

# Module livedata-module-sla-provider

## Sommaire

### Concept

#### Activation du module

Exemple d'activation du module nommé "livedata-module-sla-provider" sur le module nommé "broker-module-livedata" ( configuration livrée par défaut par Shinken )

Créer un nouveau module de type livedata\_module\_sla\_provider

#### Configuration

Exemple de fichier de configuration

Détails des sections composant le fichier de configuration

Identification du module

Accès à la base MongoDB

Configuration de l'URI de connexion et de l'authentification par mot de passe

Connexion directe au serveur MongoDB

Connexion par SSH au serveur MongoDB

Gestion de la reconnexion automatique avec un cluster MongoDB

Paramétrage des états Données manquantes ( Missing data ) et Shinken inactif ( Shinken inactive )

## Concept

Le module de type "livedata\_module\_sla\_provider" est un module qui permet au module de type "broker\_module\_livedata" du Broker de fournir une API de consultation des données SLA archivées.

- `/api/v2/sla`
- Il récupère les données SLAs présentes dans la base de données Shinken.
  - Les données SLA sont calculées à la fin de la journée, donc **la dernière donnée disponible est celle de la veille**.

Pour plus de détails sur cette API ( voir la page [V2 - \( READ \) /api/v2/sla -- OPTIONNEL --](#) )

## Activation du module

Les modules de type "livedata\_module\_sla\_provider" sont des modules qui doivent être activés sur un module de type "broker\_module\_livedata", qu'on appellera le **module parent**.

- L'activation du module s'effectue en ajoutant le **nom** du module dans la configuration du **module parent**.
  - Pour cela, il faut ouvrir le fichier de configuration du **module parent** ( de type "broker\_module\_livedata" ), et ajouter dans le paramètre **modules**, le nom du module de type "livedata\_module\_sla\_provider".
- Il est possible de faire plusieurs modules de type "livedata\_module\_sla\_provider".
  - Cela permet, **par exemple**, d'avoir des configurations différentes en fonction des royaumes.
- S'il y a plusieurs modules de type "broker\_module\_livedata" présents dans l'architecture, il ne faut pas oublier d'activer le module de type "livedata\_module\_sla\_provider" dans la configuration de chacun d'eux.
- **Contraintes :**
  - Activable uniquement sur un module de type "broker\_module\_livedata" ( voir la page [Configuration du broker-module-livedata](#) ).
  - Il ne peut y avoir qu'un seul module de type "livedata\_module\_sla\_provider" sur un module de type "broker\_module\_livedata".

Pour prendre en compte le changement de configuration, il faut redémarrer l'Arbiter :

```
service-shinken-arbiter restart
```

## Exemple d'activation du module nommé "livedata-module-sla-provider" sur le module nommé "broker-module-livedata" ( configuration livrée par défaut par Shinken )

L'exemple suivant :

- active le module "livedata-module-sla-provider" ,
- sur le module "broker-module-livedata" , dont la configuration est dans le fichier `/etc/shinken/modules/broker-module-livedata.cfg`.

Modification dans le fichier du module `/etc/shinken/modules/broker-module-livedata.cfg` :

```
define module {
    [...]
    modules          Module 1, Module 2, Module 3, livedata-module-sla-provider
    [...]
}
```

Puis redémarrage de l'Arbiter

```
service-shinken-arbiter restart
```

## Créer un nouveau module de type `livedata_module_sla_provider`

Pour pouvoir configurer un module de type "livedata\_module\_sla\_provider", il faut créer un nouveau fichier de configuration grâce au fichier d'exemple fourni par défaut.

