

# Affichage et comportement du formulaire - Édition JSON - Météo

## Sommaire

[Qu'est ce que le format JSON](#)  
[Repérer un élément dans la configuration JSON](#)  
[Affichage des définitions d'éléments dans l'éditeur](#)  
    [Indentation](#)  
[Actions possibles dans la zone d'édition](#)  
    [Copier / Coller](#)  
[Affichages des problèmes de configuration](#)  
    [Coloration syntaxique](#)  
    [Bulle d'information](#)  
    [Localiser les problèmes](#)

## Démarrage du démon

Au démarrage et tous les jour à minuit, ce log indique la version ainsi que le numéro de patch cumulatif du démon.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller ] Daemon version is: 02.07.06-release.fr culmulative-patch-11
```

Au démarrage le démon affiche plusieurs logs contenant ses informations dont :

- ses limites systèmes en nombre de fichiers ouvrables, et nombre de threads/processus

```
[YYYY-MM-DD HH:MM:SS] INFO : [daemon-master] [ SYSTEM ] System resource number of open files is set to (soft:1024 / hard:1024 ) (from parameter max_file_descriptor_limit)  
[YYYY-MM-DD HH:MM:SS] INFO : [daemon-master] [ SYSTEM ] System resource number of processes /threads is set to (soft:unlimited / hard:unlimited ) (set at system max values)
```

Les fichiers de log du Poller sont situés dans le dossier **/var/log/shinken/**. Pour plus d'informations, consultez la page [Fichiers Logs](#).

## Gestion de la configuration

### Premier chargement de la configuration

Lorsque le Poller reçoit sa configuration pour la première fois deux logs **INFO** sont affichés.

- Le premier indiquant que nous rentrons dans la phase de chargement d'une nouvelle configuration.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] ----- Loading the new configuration from the arbiter
```

- Le deuxième indiquant que nous avons reçu la configuration de l'Arbiter.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] The arbiter send us a new configuration: [configuration_uuid=configuration-uuid, arbiter=arbiter-name, architecture=architecture-name, date=YYYY-MM-DD HH:MM:SS]
```

### Mise à jour de la configuration

Lorsque qu'il y a une mise à jour de la configuration, deux logs en **INFO** sont affichés.

- Le premier indiquant que nous rentrons dans la phase de chargement d'une nouvelle configuration.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] [ UPDATE ] ----- Loading a configuration update from the arbiter
```

- Le deuxième indiquant que nous avons reçu la configuration de l'Arbiter.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] [ UPDATE ] The arbiter
send us a new configuration: [configuration_uuid=configuration-uuid, arbiter=arbiter-name,
architecture=architecture-name, date=YYYY-MM-DD HH:MM:SS]
```

- ✘ Dans le cas où le poller n'est pas de la même version que l'arbiter master:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ poller-name ] Incompatible daemon version : Your Arbiter daemon is
in version [02.07.06-release.fr culmulative-patch-01] while this daemon is in version [02.07.06-
release.fr culmulative-patch-02]. Refusing this configuration.
```

## Mise à jour des liens vers d'autres démons

Lorsque que l'Arbiter détecte un changement de lien entre les démons quatre logs en **INFO** seront affichés.

- Les deux premiers logs affichent le(les) lien(s) du(des) démon(s) supprimé(s).

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] The arbiter asked us to remove daemons:
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] - REMOVED scheduler : [name=scheduler1-name]
[shard_id= XXX] [uri=http://scheduler_address:port/]
```

- Les deux premiers logs affichent le(les) lien(s) du(des) démon(s) ajouté(s).

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] The arbiter send us new daemons:
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] + ADDED scheduler : [name=scheduler2-name]
[shard_id= XXX] [uri=http://scheduler_address:port/]
```

## Récupération des checks

### Pour récupérer les checks à exécuter

- **Poller actif**
  - Le Poller va demander au Scheduler.
  - Le Poller indique un temps de travail qu'il a de disponible ( en temps cpu ).
  - Le Scheduler lui donne des checks pour le temps disponible ( suivant le temps d'exécution moyen pour les checks constatés sur ce poller ). Il lui donne pour un temps inférieur ou égal au temps disponible.
  - Un log permet d'avoir le nombre de checks récupérés. Ce log s'affiche même si aucun check n'a été récupéré :

```
[YYYY-MM-DD HH:MM:SS] INFO : [daemon-master] [ CHECKS ] [ scheduler-master ] [ GET ] Requesting checks
todo from this scheduler for 2.000s cpu time [received=3 check(s) for 0.039s cpu time]
```

