Région Provence-Alpes-Côte d'Azur

PASS TECH

CORRÉLYCE ATRIUM

Le catalogue ouvert régional de ressources éditoriales pour les lycées

# Guide d'interfaçage à destination des éditeurs

*révision 19*
*26 février 2019*

*Ce document de référence a été rédigé conjointement par la région PACA, PASS-TECH et le CRDP d'Aix-Marseille.*
*Il est mis en œuvre pour la plate-forme Correlyce. Les autres plate-formes basées sur le code Correlyce : Courdecol, Coreprim, l'ENT Aquitaine pourront suivre le même modèle afin d'assurer un accès unifié et normalisé par les éditeurs de ressources éditoriales.*

| | |
|---|---|
| **Origine** | Région PACA |
| **Auteur** | PASS Technologie |
| **Contributeur** | CRDP d'Aix-Marseille |
| **Document** | Documentation-editeur-SSO_2019_v19a.docx |

| **Mise à jour du document** | |
|---|---|
| **Révision** | **Commentaire** |
| V13 - 17/06/2014 | Première version publique du document |
| V14 - 18/09/2014 | Correction de <ENTProfils> en <ENTPersonProfils> dans les exemples de l'annexe 3 |
| V15 - 26/11/2014 | Ajout de la clef « preprod.atrium-paca » dans l'annexe 4 |
| V16 - 02/12/2014 | Ajout de ce tableau d'identification des changements |
| V17 - 11/12/2014 | Ajout d'exemples de code PHP et Java |
| V18 - 02/01/2017 | Ajout d'une remarque pour l'utilisation de https en PHP 5.6 |
| V19 - 26/02/2019 | Modification des URLs de caséification |

# Sommaire

# Préambule

Ce document présente succinctement le système d'authentification unique (SSO[1]) utilisé dans le projet CORRELYCE : CAS et particulièrement son utilisation pour la connexion aux ressources éditoriales en ligne.

Un SSO est un système qui permet de centraliser l'authentification au sein d'applications connexes. Ceci a plusieurs avantages :

- simplifier la gestion des mots de passe : c'est le même login / mot de passe qui sert pour plusieurs applications
- gagner du temps lors de l'utilisation de plusieurs applications successivement : un utilisateur connecté sur une application peut passer à une autre (s'il y est autorisé bien sûr) sans avoir à se réauthentifier
- simplifier la conception des applications en déportant la gestion de l'authentification vers un entrepôt central (utilisant un annuaire LDAP ou une base de données par exemple)

# Présentation de CAS

Note : cette présentation est fortement inspirée de l'article CAS[2] sur Wikipedia. CAS[3] est un système d'authentification unique développé par l'Université de Yale. C'est un mécanisme très solide, qui est implanté dans plusieurs universités et organismes dans le monde. CAS est une application Web écrite en Java et distribuée comme un logiciel libre.

## Intérêt

CAS évite de s'authentifier à chaque fois qu'on accède à une application en mettant en place un système de ticket. CAS est un système de SSO : on s'authentifie sur un site Web, et on est alors authentifié sur tous les sites web qui utilisent le même serveur CAS.

## Fonctionnement de base

Dans son fonctionnement de base, une application CASifiée[4] s'en remet à CAS pour son authentification. Lors de l'accès d'un utilisateur à une page protégée, le serveur CAS est interrogé. Si l'utilisateur est identifié sur le serveur CAS, c'est cette identification qui est soumise au service demandé ; si l'utilisateur n'est pas authentifié sur le serveur CAS, le formulaire d'authentification est affiché.