- Pour commencer, il faut choisir le nom du nouveau module.
  - Pour l'exemple, on l'appelle "Mon-Module-Livedata-Sla-Provider".
  - Remplacer dans l'exemple le mot "Mon-Module-Livedata-Sla-Provider" par le nom qui a été choisi.
- Puis il faut créer le fichier de configuration :
  - Copier le fichier de définition du module d'exemple : **`/etc/shinken-user-example/configuration/daemons/brokers/modules/broker-module-livedata/modules/livedata-module-sla-provider/livedata-module-sla-provider-example.cfg`** dans le répertoire de définition des modules **`/etc/shinken/modules/`**.  
( Exemple : `/etc/shinken/modules/livedata-module-sla-provider__Mon-Module-Livedata-Sla-Provider.cfg` )

```
cp /etc/shinken-user-example/configuration/daemons/brokers/modules/broker-module-livedata/modules/livedata-module-sla-provider/livedata-module-sla-provider-example.cfg /etc/shinken/modules/livedata-module-sla-provider__Mon-Module-Livedata-Sla-Provider.cfg
```

- Ensuite, il faut modifier le fichier nouvellement créé pour configurer le nouveau module.
  - Il faut vérifier que le fichier appartienne à l'utilisateur shinken et qu'il possède le droit d'édition. Si ce n'est pas le cas, il faut effectuer les commandes suivantes :

```
chown shinken:shinken /etc/shinken/modules/livedata-module-sla-provider__Mon-Module-Livedata-Sla-Provider.cfg
chmod u+w /etc/shinken/modules/livedata-module-sla-provider__Mon-Module-Livedata-Sla-Provider.cfg
```

- On change le nom du module en "Mon-Module-Livedata-Sla-Provider" dans le fichier **`/etc/shinken/modules/livedata-module-sla-provider__Mon-Module-Livedata-Sla-Provider.cfg`**

```
...
    # Module name [ Must be unique ]
    [ MANDATORY ]
    #

    module_name          Mon-Module-Livedata-Sla-Provider
    ...
```

- Ensuite, il faut ajouter le nouveau module dans le module de type "broker\_module\_livedata" correspondant.
  - Dans notre exemple, on ajoute le module "Mon-Module-Livedata-Sla-Provider" au module "broker-module-livedata" défini dans le fichier **`/etc/shinken/modules/broker-module-livedata.cfg`**

```

define module {
    [...]
    modules
    Mon-Module-Livedata-Sla-Provider
    [...]
}
Module 1, Module 2, Module 3,

```

- Puis pour finir, il faut redémarrer l'Arbiter pour que les modifications apportées à la configuration soient appliquées.

```
service-shinken-arbiter restart
```

## Configuration

Par défaut, La configuration du module se trouve dans le fichier `/etc/shinken/modules/livedata-module-sla-provider.cfg`

- Un exemple de configuration est également disponible dans `/etc/shinken-user-example/configuration/daemons/brokers/modules/broker-module-livedata/modules/livedata-module-sla-provider/livedata-module-sla-provider-example.cfg`

### Exemple de fichier de configuration

```

# CFG_FORMAT_VERSION 1 ( SHINKEN : DON'T TOUCH THIS LINE )

#####
# livedata_module_sla_provider
#####
# Modules that can load this module:
# - broker-module-livedata (to add new route to this module)
# This module is an API providing information on SLA of monitored elements
#####

define module {

    # #
    #     MODULE IDENTITY     #
    # #

    # Module name [ Must be unique ]                               [ MANDATORY ]
    #
    module_name                                     livedata-module-sla-provider

    # Module type [ Do not edit ]                                   [ MANDATORY ]
    #
    module_type                                     livedata_module_sla_provider

    # #
    #     DATABASE CONNECTION     #
    # #

    # MongoDB parameters     #

    # MongoDB uri definition . You can find the mongodb uri syntax at
    # https://docs.mongodb.com/manual/reference/connection-string/
    #
    #     Default : mongodb://localhost/?w=1&fsync=false
    #
    # broker_module_livedata_module_sla_provider_database_uri mongodb://localhost/?w=1&fsync=false

    # Which database contains sla data
    #
    #     Default : shinken
    #
    # broker_module_livedata_module_sla_provider_database_name shinken

    # username/password to authenticate to MongoDB.