- **Poller passif**
  - Le Scheduler demande au Poller le temps CPU disponible.
  - Le Scheduler lui envoie des checks à traiter pour le temps disponible ( suivant le temps d'exécution moyen pour les checks constatés sur ce Poller ). Il lui donne pour un temps inférieur ou égal au temps disponible.
  - Si des checks sont reçus, un log permet d'avoir le nombre de checks récupérés selon le temps de travail disponible sur le Poller :

```
[YYYY-MM-DD HH:MM:SS] INFO : [daemon-master] [ CHECKS ] [ scheduler-master ] [ RECEIVED ] We received
checks todo from this scheduler for 2.000s cpu time [received=1 check(s) for 0.317s cpu time]
```

## Envoi des résultats de checks au Scheduler

- **Poller actif**
  - Une fois les commandes exécutées, le Poller envoie les résultats au Scheduler.
  - Un log permet d'avoir le nombre de résultats de checks envoyés au Scheduler et le temps mis pour être envoyé.

```
[YYYY-MM-DD HH:MM:SS] INFO : [daemon-master] [ CHECKS RESULTS ] [scheduler-master] [ PUSHED ] 1 check's result(s) sends to this scheduler in [0.043]s
```

- **Poller passif**

- Une fois les checks exécutés, le Poller stocke les résultats en attendant que le Scheduler vienne les récupérer.
- A chaque tour de boucle du Scheduler, ce dernier demande au Poller s'il a des résultats de checks disponibles.
- Si des résultats sont disponibles, un log permet d'avoir le nombre de résultats checks donnés au Scheduler.

```
[YYYY-MM-DD HH:MM:SS] INFO : [daemon-master] [ CHECKS RESULTS ] [scheduler-master] [ GIVEN ] 1 check's result(s) given to answer scheduler request
```

## Surcharge serveur en activité disque, ralentissant l'écriture des logs

Si le serveur hébergeant le démon est surchargé en termes d'IO disques sur le volume qui héberge le fichier de log, alors ce dernier va mettre du temps à s'écrire et va ralentir tout le démon. Il faut alors si c'est faisable isoler le volume des disques sur un disque moins chargé pour ne pas ralentir le démon.

En cas de soucis vous aurez dans les logs l'entrée suivante:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
-----
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ] [ WRITING ] The log writes time is very high (1.87s). Please look at your log disk performance.
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
-----
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
```

## Calcul des statistiques nécessaire au check Poller - \$KEY\$ - Performance du pack Shinken

Quand un check de supervision du démon est fait, on va avoir plusieurs entrées dans les logs qui concernent des données que le démon garde sur diverses statistiques.

Un log permet d'avoir le temps pris sur le calcul des dernières commandes en timeout:

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [daemon-master] [ STATS ] Compute "Checks in timeouts" stats : 0.000s in a total of 2048 commands in timeouts
```

Un log permet d'avoir le temps de calcul concernant les ranges d'exécution des checks/notifications en fonction du temps (<100ms, <400ms, etc):

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [daemon-master] [ STATS ] Compute "Checks per CPU running time" : 0.000s (on a total of 2048 checks)
```

Un log permet d'avoir le temps de calcul pour avoir les 5 commandes les plus longues en temps CPU:

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [daemon-master] [ STATS ] top5 execution time 0.003s (loop over 1 ranges and 343 elements)
```

Un dernier log permet d'avoir le temps complet du calcul des statistiques du démon:

- sur une partie commune à tous les démons (version, chemins, etc)
- sur la partie qui concerne uniquement ce qui concerne ce démon, donc sont inclus ici les temps précédents
- il est affiché en :

- **DEBUG**: si le temps de calcul est inférieur à la valeur du paramètre `display_statistics_compute_time_if_higher` du démon.
- **INFO**: si le temps de calcul est supérieur ou égal au paramètre `display_statistics_compute_time_if_higher` du démon.

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [daemon-master] [ STATS          ] Daemon stats were computed in 0.020s (0.001
for daemon common part, 0.020 for poller
part)
```

En cas d'affichage INFO on met un petit morceau en plus sur comment gérer le niveau de log:

```
[YYYY-MM-DD HH:MM:SS] INFO : [daemon-master] [ STATS          ] Daemon stats were computed in 0.004s (0.000
for daemon common part, 0.004 for poller part) (NOTE: log is displayed in INFO because 0.004 is higher than
display_statistics_compute_time_if_higher=1ms in the daemon cfg)
```

## Échanges entre le processus principal et les Workers

### Problème temporaire de réception de résultats depuis les workers

Il se peut que si un worker est chargé, il mette trop de temps à nous renvoyer un résultat, dans ce cas on aura le log WARNING suivant:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [daemon-master] The worker 2 reception did fail this turn (IOError), skip to
the next turn to receive more.
```

La vérification du bon fonctionnement des workers est faite lors de l'envoi des checks, donc en cas de problème worker mort il sera détecté.

- Si ces logs WARNING sont sporadiques, ils ne sont pas un problème, les résultats sont juste récupérés à la seconde suivante
- Si ces logs WARNING sont constants, ceci signifie que le worker est surchargé, et alors le mécanisme de surcharge doit être activé également