L'existence d'une authentification sur le serveur CAS est conservée sur le navigateur du client sous la forme d'un cookie. De même, tous les échanges entre l'application demandée et le serveur CAS sont réalisés au moyen de redirections via le navigateur (HTTP 302)

---

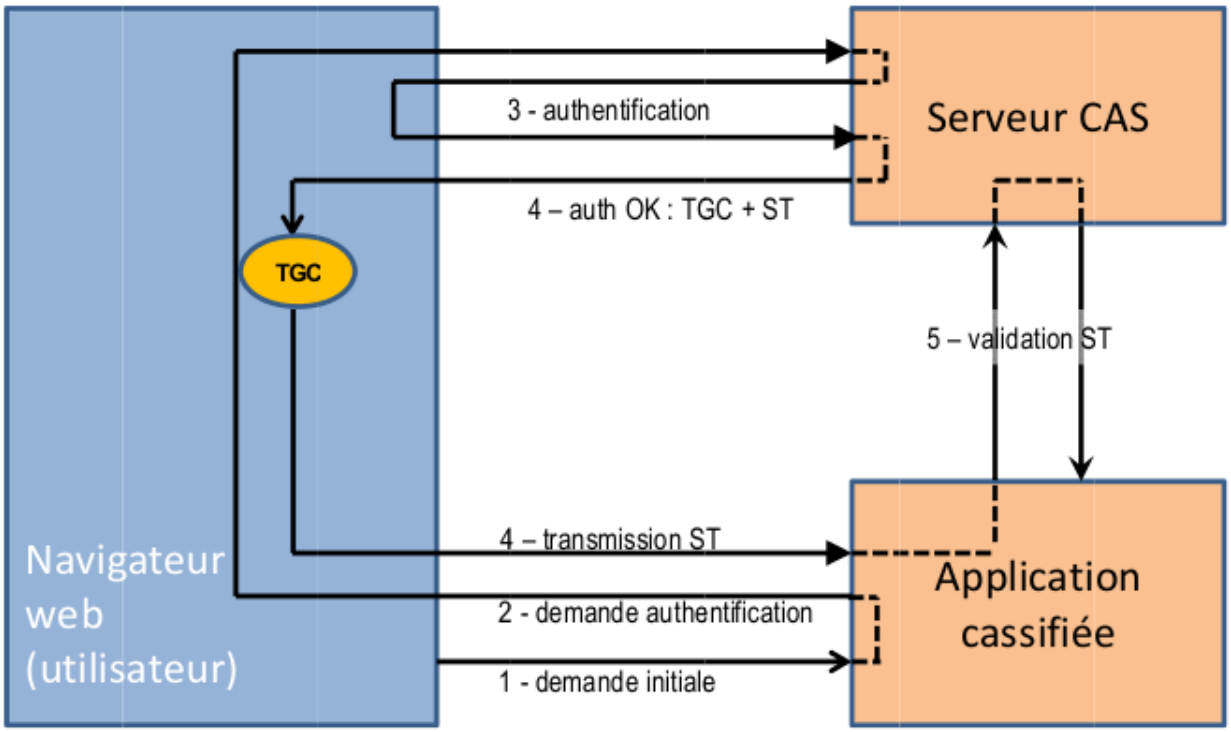
1 Single Sign On : Authentification unique

