

API REST: Gérer des hôtes

Phases lors du chargement des modules

Le chargement des modules se fait en plusieurs phases:

- import du code des modules depuis le répertoire `/var/lib/shinken/modules`
 - ATTENTION: actuellement TOUS les répertoires/modules sont chargés, ceci implique une consommation mémoire importante pour les démons (30Mo consommés par processus)
- on identifie suivant les `module_type` quels auprès de quels codes on va demander des instances de modules
 - on crée ensuite les instances de module (appel `get_instance()` sur les modules)
- on demande aux instances de s'initialiser (typiquement créer ses connexions ou charger des fichiers), via l'appel `init()` des modules

Import du code des modules depuis `/var/lib/shinken/modules`

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 ] [0.801s] 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 (removed)
```

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)
```