

SSO CAS - Logs applicatif

- [Préambule](#)
- [Installation dans le serveur CAS](#)
 - [Mon serveur CAS est il compatible ?](#)
 - [Rendre mon serveur compatible](#)
 - [CAS Version 4 ou 3.4.X \(GIT\)](#)
 - [CAS en version 3.4.X \(GIT\)](#)
 - [CAS en version 4.2.2](#)
 - [Intégration du serviceaudittrailmanager.java](#)
 - [Mise en place des logs](#)
- [Injection dans Agimus-NG \(Logstash\)](#)
- [Tableau de bord CAS](#)

Préambule

Il est possible de modifier le serveur CAS afin de générer un fichier de logs supplémentaire qui sera alimenté a chaque authentification pour un service. Le fichier généré au final aura la forme suivante :

```
i [DATE][IP:IP-CLIENT][ID:IDENTIFIANT-UTILISATEUR][TICKET:VALUER-TICKET-SERVICE-CAS][SERVICE:URL-SERVICE-ACCEDE][USER-AGENT:NAVIGATEUR-CLIENT]
[Wed Mar 18 11:45:31 CET 2015] [IP:XXX.XXX.XXX.X] [ID:dupont53][TICKET:ST-1-E5rdtcdOHsDneNgYl4ES-cas]
[SERVICE:http://ent.univ.fr/p/cms] [USER-AGENT:Mozilla/5.0 (Windows NT6.1; WOW64; rv:36.0) Gecko/20100101
Firefox/36.0]
```

Installation dans le serveur CAS

Mon serveur CAS est il compatible ?

Tout d'abord la dernière version (V4) du serveur CAS (cas-toolbox) génère ce fichier nativement : <https://github.com/EsupPortail/cas-toolbox-new>

Une version 3.4.X est aussi disponible : <https://github.com/EsupPortail/cas-toolbox-new/tree/v3.4.x>

Votre serveur CAS est compatible pour générer ce fichier si vous trouver un fichier nommé **ServiceAuditTrailManager.java** quelques part dans les sources.

Rendre mon serveur compatible

La modification comporte 3 étapes :

- Ajout d'un *FileAppender* et d'un *logger* dans la gestion de log log4j
- Ajout d'une classe Java qui implémente *AuditTrailManager* (classe *ServiceAuditTrailManager*)
- Modification de *auditTrailContext.xml* pour brancher la classe java *ServiceAuditTrailManager*

CAS Version 4 ou 3.4.X (GIT)

Vous pouvez utiliser ce pull request : <https://github.com/EsupPortail/cas-toolbox-new/pull/3>

et donc les fichier lister dans le PR : <https://github.com/EsupPortail/cas-toolbox-new/pull/3/files>

CAS en version 3.4.X (GIT)

Vous pouvez utiliser ce pull request : <https://github.com/EsupPortail/cas-toolbox-new/pull/3>

et donc les fichier lister dans le PR : <https://github.com/EsupPortail/cas-toolbox-new/pull/3/files>

<https://raw.githubusercontent.com/marchal-julien/cas-toolbox-new/6e1095c917781bcf6b833814aded75a3f31b39dd/cas-toolbox-core/src/main/java/org/esupportail/cas/audit/support/ServiceAuditTrailManager.java>

Ajouter dans le fichier *cas-toolbox-core/src/main/webapp/WEB-INF/classes/log4j.xml*

```

....
<appender name="serviceStatsAppender" class="org.apache.log4j.FileAppender">
  <param name="File" value="${log.dir}/serviceStats.log"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%m%n"/>
  </layout>
</appender>

...

<logger name="org.esupportail.cas.audit.support.ServiceAuditTrailManager" additivity="false">
  <level value="INFO" />
  <appender-ref ref="serviceStatsAppender" />
</logger>

....

```

Dans le fichier cas-toolbox-core/src/main/webapp/WEB-INF/spring-configuration/auditTrailContext.xml

```

....

<bean class="org.esupportail.cas.audit.support.ServiceAuditTrailManager"/>

....

```

CAS en version 4.2.2

- Dans le dossier cas-overlay-template récupérer le fichier auditTrailContext.xml et placez le dans src/main/webapp/WEB-INF/spring-configuration/

```

# wget -O src/main/webapp/WEB-INF/spring-configuration/auditTrailContext.xml https://raw.githubusercontent.com/apereo/cas/v4.2.2/cas-server-webapp/src/main/webapp/WEB-INF/spring-configuration/auditTrailContext.xml

```

- Dans le bean avec l'id auditTrailManagementAspect remplacez le constructeur

```
c:auditTrailManagers-ref="auditTrailManager"
```

par

```
c:auditTrailManagers-ref="auditTrailManagers"
```

en dessous de ce bean rajoutez ceci

auditTrailContext.xml

```
<bean id="serviceAuditTrailManager" class="org.esupportail.cas.audit.support.ServiceAuditTrailManager"/>

<util:list id="auditTrailManagers" value-type="java.lang.String">
  <ref bean="serviceAuditTrailManager"/>
  <ref bean="auditTrailManager"/>
</util:list>
```

Intégration du serviceaudittrailmanager.java

- Dans le dossier cas-overlay-template créez le dossier suivant:

```
# mkdir src/main/java/org/esupportail/cas/audit/support/
```

Insérez le fichier [ServiceAuditTrailManager.java](#) dans ce dossier

Mise en place des logs

- Éditez le fichier /etc/cas/log4j2.xml
- Dans la partie <Appenders> ajoutez ceci

log4j2.xml

```
    <RollingFile name="serviceStatsAppender" fileName="/home/tomcat/sso-cas/logs/serviceStats.log" append="
true"
  filePattern="/home/tomcat/sso-cas/logs/serviceStats-%d{yyyy-MM-dd-HH}-%i.log">
    <PatternLayout pattern="%m%n"/>
    <Policies>
      <SizeBasedTriggeringPolicy size="10 MB"/>
      <TimeBasedTriggeringPolicy />
    </Policies>
  </RollingFile>
```

- Dans la partie <Loggers>

log4j2.xml

```
    <AsyncLogger name="org.esupportail.cas.audit.support.ServiceAuditTrailManager" level="info" additivity="
false" includeLocation="true">
      <AppenderRef ref="console"/>
      <AppenderRef ref="serviceStatsAppender" />
    </AsyncLogger>
```

Injection dans Agimus-NG (Logstash)

Vous pouvez utiliser le fichier logstash-casrequest.conf : <https://github.com/EsupPortail/agimus-ng/blob/master/logstash/logstash-casrequest.conf>

Pour lire et injecter dans Agmius-NG le fichier résultant du CAS

Tableau de bord CAS

A venir :

- service
- agent
- 'eduPersonPrimaryAffiliation', 'supannEntiteAffectationPrincipale', 'supannEtuCursusAnnee', 'supannEtuSecteurDisciplinaire'