

Module de source discovery (de type discovery-import)

Sommaire

- Concept
- Activation du collecteur
- Configuration
 - Fichier de configuration
 - Détails des sections composant le fichier de configuration
 - Identification du module
 - Fonctionnement de la source
 - Options d'accès à la base de données
 - Options du module

Objectifs



Pré-requis

Le module **livedata-module-sla-provider** doit être activé sur le **broker-module-livedata** pour que la route **/api/v1/sla/** soit accessible.

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

Méthode POST de type READ qui permet 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
- 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 de la veille,
 - le nombre d'élément par page (*optionnel*).
- Les données SLA récupérées sont triées dans l'ordre chronologie (du plus récent au plus vieux).

Paramètres

Pour définir l'appel, 5 paramètres sont disponibles :

- standards :
 - **filterX**
 - **checks_in_tree**
 - **ouput_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**

Vous pouvez trouver la description de ces filtres dans la page [V1 - Les paramètres de l'API du module livedata-module-sla-provider du broker-module-livedata](#)

checks_in_tree (Rangées sous forme d'arbre)

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

- **true** : les checks sont accrochés à leur hôtes / clusters (*forme d'arbre*)
- **false** : les checks sont listés au niveaux des hôtes / clusters (*une liste*)

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

- mais où **checks_in_tree = true**, les hôtes / clusters seront quand même présents pour les checks correspondant à ce filtre.
- si le **checks_in_tree = false**, 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 champs qui seront affichés sur le résultat en sortie.

- Les champs présents par défaut sont :

Nom	Description
type	Type de l'élément
father_name	Nom de l'hôte / cluster
father_uuid	UUID de l'hôte / cluster
check_name	Nom du check
check_uuid	UUID du check

- Les champs suivants peuvent être utilisés dans ce paramètre :

Nom	Description
father_templates	Noms des modèles d'hôtes
address	Adresse IP de l'hôte
realm	Royaume de l'élément
host_groups	Groupe d'hôte
notification_contacts	Noms des contacts à notifier
notification_contact_groups	Noms des groupes de contacts à notifier
business_impact	Impact métier

- Les champs ci-dessus sont décrits dans la page [V1 - Les paramètres de l'API du module livedata-module-sla-provider du broker-module-livedata](#)

- Les champs suivants sont propres à cette route et sont présent par défaut :

Nom	Format	Description
sla_total	Secondes	Temps totale 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"> • le première valeur est seuil d'avertissement • la deuxième valeur est le seuil de critique
sla_date	Chaîne de caractères	au format jj_mm_aaaa

period (entre quel date de début et de fin, prendre les données SLA)

Nom	Valeur par défaut	Description et syntaxe
period=start:date~end:date	La dernière heure	Défini la période ou collecter les données SLA <ul style="list-style-type: none"> • Les dates sont au format jj_mm_aaaa • Si start n'est pas précisé, cela signifie que le début de la période est la date actuelle. • Si end n'est pas précisé, la fin de la période est maintenant.

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 si il y a une page suivante.

Nom	Valeur par défaut	Info
page_settings=nb_element:size	Le nombre d'élément par défaut d'une page est 100	Pour la première requête <ul style="list-style-type: none"> • nb_element étant la taille de la page
page_settings=page:page_index~nb_element:size	Le nombre d'élément 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 le **check_in_tree** est à **True**, le nombre d'éléments par page correspondra aux hôtes / clusters.

