

# Tagger utilisant un module basé sur des expressions régulières

## Sommaire

[Concept](#)

[Définir ou configurer un Tagger](#)

[Exemple : automatiquement assigner au royaume Bordeaux les hôtes/clusters dont le nom commence par bdx](#)

[Exemple 1 : mettre le nom du royaume directement dans la propriété définissant le royaume](#)

[Exemple 2 : plus flexible, rajouter un modèle "bordeaux", prioritaire sur les autres modèles](#)

[Visualisation dans l'interface de configuration](#)

## Concept

Un Tagger basé sur des expressions régulières s'applique sur les éléments suivants issus de l'import des sources :

- les hôtes,
- les clusters,
- les modèles d'hôtes ou
- les modèles de clusters

Il utilise une expression régulière ( *regex* ) sur une propriété ou une donnée pour sélectionner les éléments auxquels appliquer la modification configurée pour le Tagger.

Un cas d'usage classique est de rajouter/modifier des modèles d'hôtes/clusters sur les éléments s'ils respectent une règle de nommage sur le nom.

## Configurer un Tagger

L'activation d'un Tagger se fait en trois étapes :

- Définir un Tagger qui utilise un module de type `sync_ip_tag` ( *ex* : `/etc/shinken/taggers/ip-tags.cfg` ).
- Configurer les règles du module ( *ex* : `/etc/shinken/modules/ip-tag-dmz.cfg` ).
- Déclarer le Tagger dans le Synchronizer ( `/etc/shinken/synchronizers/synchronizer-master.cfg` ).

( voir la page [Module sync-regex-tag](#) ).

**Exemple : automatiquement assigner au royaume Bordeaux les hôtes/clusters dont le nom commence par bdx**

### Exemple 1: mettre la valeur du royaume directement dans la valeur du royaume

Si les éléments dont le nom commence par la chaîne **bdx**, sont situés à Bordeaux, il est pertinent de les assigner automatiquement au royaume **Bordeaux**. Cela permet une supervision plus efficace, par exemple en utilisant le Poller du royaume Bordeaux, autorisé par les pare-feux de ces serveurs.

Le module de type `sync_regex_tag` suivant permet ce comportement :

- **SI** la propriété `host_name` ( *le nom* ) de l'hôte/cluster commence par `bdx`
  - **ALORS**, on va écraser ( *paramètre `method` à la valeur `"set"`* ) la propriété `realm` ( *paramètre `"property"`* ) avec la valeur **Bordeaux** ( *paramètre `"value"`* ).

```
define module{
    module_name      sync-regex-tag-bordeaux-basic
    module_type      sync-regex-tag

    # La regex a appliquer
    matched_regex    ^bdx.*

    # on va appliquer la regex sur le nom de l'hôte
    /cluster
    matched_prop      host_name

    # On va alors écraser la propriété realm avec la
    valeur Bordeaux
    property          realm
    method            set
    value             Bordeaux
}
```



Shinken conseil de passer par des modèles d'hôtes, car ils permettent plus de flexibilité. Il suffit de faire évoluer le modèle dans le temps sans avoir besoin de modifier la configuration de Shinken. ( voir l'exemple suivant )

## Exemple 2: plus flexible, rajouter un modèle "bordeaux", prioritaire sur les autres modèles

Si la méthode précédente fonctionne, elle n'est pas optimale : en effet, changer le royaume par Bordeaux est utile, mais d'autres éditions de propriétés seront peut-être nécessaires dans le futur ( *ajout d'utilisateur à notifier sur l'hôte, dépendances réseaux ...* ). Or la méthode précédente impose de créer un nouveau Tagger avec un nouveau module de type `sync_regex_tag` pour chaque nouvelle édition de propriété.

Il est donc **fortement recommandé** de ne pas modifier les propriétés directement, mais plutôt de **passer par des modèles d'hôtes/cluster**.

Il sera ainsi facile de faire des changements sur toutes les machines situées à Bordeaux, en modifiant juste le modèle Bordeaux.

Le module de type `sync_regex_tag` suivant permet ce comportement :

- SI la propriété `host_name` ( *le nom* ) de l'hôte/cluster commence par `bdx`
  - **ALORS**, on va **rajouter au début** ( *paramètre `method` à la valeur `"prepend"`* ) de la propriété `use` ( *paramètre `property`* ) la valeur **bordeaux** ( *paramètre `value`* ).

```
define module{
    module_name      sync-regex-tag-bordeaux-
    recomanded
    module_type      sync-regex-tag

    # La regexp a appliquer
    matched_regex    ^bdx.*

    # on va appliquer la regexp sur le nom de l'hôte
    /cluster
    matched_prop      host_name

    # On va alors rajouter le template bordeaux au
    debut de use (prioritaire)
    property          use
    method            prepend
    value             bordeaux
}
```

## Visualisation dans l'interface de configuration

Sur l'Interface de Configuration, les noms des Taggers listés, sont des liens cliquables qui redirigent sur leurs configurations.

| Taggers > regex-tags <span>Ok</span> |                                                  |
|--------------------------------------|--------------------------------------------------|
| Configuration                        | Règles                                           |
| Clé                                  | Valeur                                           |
| Ordre                                | 2                                                |
| Modules                              | sync-regex-tag                                   |
| Description                          | This tagger will tag host based on the host_name |

Un onglet permet de visualiser également un résumé des règles qui vont s'appliquer.

| Taggers > regex-tags <span>Ok</span> |                      |           |               |         |
|--------------------------------------|----------------------|-----------|---------------|---------|
| Configuration                        | Règles               |           |               |         |
| Propriété ciblée                     | Expression Régulière | Propriété | Valeur        | Méthode |
| host_name                            | .*shinkendemo.*      | use       | ADD BY TAGGER | append  |