```

```

# Both parameters must be provided for authentication to function correctly.
#
# broker__module_livedata__module_sla_provider__database__username
#
# broker__module_livedata__module_sla_provider__database__password
#
# SSH tunnel activation to secure your mongodb connection
# That will allow all mongodb to be encrypted & authenticated with SSH
#
#     Default : 0 => Disable ( disable ssh tunnel )
#     ...     : 1 => Enable  ( enable ssh tunnel )
#
# broker__module_livedata__module_sla_provider__use_ssh_tunnel 0
#
# SSH user to connect to the mongodb server.
#
#     Default : shinken
#
# broker__module_livedata__module_sla_provider__ssh_user shinken
#
# SSH keyfile to connect to the mongodb server.
#
#     Default : ~shinken/.ssh/id_rsa
#
# broker__module_livedata__module_sla_provider__ssh_keyfile ~shinken/.ssh/id_rsa
#
# SSH Timeout used to test if the SSH tunnel is viable or not, in seconds.
#
#     Default : 10 ( seconds )
#
# broker__module_livedata__module_sla_provider__ssh_tunnel_timeout 10
#
#   AutoReconnect Management   #
#
# When MongoDB require you to reconnect ( For example, It can occur when a new PRIMARY is elected
# in a MongoDB cluster ), it will raised the MongoDB AutoReconnect exception.
#
# How many try to reconnect before module go in error
#
#     Default : 4 ( number of try )
#
#
broker__module_livedata__module_sla_provider__database__retry_connection_X_times_before_considering_an_error
4
#
# Time between each try
#
#     Default : 3 ( seconds )
#
# broker__module_livedata__module_sla_provider__database__wait_X_seconds_before_reconnect 3
#
# NOTE: Change these values only if you have a MongoDB cluster and you change the
# heartbeatTimeoutSecs of your MongoDB replica set
# The value of
# retry_connection_X_times_before_considering_an_error * wait_X_seconds_before_reconnect
# must be higher than heartbeatTimeoutSecs in the rs.conf(); of your MongoDB replica set.
#
# #
#   SLA CALCULATION           #
# #
#
# No_data periods ( "Missing data" and "Shinken inactive" status )
#
#     Default : include => Only status is considered. "Missing data" and "Shinken inactive"
#                 status are counted negatively in the SLA.
#     ...     : exclude => No_data are not counted from SLA considered period.
#     ...     : ok      => No_data are considered as UP periods.
#
# broker__module_livedata__module_sla_provider__no_data_period include

```

```
}
```

## Détails des sections composant le fichier de configuration

### Identification du module

Il est possible de définir plusieurs instances de module de type `livedata-module-sla-provider` dans l'architecture Shinken .

- Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Description
<code>module_name</code>	Texte	---	<code>livedata-module-sla-provider</code>	<b>Il est conseillé de choisir un nom en fonction de l'utilisation du module pour que la configuration soit simple à maintenir.</b>  Doit être unique.
<code>module_type</code>	Texte	---	<code>livedata_module_sla_provider</code>	Ne doit pas être modifié.

### Accès à la base MongoDB

Cette configuration s'effectue dans le fichier de configuration du module.

Pour se connecter à la base MongoDB utilisée pour le stockage des données, 2 méthodes sont disponibles :

- **Connexion directe** : Par défaut, mais non sécurisée.
- **Tunnel SSH** : Shinken se connecte à la base MongoDB au travers d'un module SSH pour plus de sécurité

### Configuration de l'URI de connexion et de l'authentification par mot de passe

```
# MongoDB parameters #  
  
# MongoDB uri definition . You can find the mongodb uri syntax at  
# https://docs.mongodb.com/manual/reference/connection-string/  
#  
# Default : mongodb://localhost/?w=1&fsync=false  
#  
# broker_module_livedata_module_sla_provider_database_uri mongodb://localhost/?w=1&fsync=false  
  
# Which database contains sla data  
#  
# Default : shinken  
#  
# broker_module_livedata_module_sla_provider_database_name shinken  
  
# username/password to authenticate to MongoDB.  
# Both parameters must be provided for authentication to function correctly.  
#  
# broker_module_livedata_module_sla_provider_database_username  
  
#  
# broker_module_livedata_module_sla_provider_database_password
```

