3.18.1 Gestion des caches (fonctionnement V1)

⚠ Cette documentation décrit le mode de fonctionnement compatible avec ESUP-Commons V1. Nous recommandons d'utiliser les annotations.

Si la gestion du cache est en général anecdotique (peu importante en terme de développement), elle intervient en général à plusieurs endroits dans les applications et son intégration dans *esup-commons* permet :

- D'uniformiser les mécanismes utilisés,
- De centraliser la configuration des caches.

esup-commons s'appuie sur la bibliothèque *Ehcache*. Elle est utilisée en particulier pour cacher les requêtes à l'annuaire *LDAP*, ainsi qu'à cacher les exceptions remontées par courrier électronique.
L'utilisation de *EhCache* à toute autre fin est très simple, comme nous le montrons ci-après.

Sommaire :

- Configuration
- Utilisation

Configuration

Un *bean* gestionnaire de caches nommé **cacheManager** est déclaré dans le fichier de configuration **/properties/cache/cache.xml** :

```
<bean
    id="cacheManager"
    class="org.springframework.cache.ehcache.EhCacheManagerFactoryBean"
    >
  <property
      name="configLocation"
      value="classpath:/properties/cache/ehcache.xml" />
</bean>
```

Comme on le voit, la configuration de ce gestionnaire de caches est localisée dans le fichier de propriétés **/properties/cache/ehcache.xml**, il est au format *EhCache*.
Le lecteur se reportera à la documentation complète de *EhCache* pour peaufiner la configuration des caches. Elle est simple et consiste à définir des paramètres pour chaque zone de cache. On trouvera par exemple pour le cache des requêtes *LDAP* :

```
<cache
  name="org.esupportail.commons.services.ldap.CachingLdapServiceImpl"
  maxElementsInMemory="1000"
  eternal="false"
  timeToIdleSeconds="300"
  timeToLiveSeconds="600"
  overflowToDisk="true"
/>
```

Utilisation

L'utilisation dans le code *Java* est très simple, il faut seulement s'assurer que tout objet stocké en cache implémente l'interface **Serializable**.

Lorsque l'on veut cacher un objet, il faut un cache, que l'on récupère de cette manière à partir d'un gestionnaire de cache :

```
String cacheName = getClass().getName();
if (!cacheManager.cacheExists(cacheName)) {
  cacheManager.addCache(cacheName);
}
cache = cacheManager.getCache(cacheName);
```

Nom du cache

Le nom donné à la récupération du cache correspond à l'attribut **name** de la configuration de *EhCache*.

Il faut ensuite définir comment associer une clé unique à tout objet que l'on veut stocker dans le cache. On prendra souvent une chaîne de caractère, mais pas toujours.
Par exemple, si l'on veut stocker le résultat d'une requête *HTTP*, on prendra comme clé l'*URL* de la requête. Un court morceau de code illustrant comment on peut cacher des requêtes est le suivant :

```
String getUrlContent(String url) {
  Element element = cache.get(url);
  if (element != null) {
    return (String) element.getObjectValue();
  }
  String content = // do the request here
  cache.put(new Element(url, content));
  return content;
}
```