2 http://fr.wikipedia.org/wiki/Central_Authentication_Service

3 Central Authentication Service : http://www.ja-sig.org/products/cas/index.html

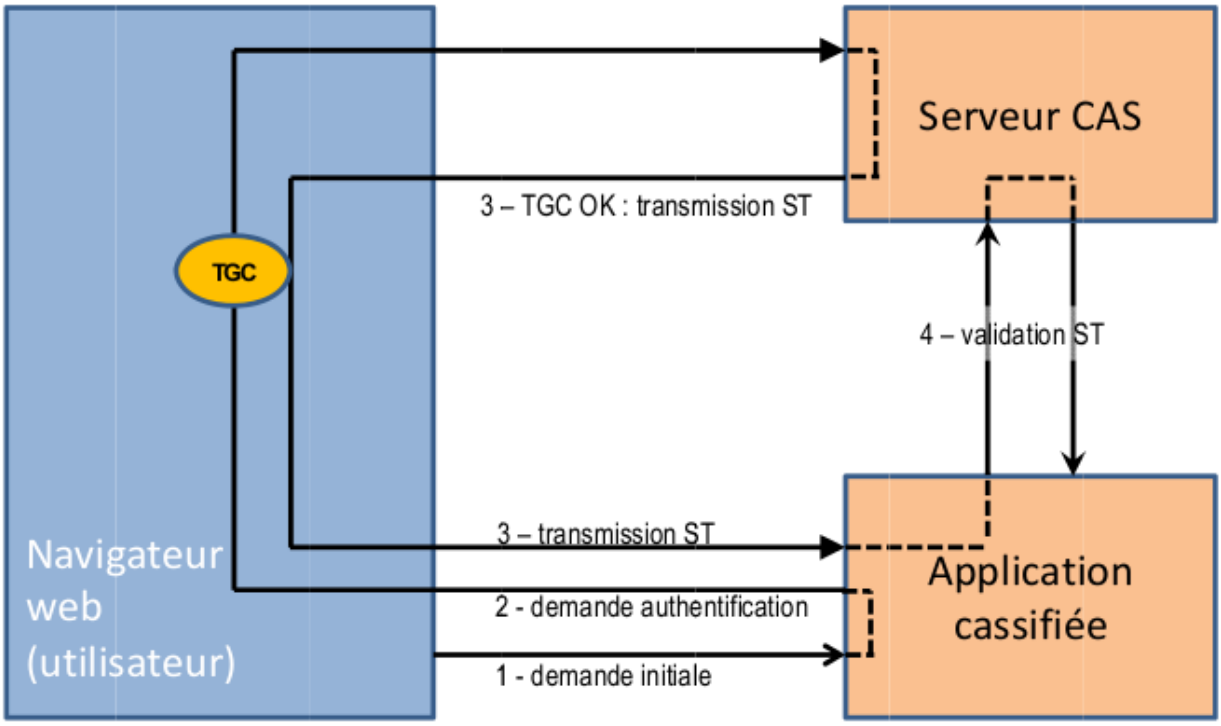
4 Une application CASifiée est une application qui se connecte à un serveur CAS pour son identification

## Synoptique des échanges (1ère connexion)



## Synoptique des échanges (connexion suivante)

Dès lors, l'utilisateur est authentifié auprès du serveur CAS, s'il cherche à accéder à une page protégée d'une autre application connectée au même serveur CAS :




1. L'utilisateur accède à une page protégée d'une autre application connectée au même serveur CAS
2. Il est redirigé vers le serveur CAS avec l'adresse de la page en paramètre
3. le serveur CAS reconnaît le cookie d'autorisation de ticket (TGC) et redirige l'utilisateur vers la page de l'application avec le ticket de service (ST) en paramètre
4. L'application valide le ST reçu auprès du serveur CAS et reçoit en échange l'information de login sur l'utilisateur

Note : l'utilisateur n'a pas eu à se ré-authentifier (c'est tout l'intérêt du SSO).

Ce fonctionnement de base est utilisé dans Correlyce et permet, par exemple, à un utilisateur connecté à l'application Correlyce d'être reconnu lorsqu'il bascule sur l'application SPIP actus.

## Ticket-Granting Cookie (TGC)

C'est un cookie de session qui est transmis par le serveur CAS au navigateur du client lors de la phase de login. Ce cookie ne peut être lu / écrit que par le serveur CAS, sur canal sécurisé (https). Ce cookie est présent durant toute la session de l'utilisateur et permet d'obtenir des Service Ticket auprès du serveur CAS.

Si le navigateur web n'accepte pas les cookies, l'utilisateur devra se ré-authentifier à chaque appel au serveur CAS.

## Service Ticket (ST)

Ce ticket va servir à authentifier une personne pour une application web donnée. Il est envoyé par le serveur CAS après que l'utilisateur s'est authentifié et est transporté dans l'URL.