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

Exemple

Pour obtenir la 4e page d'une requête renvoyant les données sla, de l'année 2020 à aujourd'hui

Première requête pour créer l'ensemble de résultat :

```
curl -s -S -H "x-api-token: XYZ" \
-d "filter0=business_impact:3" \
-d "page_settings=nb_element:100" \
http://broker-module-livodata:50100/api/v1/sla
```

Requête pour accéder à la 4ème page de l'ensemble de résultat :

```
curl -s -S -H "x-api-token: XYZ" \
-d "page_settings=page:4~nb_element:100" \
http://broker-module-livodata:50100/api/v1/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 information donnant le nombre d'éléments :

- **elements_count**
 - nb_elements_total_in_query
 - nb_elements_found
 - nb_elements_not_found
 - nb_fathers_total_in_query
 - nb_fathers_found
 - nb_fathers_not_found
 - nb_checks_total_in_query
 - nb_checks_found
 - nb_checks_not_found

Avec deux informations supplémentaires propres à la route :

Nom	Format	Description
nb_sla_total_in_query	Entier	Nombre de sla total (Sans la prise en compte des filtres)
nb_sla_found	Entier	Nombre de sla trouvés (Prise en compte des filtres)

Les données SLA vont être retournées dans le format suivant :

- **pagination**
 - has_next_page
 - nb_total_page
 - nb_element_in_page
 - page
 - page_size

Les champs présents pour chaque élément retourné possédant des données SLA doivent être choisis avec l'option **output_field**, mais les champs suivants sont au minimum automatiquement retournés :

Checks_in_tree à True :

- **fathers_found**
 - type
 - father_name
 - father_uuid
 - checks
 - type
 - check_name
 - check_uuid
 - sla
 - sla_total
 - sla_missing
 - sla_ok
 - sla_inactive
 - sla_unknown
 - sla_crit
 - sla_warn
 - sla_thresholds
 - sla_date
- **fathers_not_found**

Checks_in_tree à False :

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

Voir la page [V1 - Les champs présents dans le retour 200 des API du broker-module-livedata](#), pour la description complète de tous les champs pouvant être renvoyés. Pour les éléments non trouvés, les critères utilisés dans les filtres seront renvoyés dans la description des l'élément.

Exemple 1 : checks_in_tree=true

```
curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:01_03_2021~end:02_03_2021" \  
-d "filter0=type:CHECK_HOST" \  
-d "page_settings=page:3~nb_element:100" \  
-d "checks_in_tree=true" \  
http://broker-module-livedata:50100/api/v1/sla
```

```
{  
  "elements_count": {  
    "nb_elements_total_in_query": 1050,  
    "nb_elements_found": 666,  
    "nb_fathers_total_in_query": 250,  
    "nb_fathers_found": 201,  
    "nb_checks_total_in_query": 800,  
    "nb_checks_found": 465,  
    "nb_sla_total_in_query": 2100,  
    "nb_sla_found": 1344,  
  },  
  "pagination": {  
    "nb_total_page": 3,  
    "page": 3,  
    "page_size": 100,  
    "has_next_page": false,  
    "nb_element_in_page": 3  
  },  
  "fathers_found": [{  
    "type": "host",  
    "father_uuid": "5f058e7cad2d40dba0111045503480a8",  
    "father_name": "Broker Master",  
    "checks": [{  
      "type": "check_host",  
      "check_name": "Daemon Uptime",  
      "check_uuid": "5f058e7cad2d40dba0111045503480a8-d963f3e228c211eb9b7e080027774a8d",  
      "sla": [{  
        "sla_total": 86400,  
        "sla_missing": 0,  
        "sla_ok": 0,  
        "sla_inactive": 83777,  
        "sla_crit": 2623,  
        "sla_unknown": 0,  
        "sla_warn": 0,  
        "sla_thresholds": [99, 97],  
        "sla_date": "01_03_2021"  
      }, {  
        "sla_total": 86400,  
        "sla_missing": 0,  
        "sla_ok": 0,  
        "sla_inactive": 83577,  
        "sla_crit": 2823,  
        "sla_unknown": 0,  
        "sla_warn": 0,  
        "sla_thresholds": [99, 97],  
        "sla_date": "02_03_2021"  
      }  
    ]  
  }, {  
    "type": "check_host",  
    "check_name": "Daemon Alive",  
    "check_uuid": "5f058e7cad2d40dba0111045503480a8-d963f3e228c211eb9b7e080027774a8e",  
    "sla": [{  
      "sla_total": 86400,  
      "sla_missing": 0,  
      "sla_ok": 0,  
      "sla_inactive": 83577,  
      "sla_crit": 2823,  
      "sla_unknown": 0,  
      "sla_warn": 0,  
      "sla_thresholds": [99, 97],  
      "sla_date": "02_03_2021"  
    }  
  }  
}]
```