Nom	Type	Unité	Défaut	Description
-----	------	-------	--------	-------------

<code>broker__module_livedata__module__sla_provider__database_uri</code>	Texte	URL	<b>mongodb://local host/?w=1&amp;fsync=false</b>	La description de la syntaxe de l'URI de MongoDB est disponible à l'adresse suivante <a href="https://docs.mongodb.com/manual/reference/connection-string/">https://docs.mongodb.com/manual/reference/connection-string/</a>
<code>broker__module_livedata__module__sla_provider__database_name</code>	Texte	---	<b>shinken</b>	Nom de la base de données où sont stockées les données SLA.
<code>broker__module_livedata__module__sla_provider__database_username</code>	Texte	---		Utilisateur pour l'authentification avec mot de passe à la base MongoDB. Utile uniquement si l'activation par mot de passe a été activé ( voir la page <a href="#">MongoDB - activation de l'authentification par mot de passe</a> )
<code>broker__module_livedata__module__sla_provider__database_password</code>	Texte	---		Mot de passe de l'utilisateur utilisé pour l'authentification avec mot de passe à la base MongoDB. Utile uniquement si l'activation par mot de passe a été activé ( voir la page <a href="#">MongoDB - activation de l'authentification par mot de passe</a> )

### Connexion directe au serveur MongoDB

Par défaut, le module se connecte de manière directe à la base MongoDB, définie avec les paramètres communs listés ci-dessus, car le paramètre "use\_ssh\_tunnel" est à 0.

### Connexion par SSH au serveur MongoDB

Par défaut, le module se connecte de manière directe à la base MongoDB pour y lire et écrire les données.

Dans la configuration du module, ceci correspond au paramètre "use\_ssh\_tunnel" à 0.

C'est la méthode de connexion par défaut lorsque la base est sur la même machine que le démon ( quand l'URL de la base est localhost ).

Si la base est sur une autre machine, il faudra alors se connecter à la base via un tunnel SSH. Cela permet à la base distance de rester en écoute réseau sur l'interface réseau local, ce qui la sécurise des accès extérieurs ( voir la page [Sécurisation des connexions aux bases MongoDB](#) ).

```
# SSH tunnel activation to secure your mongodb
connection
# That will allow all mongodb to be encrypted & authenticated with SSH
#
#     Default : 0 => Disable ( disable ssh tunnel )
#     ...     : 1 => Enable  ( enable ssh tunnel )
#
# broker__module_livedata__module_sla_provider__use_ssh_tunnel      0

# SSH user to connect to the mongodb server.
#
#     Default : shinken
#
# broker__module_livedata__module_sla_provider__ssh_user           shinken

# SSH keyfile to connect to the mongodb server.
#
#     Default : ~shinken/.ssh/id_rsa
#
# broker__module_livedata__module_sla_provider__ssh_keyfile        ~shinken/.ssh/id_rsa

# SSH Timeout used to test if the SSH tunnel is viable or not, in seconds.
#
#     Default : 10 ( seconds )
#
# broker__module_livedata__module_sla_provider__ssh_tunnel_timeout 10
```

Nom	Type	Unité	Défaut	Description
-----	------	-------	--------	-------------

<code>broker__module_livedata__module_sla_provider__use_ssh_tunnel</code>	Booléen	---	0	<ul style="list-style-type: none"> <li>• 1 : Connexion par tunnel SSH</li> <li>• 0 : Connexion directe</li> </ul>
<code>broker__module_livedata__module_sla_provider__ssh_user</code>	Texte	Utilisateur unix	shinken	L'utilisateur avec lequel le tunnel sera établi.
<code>broker__module_livedata__module_sla_provider__ssh_keyfile</code>	Texte	Chemin de fichier	~shinken/.ssh/id_rsa	La clé SSH privée présente sur le serveur Shinken qui sera utilisé pour établir le tunnel.
<code>broker__module_livedata__module_sla_provider__ssh_tunnel_timeout</code>	Entier	Seconde	10	Spécifie le timeout en secondes de la vérification du tunnel SSH avant que la connexion vers MongoDB soit effectuée.

