

# Configurer un widget - Édition visuelle - Météo

## Sommaire

- Concept
- Ouvrir la configuration
- Fonctionnement général
  - Regroupement par dossier des propriétés
  - Valeurs par défaut
  - Les différents types de propriété
    - Propriété de type texte
    - Propriété de type sélecteur
    - Propriété de type radio
    - Propriété de type radio et sélecteur
    - Propriété combinant type radio et champ texte
  - La configuration des différents types de widgets

## 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,
  - 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 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**
  - **output\_format**
  - **output\_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<sup>^</sup>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](#)

## output\_format ( Format de retour de la requête )

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* )
- **list\_of\_sla** : *Seul 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 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
<b>sla_total</b>	Secondes	Temps total de SLA ( <i>86400 secondes étant 1 journée complète</i> )
<b>sla_missing</b>	Secondes	Temps en statut <b>Données manquantes</b>
<b>sla_ok</b>	Secondes	Temps en statut <b>OK</b>
<b>sla_inactive</b>	Secondes	Temps en statut <b>Shinken Inactive</b>
<b>sla_unknown</b>	Secondes	Temps en statut <b>Inconnu</b>
<b>sla_crit</b>	Secondes	Temps en statut <b>Critique</b>
<b>sla_warn</b>	Secondes	Temps en statut <b>Attention</b>

<b>sla_thresholds</b>	Liste de pourcentages	Deux pourcentages : <ul style="list-style-type: none"> <li>la première valeur est le seuil d'avertissement</li> <li>la deuxième valeur est le seuil de critique</li> </ul> Les pourcentages ont une précision à 3 chiffres ( <i>ex: 90.001</i> )
<b>sla_date</b>	Chaîne de caractères	au format <b>jj_mm_aaaa</b> ( <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
<b>period=start:date~end:date</b>	La date de la veille	Définit la période où collecter les données SLA <ul style="list-style-type: none"> <li>Les dates sont au format <b>jj_mm_aaaa</b> ( <i>ex: 12_05_2021</i> )</li> <li>Si <b>start</b> n'est <b>pas précisé</b>, cela signifie que le début de la période est la date de la veille.</li> <li>Si <b>end</b> n'est <b>pas précisé</b>, la fin de la période est égal à la date de début.</li> </ul>

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 supérieur à la date de la veille.
- La date de fin ne peut pas être supérieur à la date de début.



Si une de ces trois règles n'est pas respectée, un de ces trois messages d'erreur apparaîtra indiquant à l'utilisateur son erreur :

- Cette erreur indique que l'utilisateur a saisi une date de début inférieure à la date de la plus ancienne donnée SLA enregistrer dans la base MongoDB

```
The start period is not valid, as there is no SLA data for this date. You can filter elements from 13_02_2021.
```

- Cette erreur indique que l'utilisateur a saisi une date de fin supérieur à la date de la veille

```
The end period is invalid, as the requested period is in the future. You can filter elements until 24_05_2021.
```

- Cette erreur indique que l'utilisateur a saisi une date de fin inférieur à la date de début saisie

```
The end period is invalid, as it's less than the start period.
```

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



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=start: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=start: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 :

- **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
<b>nb_sla_in_page</b>	<i>Entier</i>	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 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 checks\_attached\_to\_father :**

- elements\_found
  - hosts
    - father\_name
    - father\_uuid
    - checks
      - check\_name
      - check\_uuid
      - sla
        - sla\_missing
        - sla\_ok
        - sla\_unknown
        - sla\_critical
        - 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



```
curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:24_05_2021~end:25_05_2021" \  
-d "output_format=checks_attached_to_father" \  
-d "filter01=type:check" \  
-d "page_settings=page:0~size:2" \  
http://broker-module-livedata:50100/api/v1/sla
```

```
{  
  "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": "24_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": "25_05_2021",  
          "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  
        }  
      ]  
    }  
  ]  
  },  
  "hosts": [{  
    "father_uuid": "f87c2e56b94b11ebaf7e080027c44e8f",  
    "father_name": "Host 01",  
    "checks": [{  
      "check_name": "Check 01",  
      "check_uuid": "f87c2e56b94b11ebaf7e080027c44e8f-fdd0c038b94b11ebb21f080027c44e8f",  
    }  
  ]  
}]
```

```

        "sla": [{
            "sla_date": "24_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": "25_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\_on\_same\_level

```

curl -s -S -H "x-api-token: XYZ" \
-d "period=start:24_05_2021~end:25_05_2021" \
-d "output_format=elements_on_same_level" \
-d "filter01=type:check" \
-d "page_settings=page:0~size:2" \
http://broker-module-livedata:50100/api/v1/sla

```

```

{
  "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": 2,
    "nb_hosts_in_page": 0,
    "nb_clusters_in_page": 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": "24_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": "25_05_2021",
      "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": "24_05_2021",
    "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": "25_05_2021",
    "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:24_05_2021~end:25_05_2021" \
-d "output_format=list_of_sla" \
-d "filter01=type:check" \
-d "page_settings=page:0~size:2" \
http://broker-module-livedata:50100/api/v1/sla

```

```
{
  "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": "24_05_2021",
    "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": "25_05_2021",
    "father_name": "Cluster 01",
    "sla_unknown": 24183,
    "sla_missing": 3916,
    "check_uuid": "12760f56bc6d11eb85a3080027c44e8f-9d86c522bd3511ebb58c080027c44e8f",
    "sla_warn": 0
  }
]
}
```