```

        "sla_ok": 0,
        "sla_inactive": 83777,
        "sla_crit": 2623,
        "sla_unknown": 0,
        "sla_warn": 0,
        "sla_thresholds": [99, 97],
        "sla_date": "01_03_2021"
    }, {
        "sla_total": 86400,
        "sla_missing": 0,
        "sla_ok": 0,
        "sla_inactive": 83577,
        "sla_crit": 2823,
        "sla_unknown": 0,
        "sla_warn": 0,
        "sla_thresholds": [99, 97],
        "sla_date": "02_03_2021"
    }
]
}
],
"fathers_not_found": []
}

```

Exemple 2 : checks_in_tree=false

```

curl -s -S -H "x-api-token: XYZ" \
-d "period=start:01_03_2021~end:02_03_2021" \
-d "filter0=type:CHECK_HOST" \
-d "page_settings=page:3~nb_element:100" \
-d "checks_in_tree=false" \
http://broker-module-livedata:50100/api/v1/sla

```

```

{
  "elements_count": {
    "nb_elements_total_in_query": 1050,
    "nb_elements_found": 666,
    "nb_fathers_total_in_query": 250,
    "nb_fathers_found": 201,
    "nb_checks_total_in_query": 800,
    "nb_checks_found": 465,
    "nb_sla_total_in_query": 2100,
    "nb_sla_found": 1344,
  },
  "pagination": {
    "nb_total_page": 3,
    "page": 3,
    "page_size": 100,
    "has_next_page": false,
    "nb_element_in_page": 2
  },
  "elements_found": [{
    "type": "check_host",
    "check_name": "Daemon Uptime",
    "check_uuid": "5f058e7cad2d40dba0111045503480a8-d963f3e228c211eb9b7e080027774a8d",
    "father_name": "Broker Master",
    "father_uuid": "5f058e7cad2d40dba0111045503480a8",
    "sla": [{
      "sla_total": 86400,
      "sla_missing": 0,
      "sla_ok": 0,
      "sla_inactive": 83777,
      "sla_crit": 2623,
      "sla_unknown": 0,
    }
  ]
}

```

```
        "sla_warn": 0,
        "sla_thresholds": [99, 97],
        "sla_date": "01_03_2021"
    }, {
        "sla_total": 86400,
        "sla_missing": 0,
        "sla_ok": 0,
        "sla_inactive": 83577,
        "sla_crit": 2823,
        "sla_unknown": 0,
        "sla_warn": 0,
        "sla_thresholds": [99, 97],
        "sla_date": "02_03_2021"
    }
]
}, {
    "type": "check_host",
    "check_name": "Daemon Alive",
    "check_uuid": "5f058e7cad2d40dba0111045503480a8-d963f3e228c211eb9b7e080027774a8e",
    "father_name": "Broker Master",
    "father_uuid": "5f058e7cad2d40dba0111045503480a8",
    "sla": [{
        "sla_total": 86400,
        "sla_missing": 0,
        "sla_ok": 0,
        "sla_inactive": 83777,
        "sla_crit": 2623,
        "sla_unknown": 0,
        "sla_warn": 0,
        "sla_thresholds": [99, 97],
        "sla_date": "01_03_2021"
    }, {
        "sla_total": 86400,
        "sla_missing": 0,
        "sla_ok": 0,
        "sla_inactive": 83577,
        "sla_crit": 2823,
        "sla_unknown": 0,
        "sla_warn": 0,
        "sla_thresholds": [99, 97],
        "sla_date": "02_03_2021"
    }
]
},
],
"elements_not_found": []
}
```