Ce ticket ne peut être utilisé qu'une seule fois. Il y a ensuite dialogue direct entre l'application web et le serveur CAS via un GET http, avec le ST en paramètre. En réponse, le serveur CAS retourne l'identifiant de la personne, et donc l'authentifie. Il invalide également le ticket (libération des ressources associées).

En fait, ce ticket concerne une personne, pour un service. Il n'est utilisable qu'une seule fois.

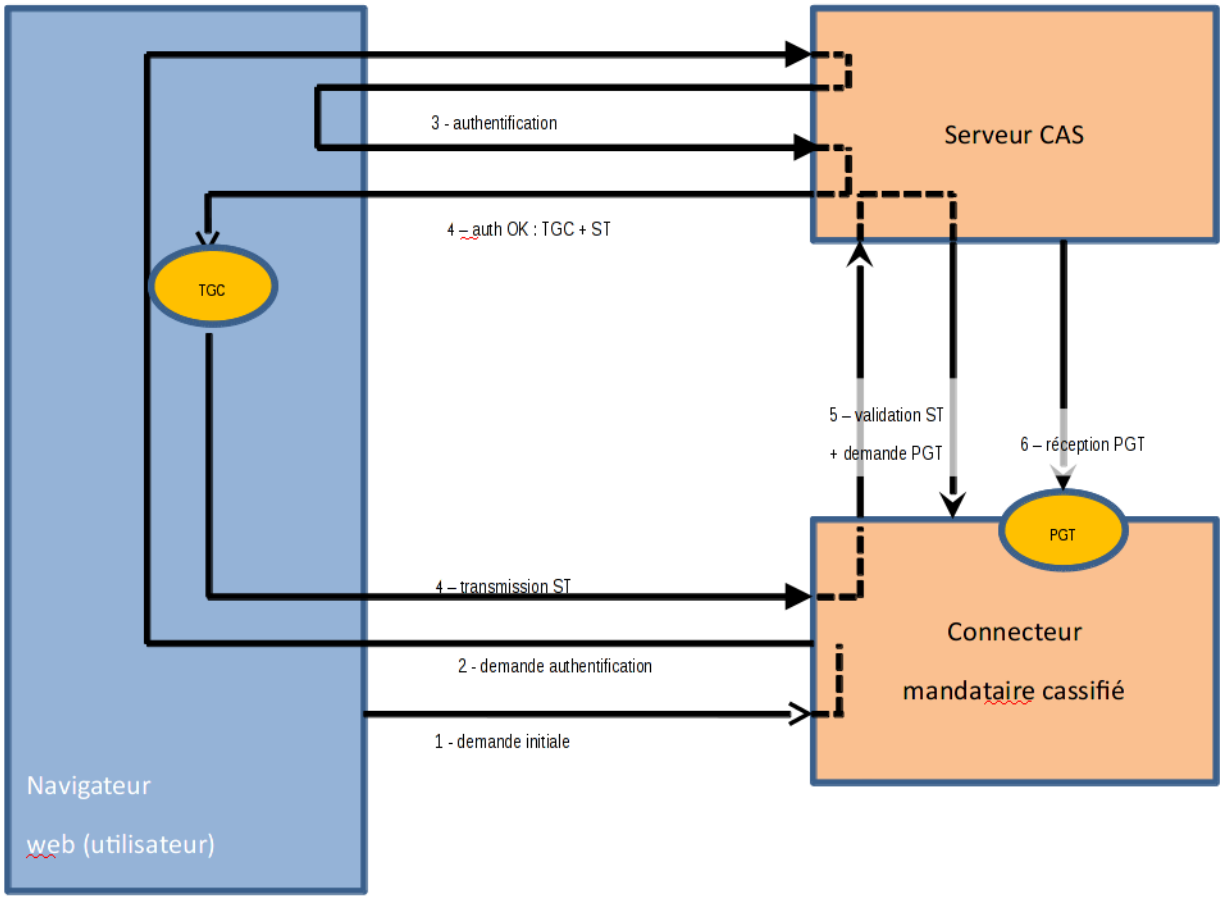
Le service ticket est de la forme « ST-.* », ex : « ST-956-Lyg0BdLkgdrBO9W17bXS »

