

Modèle shinken-broker

Sommaire

- Contexte
- Sommaire des checks
- Les données
 - Les données communes pour tous les checks
 - Provenant du modèle shinken
 - Provenant du modèle shinken-daemon
 - Provenant du modèle shinken-broker-db
 - Provenant du modèle shinken-broker
 - Les données spécifiques
 - Pour le check "Broker - DB - Connection"
 - Pour le check "Broker - DB - Last Flush Time"
 - Pour le check "Broker - DB - Connection"
 - Les données DFE (Duplicate Foreach)
- Comment appliquer un modèle d'hôte à un hôte
 - Application du modèle via l'interface de Configuration
 - Application du modèle via un collecteur d'import de fichiers au format .cfg

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'éléments 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**
 - **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 leurs hôtes / clusters (*forme d'arbre*)
- **false** : 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*)

- Si le **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 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 :
 - **father_name**
 - **father_uuid**
 - **check_name**
 - **check_uuid**
- Les champs pouvant être ajoutés par l'utilisateur
 - **type**



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 total de SLA (<i>86400 secondes étant 1 journée complète</i>)
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 jj_mm_aaaa (<i>ex: 12_05_2021</i>)

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

Nom	Valeur par défaut	Description et syntaxe
<code>period=start:date~end:date</code>	La dernière heure	Défini la période où collecter les données SLA <ul style="list-style-type: none">• Les dates sont au format jj_mm_aaaa (ex: 12_05_2021)• 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 s'il y a une page suivante.

Nom	Valeur par défaut	Info
<code>page_settings=nb_element:size</code>	Le nombre d'éléments 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
<code>page_settings=page:page_index~nb_element:size</code>	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 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

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=01_01_2021:~end:01_05_2021" \  
-d "page_settings=page:0~nb_element:100" \  
http://broker-module-livedata:50100/api/v1/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=01_01_2021:~end:01_05_2021" \  
-d "page_settings=page:4~nb_element:100" \  
http://broker-module-livedata: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 suffisants.
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 :

- elements_count
 - nb_elements_total
 - nb_elements_filtered
 - nb_elements_in_page
 - nb_elements_not_found
 - nb_fathers_total
 - nb_fathers_filtered
 - nb_fathers_in_page
 - nb_fathers_not_found
 - nb_checks_total
 - nb_checks_filtered
 - nb_checks_in_page
 - nb_checks_not_found

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 position, Les données SLA vont être retournées dans le format suivant :

- pagination
 - has_next_page
 - nb_total_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 :

Output_format à fathers:

- 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:

- 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 sla_entries:

- 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

- u
 - n
 - k
 - n
 - o
 - wn
 - sla
 -
 - c
 - rit
 - sla
 -
 - w
 - a
 - rn
- sla
- _t
- h
- r
- e
- s
- h
- ol
- ds
- sla
-
- d
- ate
- clusters
 - father_name
 - father_uuid
 - checks
 - check_name
 - check_uuid
 - sla
 - sla
 - _t
 - o
 - tal
 - sla
 -
 - m
 - is
 - si
 - ng
 - sla
 -
 - ok
- sla
- _i
- n
- a
- ct
- ive
- sla
-
- u
- n
- k
- n
- o
- wn
- sla
-
- c
- rit


```

{
  "request_statistics": {
    "nb_elements_found": 32,
    "nb_hosts_found": 3,
    "nb_checks_found": 29,
    "nb_clusters_found": 0,
    "nb_elements_filtered": 31,
    "nb_hosts_filtered": 2,
    "nb_clusters_filtered": 0,
    "nb_checks_filtered": 29,
    "nb_elements_in_page": 2,
    "nb_hosts_in_page": 1,
    "nb_clusters_in_page": 0,
    "nb_checks_in_page": 1,
    "nb_sla_in_page": 2
  },
  "pagination": {
    "nb_total_page": 2,
    "page": 0,
    "page_size": 1,
    "has_next_page": true
  },
  "elements_found": [
    "hosts": [{
      "father_name": "Host 01",
      "father_uuid": "f87c2e56b94b11ebaf7e080027c44e8f",
      "checks": [{
        "check_name": "Check 01",
        "check_uuid": "f87c2e56b94b11ebaf7e080027c44e8f-fdd0c038b94b11ebb21f080027c44e8f",
        "sla": [{
          "sla_date": "10_05_2021",
          "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": "11_05_2021",
          "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
        }
      ]
    }
  ]
}

```

Exemple 2 : output_format=elements

```

curl -s -S -H "x-api-token: XYZ" \
-d "period=start:10_05_2021~end:11_05_2021" \
-d "return_format=elements" \
-d "filter01=type:check" \
-d "page_settings=page:0~size:1" \
http://broker-module-livedata:50100/api/v1/sla

```

```

{
  "elements_count": {
    "nb_elements_found": 32,
    "nb_hosts_found": 3,
    "nb_clusters_found": 0,
    "nb_checks_found": 29,
    "nb_elements_filtered": 29,
    "nb_hosts_filtered": 0,
    "nb_clusters_filtered": 0,
    "nb_checks_filtered": 29,
    "nb_elements_in_page": 0,
    "nb_hosts_in_page": 0,
    "nb_clusters_in_page": 0,
    "nb_checks_in_page": 0,
    "nb_sla_in_page": 2
  },
  "pagination": {
    "nb_total_page": 29,
    "page": 0,
    "page_size": 1,
    "has_next_page": true
  },
  "elements_found": [{
    "check_name": "Check 01",
    "type": "check_host",
    "father_uuid": "f87c2e56b94b11ebaf7e080027c44e8f",
    "father_name": "Host 01",
    "check_uuid": "f87c2e56b94b11ebaf7e080027c44e8f-fdd0c038b94b11ebb21f080027c44e8f",
    "sla": [{
      "sla_date": "10_05_2021",
      "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": "11_05_2021",
      "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
    }
  ]
}
]
}

```

Exemple 3 : output_format=sla_entries

```
curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:10_05_2021~end:11_05_2021" \  
-d "return_format=sla_entry" \  
-d "filter01=type:check" \  
-d "page_settings=page:0~size:1" \  
http://broker-module-livedata:50100/api/v1/sla
```

```

{
  "elements_count": {
    "nb_elements_found": 32,
    "nb_hosts_found": 3,
    "nb_clusters_found": 0,
    "nb_checks_found": 29,
    "nb_elements_filtered": 29,
    "nb_hosts_filtered": 0,
    "nb_clusters_filtered": 0,
    "nb_checks_filtered": 29,
    "nb_elements_in_page": 0,
    "nb_hosts_in_page": 0,
    "nb_clusters_in_page": 0,
    "nb_checks_in_page": 2,
    "nb_sla_in_page": 2
  },
  "pagination": {
    "nb_total_page": 29,
    "page": 0,
    "page_size": 1,
    "has_next_page": true
  },
  "sla": [{
    "type": "check_host",
    "father_uuid": "f87c2e56b94b11ebaf7e080027c44e8f",
    "father_name": "Host 01",
    "check_name": "Check 01",
    "check_uuid": "f87c2e56b94b11ebaf7e080027c44e8f-fdd0c038b94b11ebb21f080027c44e8f",
    "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": "10_05_2021"
  }, {
    "type": "check_host",
    "father_uuid": "f87c2e56b94b11ebaf7e080027c44e8f",
    "father_name": "Host 01",
    "check_name": "Check 01",
    "check_uuid": "f87c2e56b94b11ebaf7e080027c44e8f-fdd0c038b94b11ebb21f080027c44e8f",
    "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": "11_05_2021"
  }
]
}

```