

Bugs connus

CAS 6.3.x à 6.6.x : petit memory leak avec l'embedded tomcat

Les sessions tomcat s'accumulent, nécessitant un restart régulier pour éviter un OutOfMemory.

Corrigé en 7.x, cf <https://github.com/apereo/cas/pull/5652>

Solution, ajouter dans src/main/resources/application.yml :

```
server:
  tomcat:
    background-processor-delay: 10s
```

CAS 6.6.[0-1] : RegexRegisteredService ne fonctionne plus

"RegexRegisteredService" doit être remplacé par "CasRegisteredService".

Le msg de log indique que "RegexRegisteredService" est déprécié, mais en fait il ne fonctionne plus.

CAS >= 6.6.0 : pas de validation de tickets en parallèle pour la même application

Ce [commit](#) a modifié le comportement en cas où plusieurs tickets sont émis pour le même service (notamment avec des query parameters différents)

Pour revenir au comportement précédent :

```
cas.ticket.tgt.core.only-track-most-recent-session=false
```

NB : cela implique qu'un utilisateur se reconnectant sur la même application conservera tous les tickets dans le TGT ticket registry. Donc potentiellement des implications sur la taille...

CAS <= 6.1 : mélange de session

Cf "Danger du cache attribute-repository" sur la page [Paramètres importants de la configuration CAS](#).

Ticket registry MongoDB : consommation CPU

Un TextIndex est calculé, ce qui est très coûteux, et n'est plus nécessaire du tout en CAS 6.6 .

Détails : <https://groups.google.com/a/apereo.org/g/cas-user/c/B46dTjJa4Ac>

Solution pour CAS 6.6 : <https://github.com/apereo/cas/pull/5627>

Ticket registry JPA bug

Voir le [retour de l'URN sur mise en place de CAS 6.0.4](#)

RememberMe et ticket registry

tl;dr :

- ne pas utiliser Memcache pour le ticket registry si RememberMe
- avec redis, les TGT non remember-me sont nettoyés par le "registry cleaner"

Détails :

- [RememberMeDelegatingExpirationPolicy](#) peut choisir des ExpirationPolicy en fonction du TicketState. C'est notamment utilisé par la méthode `isExpired`.
- quand le [ticket registry cleaner](#) est actif, il utilise `isExpired`
- mais pas de "ticket registry cleaner" avec memcache (contrairement à redis) (ref : [MemcachedTicketRegistryConfiguration](#))
- à l'ajout du ticket dans memcache (pareil pour redis), c'est le default policy qui est utilisé (cf [MemcachedTicketRegistry](#) et [BaseDelegatingExpirationPolicy](#)). La durée de mise en memcache est indépendante de la case rememberMe...

Donc avec redis,

- mettre un très haut `cas.ticket.tgt.primary.max-time-to-live-in-seconds` (appelé `cas.ticket.tgt.hardTimeout.timeToKillInSeconds` en CAS 5.3) qui sera utilisé dans redis

- et CAS fera des passes (bien moches) pour appliquer les règles plus précises (cf `cas.ticket.registry.cleaner.schedule.repeatInterval` qui est toutes les 2 min par défaut)

CAS >= 6.6.8 : #xxx perdu dans l'url

Lorsqu'une application redirige vers CAS, le *fragment* d'url est ajouté par le navigateur. Par contre le form POST de la mire CAS peut perdre le fragment.

Depuis CAS 5.2, CAS tente de conserver le fragment, mais depuis la 6.6.8, le fragment est perdu.

Cf <https://github.com/apereo/cas/commit/1d9b5bad50493d6bca62f8e0ca38ca21c66199aa#r116140207>

CAS <= 6.5 : #xxx doublé dans l'url

Lorsqu'une application redirige vers CAS, le *fragment* d'url est ajouté par le navigateur. Par contre le form POST de la mire CAS peut perdre le fragment.

Depuis CAS 5.2, CAS tente de conserver le fragment, mais ce fragment est ajouté deux fois, ce qui perturbe certaines applications (Horde notamment)

Correctif : <https://github.com/apereo/cas/pull/5371/commits/f41689c05b8ab62921f4a21635f40c8d99025312>

Cf <https://github.com/apereo/cas/pull/5371>

CAS 6.4.[0-4] : délégation OpenID Connect

NB : le fix a été accepté et est inclus en 6.4.5

CAS 6.4 intègre `pac4j >= 5.1.4`, or `pac4j oidc` ne fonctionne pas avec `nimbus oidc < 9.14` :

```
java.lang.IllegalAccessException: class org.pac4j.oidc.profile.creator.OidcProfileCreator tried to access protected method com.nimbusds.oauth2.sdk.ProtectedResourceRequest.<init>(Ljava/net/URI;Lcom/nimbusds/oauth2/sdk/token/AccessToken;)V (org.pac4j.oidc.profile.creator.OidcProfileCreator and com.nimbusds.oauth2.sdk.ProtectedResourceRequest are in unnamed module of loader 'app')
```

Solution : modifiez `build.gradle` :

```
+ // pour délégation :
+ // ( https://github.com/apereo/cas/pull/5334 )
+ implementation "com.nimbusds:oauth2-oidc-sdk:9.14"
+ implementation "org.apereo.cas:cas-server-support-pac4j-webflow"
@@ -562,17 +559,17 @@ bootWar {
    cas {
        from "org.apereo.cas:cas-server-webapp${project.appServer}:${project.'cas.version'}@war"

        provided = false
-        excludes = ["WEB-INF/lib/servlet-api-2*.jar"]
+        excludes = ["WEB-INF/lib/servlet-api-2*.jar", "WEB-INF/lib/oauth2-oidc-sdk-9.13.jar"]
```

Détails : <https://github.com/apereo/cas/pull/5334>

CAS 6.4 : cas-server-support-trusted-mfa-redis bug

`cas-server-support-trusted-mfa-redis` seems to trigger `spring-boot RedisRepositoriesAutoConfiguration`, which fails to start with a `"redisTemplate"` error.

Solution : ajouter ceci dans `cas.properties` :

```
spring.data.redis.repositories.enabled: false
```