## Gestion de la reconnexion automatique avec un cluster MongoDB

```

#   AutoReconnect Management

#   When MongoDB require you to reconnect ( For example, It can occur when a new PRIMARY is elected
#   in a MongoDB cluster ), it will raised the MongoDB AutoReconnect exception.

#   How many try to reconnect before module go in error
#
#       Default : 4 ( number of try )
#
#
broker__module_livedata__module_sla_provider__database__retry_connection_X_times_before_considering_an_error
4

#   Time between each try
#
#       Default : 3 ( seconds )
#
#   broker__module_livedata__module_sla_provider__database__wait_X_seconds_before_reconnect 3

#   NOTE: Change these values only if you have a MongoDB cluster and you change the
#   heartbeatTimeoutSecs of your MongoDB replica set
#   The value of
#   retry_connection_X_times_before_considering_an_error * wait_X_seconds_before_reconnect
#   must be higher than heartbeatTimeoutSecs in the rs.conf(); of your MongoDB replica set.

```



### Définitions

- **Primaire**: nom de MongoDB pour désigner un serveur maître, le serveur sur lequel il est possible de faire des requêtes d'écriture dans la base.
- **Election** : processus de MongoDB pour choisir un nouveau membre Primaire si le membre Primaire devient inaccessible

( voir la page [Haute disponibilité de la base MongoDB \(mise en place d'un cluster\)](#) )

Dans le cas de l'utilisation d'un cluster MongoDB, lorsque le membre Primaire devient inaccessible, une nouvelle élection est déclenchée, ce qui provoque une coupure temporaire de l'accès à la base.

Dans le but de ne pas interrompre le service, le module "event-manager-reader" va se reconnecter automatiquement au cluster MongoDB. Pour ce faire, il va faire un nombre d'essais égaux au paramètre "auto\_reconnect\_max\_try" avec une pause de X secondes entre chaque essai ( correspondant au paramètre "auto\_reconnect\_sleep\_between\_try" ).

Par défaut pour MongoDB le temps maximum avant qu'un membre Primaire soit considéré comme indisponible et qu'une nouvelle élection ait lieu est de 10 secondes.

Voir : " heartbeatTimeoutSecs" donné par la commande `rs.conf()`; dans un shell de MongoDB.

Nom	Type	Unité	Défaut	Description
broker__module_livedata__module_sla_provider__database__retry_connectio n_X_times_before_considering_an_error	Entier	---	4	Nombre d'essais de reconnexion à la base.
broker__module_livedata__module_sla_provider__database__wait_X_seconds_ before_reconnect	Entier	seconde	3	Temps entre chaque essai en seconde.

Les valeurs par défauts du fichier laissent 12 secondes, ce qui est amplement suffisant avec la configuration par défaut de MongoDB.



Il est conseillé de ne pas modifier ces valeurs.

### Paramétrage des états Données manquantes ( Missing data ) et Shinken inactif ( Shinken inactive )

```
# #
# SLA CALCULATION #
# #

# No_data periods ( "Missing data" and "Shinken inactive" status )
#
# Default : include => Only status is considered. "Missing data" and "Shinken inactive"
# status are counted negatively in the SLA.
# ... : exclude => No_data are not counted from SLA considered period.
# ... : ok => No_data are considered as UP periods.
#
# broker__module_livedata__module_sla_provider__no_data_period include
```

Les états Données manquantes ( *Missing data* ) et Shinken inactif ( *Shinken inactive* ) ont été regroupés dans un paramètre. Ce paramètre correspond à la période durant laquelle Shinken n'a pas effectué les vérifications pour un check ( *plateforme Shinken éteinte, ou vérification du check désactivé grâce aux Périodes de temps, voir la page Périodes de temps* ). Le statut de ces checks est donc Données manquantes ( *Missing data* ) ou Shinken inactif ( *Shinken inactive* ).

Nom	Type	Unité	Défaut	Description
broker__module_livedata__module_sla_prov ider__no_data_period	Texte	---	include	Choix possible : <ul style="list-style-type: none"> <li><b>include</b> : On considère que si l'on ne peut pas savoir si le service est rendu, c'est qu'il n'est pas rendu.</li> <li><b>exclude</b> : On considère que l'état de la supervision ( <i>de Shinken</i> ) n'impacte pas le taux de disponibilité.</li> <li><b>ok</b> : On considère que même si la supervision ne l'a pas confirmé, le service est rendu.</li> </ul>