

Onglet Graphiques

Import des modules depuis /var/lib/shinken/modules - chapitre [CODE-LOADING]

Les démons vont démarrer leurs modules manager qui vont lister les répertoires dans **/var/lib/shinken/modules** et vont tenter de charger chaque module.

Chargement du code du module

Le "modules manager" va regarder si le fichier "module_info.json" placé dans le répertoire du module existe. Si il n'existe pas un log en **WARNING** sera affiché.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=architecture-export ] Missing 'module_info.json' file at the path /var/lib/shinken/modules
/architecture-export/module_info.json. The code of the module gonna be charged.
```

Après que le "modules manager" est regarder si le fichier "module_info.json" existe, il va regardé si la clé "daemons" est bien présente dans le fichier. Si elle n'existe pas un log en **WARNING** sera affiché.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=architecture-export ] Missing 'daemons' key in the 'module_info.json' file at the path /var/lib
/shinken/modules/architecture-export/module_info.json. The code of the module gonna be charged.
```

Si le "modules manager" n'a pas réussi à lire le fichier à cause d'un problème de formatage, un log en **WARNING** sera affiché.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=architecture-export ] The 'module_info.json' file at the path /var/lib/shinken/modules
/architecture-export/module_info.json is malformed. The code of the module gonna be charged.
```

Si le "modules manager" n'a pas réussi à lire le fichier à cause d'un problème de permissions, un log en **WARNING** sera affiché.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=architecture-export ] Permission denied to read file 'module_info.json' file at the path /var/lib
/shinken/modules/architecture-export/module_info.json. The code of the module gonna be charged.
```

Le "modules manager" annonce qu'il va commencer à charger son module. Un log en **INFO** apparaîtra.

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=architecture-export ] Starting to load the module code directory /var/lib/shinken/modules
/architecture-export.
```

Si le "modules manager" trouve une exception Python durant le procédé d'import du module, un log en **ERROR** avec l'exception sera affiché et le module ne sera pas importé.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=architecture-export ] Import module [architecture-export] failed: EXCEPTION.
```

Chargement du module en tant que module Python

Une fois que le "modules manager" à charger le code du module il va vérifier si le module peut être importé en tant qu'un module python. Un log en **DE BUG** apparaîtra.

aaaaa

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [ directory=architecture-
export ] Starting to load the module code directory /var/lib/shinken/modules/architecture-export.
```

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [ directory=architecture-export ] Try to import a module code in /var/lib/shinken/modules/architecture-export as python module.
```

Si le "modules manager" ne peut pas importer le code du module en tant que module python alors des logs en **WARNING** formant un bloc apparaîtrons.

- Un premier log expliquant que le "modules manager" n'a pas réussi à importer le module en tant que python module.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [ directory=architecture-export ] Failed to import the directory /var/lib/shinken/modules/architecture-export as a python module.
```

- Un second log s'affichera seulement s'il manque le fichier '__init__.py' dans le dossier du module. Sans ce fichier il n'est pas possible d'importer le module en tant que module Python.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [ directory=architecture-export ] - because of missing file : /var/lib/shinken/modules/architecture-export/__init__.py.
```

- Un troisième log informant qu'il n'est pas possible de faire un import : 'import architecture-export.my_file'. Car le module n'a pas été importé en tant que module python.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [ directory=architecture-export ] - 'import architecture-export.my_file' will not work in your module.py file.
```

Si le "modules manager" a réussi à importer, le module en tant que Python module alors un log en **DEBUG** apparaîtra.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [ directory=architecture-export ] Success to import the directory /var/lib/shinken/modules/architecture-export/ as a python module.
```

Vérification des propriétés du module

Une fois que le modules manager à importer (ou non) son module en tant que python module, il va vérifier si le dictionnaire "properties" existe.

S'il n'existe pas alors un log en **ERROR** sera affiché expliquant que le dictionnaire est manquant et que le module ne sera par conséquent pas importé.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [ directory=architecture-export ] Missing properties dict in your module file /var/lib/shinken/modules/architecture-export/module.py. The module will not be imported.
```

Les démons vont lister les répertoires dans **/var/lib/shinken/modules** et vont tenter de charger chaque module.

Ceci va créer une entrée **DEBUG**:

```
[2020-07-08 16:31:36] DEBUG : [broker-master ] [modules-manager ] [MODULE-CODE-LOADING] [directory=architecture-export ] Starting to load the module code directory /var/lib/shinken/modules/architecture-export.
```

Quand l'import est effectué, on aura une ligne en **DEBUG**:

```
[2020-07-08 16:31:36] DEBUG : [broker-master ] [modules-manager ] [MODULE-CODE-LOADING] [directory=architecture-export ] [0.021s] Module code was loaded. Did import 10 python new librairies (PIL, PIL. Image, PIL. ImageMode, PIL. _binary, _imaging, architecture-export, architecture-export.mapper, shinken.webui. bottlewebui, shinken.webui.bottlewebui.ext, shinken.webui.cherrybackend).
```

Ceci donne le temps de chargement, mais également les nouvelles librairies importées.

À la fin des chargements, il y a un résumé en **INFO** qui est affiché:

```
[2020-07-08 16:31:38] INFO : [broker-master ] [modules-manager ] [MODULE-CODE-LOADING] A total of 9 Shinken Enterprise modules are available for this daemon/module (broker): broker-module-livedata, event-container, graphite-perfdata, livestatus, simple-log, sla, trending, webui, MODULE-CODE-NAME (on a total of 40, loaded in 1.844s)
```

Les modules éligibles à ce démon/module sont listés.

Création des instances de modules

Une fois le code python chargé, on lui demande de créer des instances pour les modules (les vrais objets):

```
[2020-07-08 16:31:38] DEBUG : [broker-master ] [modules-manager ] [MODULE-INSTANCE-CREATION] [module type=webui ] [name=WebUI ] Start to create the module instance
```

Une fois créé, on a un log en **INFO** concernant le temps que ça a pris.

```
[2020-07-08 16:31:38] INFO : [broker-master ] [modules-manager ] [MODULE-INSTANCE-CREATION] [module type=webui ] [name=WebUI ] [0.046s] SUCCESS The module instance is created.
```

Initialisation des instances de modules

Une fois les instances créées, on leur demande de se préparer (ouvrir leurs connexions, lancer les processus pour les modules externes ou avec des workers, etc).

Ceci sera de l'ordre de la seconde pour le cas des modules avec workers (il faut créer les queues de connexions, lancer les processus et le manager de queue, etc).

```
[2020-07-08 16:31:38] INFO : [broker-master ] [modules-manager ] [MODULE-INSTANCE-CREATION] [WebUI ] Trying to initialize module by calling it's init method.
```

Quand l'initialisation est finie, un log **INFO** est affiché.

```
[2020-07-08 16:31:39] INFO : [broker-master ] [modules-manager ] [MODULE-INSTANCE-CREATION] [WebUI ] [0.801 s] SUCCESS The module was initialized successfully.
```

Rajout d'un nouveau module dans un démon

Lorsqu'un démon reçoit un nouveau module, on va avoir une entrée **MODULES-CHANGE** en **INFO** comme quoi le module est **started**:

```
[2020-07-09 13:29:58] INFO : [broker-master ] [modules-manager ] [MODULES-CHANGE ] Modules did changed:
[2020-07-09 13:29:58] INFO : [broker-master ] [modules-manager ] [MODULES-CHANGE ] - WebUI -> started (new)

[2020-07-09 13:29:58] INFO : [broker-master ] [modules-manager ] [MODULES-CHANGE ] - sla -> started (new)
```

À noter que les sous-modules d'un module seront eux affichés en **DEBUG**:

```
[2020-07-09 13:29:58] INFO : [broker-master ] [modules-manager ] [MODULES-CHANGE ] - WebUI -> started (new)
[2020-07-09 13:29:58] DEBUG : [broker-master ] [modules-manager ] [MODULES-CHANGE ] * Cfg_password
[2020-07-09 13:29:58] DEBUG : [broker-master ] [modules-manager ] [MODULES-CHANGE ] * sla
```

Suppression d'un module dans un démon

Quand un module est enlevé de la configuration d'un démon, celui-ci l'affichera avec une entrée **MODULES-CHANGE** en **INFO** comme quoi le module est **stopped**:

```
[2020-07-09 13:57:31] INFO : [broker-master ] [modules-manager ] [MODULES-CHANGE ] Modules did changed:
[2020-07-09 13:57:31] INFO : [broker-master ] [modules-manager ] [MODULES-CHANGE ] - Livestatus -> stopped
(remove)
```

Changement de configuration d'un module (et sa relance)

Quand la configuration d'un module change (ou la configuration d'un de ses sous-modules) alors le module est redémarré (et ses processus redémarrés si besoin). Ceci est loggué avec une entrée **MODULES-CHANGE** en **INFO** comme quoi le module est **restarted**:

```
[2020-07-09 13:51:49] INFO : [broker-master ] [modules-manager ] [MODULES-CHANGE ] Modules did changed:
[2020-07-09 13:51:49] INFO : [broker-master ] [modules-manager ] [MODULES-CHANGE ] - Livestatus -> restarted
(configuration change)
```