0c038b94b11ebb21f080027c44e8f",
  "sla": [{
    "sla_date": "2021_05_24",
    "sla_total": 86400,
    "sla_warn": 0,
    "sla_unknown": 0,
    "sla_thresholds": [99.0, 97.0],
    "sla_missing": 165,
    "sla_ok": 27344,
    "sla_inactive": 58891,
    "sla_crit": 0
  }, {
    "sla_date": "2021_05_25",
    "sla_total": 86400,
    "sla_warn": 0,
    "sla_unknown": 0,
    "sla_thresholds": [58.654, 32.451],
    "sla_missing": 245,
    "sla_ok": 27881,
    "sla_inactive": 58274,
    "sla_crit": 0
  }
]
}
}
```

Exemple 3 : output_format=list_of_sla

```
curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:2021_05_24~end:2021_05_25" \  
-d "output_format=list_of_sla" \  
-d "filter01=type:check" \  
-d "page_settings=page:0~size:2" \  
http://broker-module-livedata:50100/api/v2/sla
```

Exemple de sortie attendue :

```
{  
  "request_statistics": {  
    "nb_elements_total": 34,  
    "nb_hosts_total": 3,  
    "nb_clusters_total": 1,  
    "nb_checks_total": 30,  
    "nb_elements_filtered": 30,  
    "nb_hosts_filtered": 0,  
    "nb_clusters_filtered": 0,  
    "nb_checks_filtered": 30,  
    "nb_elements_in_page": 1,  
    "nb_hosts_in_page": 0,  
    "nb_clusters_in_page": 0,  
    "nb_checks_in_page": 1,  
    "nb_sla_in_page": 2  
  },  
  "pagination": {  
    "has_next_page": "true",  
    "nb_total_page": 30,  
    "page": 0,  
    "page_size": 2  
  },  
  "elements_found": [{  
    "sla_total": 86400,  
    "sla_thresholds": [99.0, 97.0],  
    "check_name": "Check Cluster 01",  
    "check_uuid": "12760f56bc6d11eb85a3080027c44e8f-9d86c522bd3511ebb58c080027c44e8f",  
    "father_uuid": "12760f56bc6d11eb85a3080027c44e8f",  
    "sla_ok": 0,  
    "sla_inactive": 86400,  
    "sla_warn": 0,  
    "sla_date": "2021_05_24",  
    "father_name": "Cluster 01",  
    "sla_unknown": 0,  
    "sla_missing": 0,  
    "type": "check_cluster",  
    "sla_crit": 0  
  }, {  
    "sla_total": 51527,  
    "sla_thresholds": [99.0, 97.0],  
    "check_name": "Check Cluster 01",  
    "type": "check_cluster",  
    "father_uuid": "12760f56bc6d11eb85a3080027c44e8f",  
    "sla_ok": 0,  
    "sla_inactive": 23428,  
    "sla_crit": 0,  
    "sla_date": "2021_05_25",  
    "father_name": "Cluster 01",  
    "sla_unknown": 24183,  
    "sla_missing": 3916,  
    "check_uuid": "12760f56bc6d11eb85a3080027c44e8f-9d86c522bd3511ebb58c080027c44e8f",  
    "sla_warn": 0  
  }  
]  
}
```

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/sla  
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 "sort=father_name:desc" \  
http://broker-module-livedata:50100/api/v2/sla  
ERROR 400: POST parameter [ sort ] is not  
available for this route
```

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

Messages d'erreurs des filtres (filterX)

Filtre inexistant

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "filter01=is_status:true" \  
http://broker-module-livedata:50100/api/v2/sla  
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/sla  
ERROR 400: filtering[0]: missing value for field [ next_check ]
```

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

Filtre existant, mais non autorisé

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "filter01=is_status_confirmed:true" \  
http://broker-module-livedata:50100/api/v2/sla  
ERROR 400: filtering[0]: field name [ is_status_confirmed ] not  
allowed in this route
```

```
ERROR 400: filtering[0]: field name [ is_status_confirmed ] not  
allowed in this route
```

Valeur incorrecte pour ce type de filtre

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

```
ERROR 400: filtering[0]: field [ business_impact ] => wrong value  
['cinq']
```

Messages d'erreurs du format de retour de la requête (output_format)

Valeur invalide

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

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "output_format=all_elements" \  
http://broker-module-livedata:50100/api/v1/sla  
ERROR 400: output_format: invalid value [  
all_elements ]
```

Messages d'erreurs des propriétés présentes dans la sortie (output_field)

Propriété de sortie existante, mais non autorisé

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "output_field=is_status_confirmed" \  
http://broker-module-livedata:50100/api/v2/sla  
ERROR 400: output_field: field name [  
is_status_confirmed ] not allowed in this route
```

```
ERROR 400: output_field: field name [ is_status_confirmed ] not  
allowed in this route
```

Propriété de sortie inexistante

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

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

Messages d'erreurs de la période sur laquelle récupérer les données SLA (period)

Date de début inférieure à la date de la plus ancienne donnée SLA enregistrée dans la base MongoDB

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:2021_01_01~end:2021_05_25" \  
http://broker-module-livedata:50100/api/v2/sla  
ERROR 400: period: the start period is not valid,  
as there is no SLA data for this date. You can  
filter elements from 2021_02_13.
```

```
ERROR 400: period: the start period is not valid, as there is no SLA  
data for this date. You can filter elements from 2021_02_13.
```

Date de fin supérieur à la date de la veille

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:2021_01_01~end:2021_05_26" \  
http://broker-module-livedata:50100/api/v2/sla  
ERROR 400: period: the end period is invalid, as  
the requested period is in the future. You can  
filter elements until 2021_05_25.
```

```
ERROR 400: period: the end period is invalid, as the requested  
period is in the future. You can filter elements until 2021_05_25.
```

Date de fin inférieur à la date de début saisie

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:2021_05_25~end:2021_05_24" \  
http://broker-module-livedata:50100/api/v2/sla  
ERROR 400: period: the end period is invalid, as  
it's less than the start period.
```

```
ERROR 400: period: the end period is invalid, as it's less than the  
start period.
```

Messages d'erreurs de la pagination (page_settings)

Mauvaise valeur pour le paramètre page

```
ERROR 400: Wrong value:[ cinq ] for page number parameter
```

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "page_settings=page:cinq~size:2" \  
http://broker-module-livedata:50100/api/v2/sla  
ERROR 400: Wrong value:[ cinq ] for page number  
parameter
```

Mauvaise valeur pour le paramètre size

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "page_settings=page:5~size:deux" \  
http://broker-module-livedata:50100/api/v2/sla  
ERROR 400: Wrong value:[ deux ] for page size  
parameter
```

```
ERROR 400: Wrong value:[ deux ] for page size parameter
```

Pas de valeur pour le paramètre size

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "page_settings=page:5~size" \  
http://broker-module-livedata:50100/api/v2/sla  
ERROR 400: Missing page size parameter
```

```
ERROR 400: Missing page size parameter
```

Paramètre size manquant

```
$ curl -s -S -H "x-api-token: XYZ" \  
-d "page_settings=page:5" \  
http://broker-module-livedata:50100/api/v2/sla  
ERROR 400: Wrong format for pagination parameters,  
expected:[name:integer~name:integer], got:[ page:2  
]
```

```
ERROR 400: Wrong format for pagination parameters, expected:  
[name:integer~name:integer], got:[ page:2 ]
```

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
```