## Fonctionnement en mode proxy CAS[5]

Le fonctionnement en mode proxy permet à une application proxy CAS de donner accès à d'autres applications clientes sous condition. Ces applications clientes valideront le ticket transmis par l'application proxy auprès du serveur CAS.

- c'est l'application proxy qui est chargée de faire tous les tests nécessaires pour autoriser ou non l'accès à l'application cliente demandée (i.e. générer ou non le PT nécessaire)
- l'application cliente ne peut être accédée par l'utilisateur qu'en passant par l'application proxy

*Correlyce joue le rôle de proxy CAS pour l'accès des utilisateurs aux ressources éditoriales. Lorsqu'un utilisateur souhaite accéder à une*

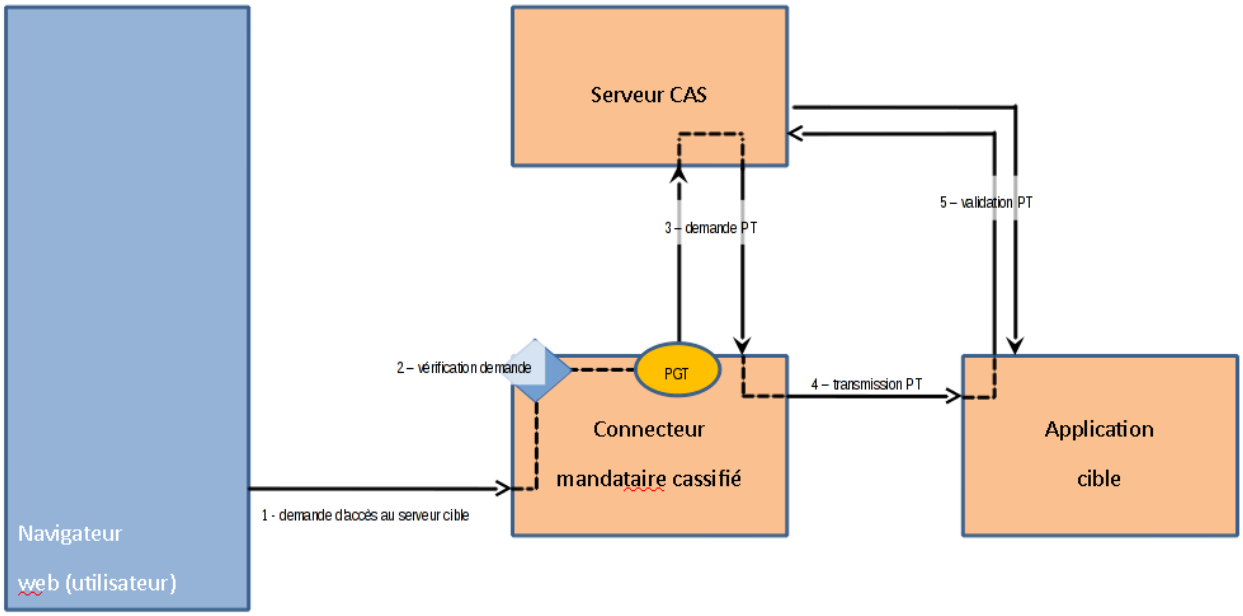


3. Le serveur CAS vérifie son identité
4. si l'authentification est réussie, l'utilisateur est redirigé vers l'application avec le ticket de service (ST) en paramètre et positionne le cookie TGC.
5. L'application valide le service ticket auprès du serveur CAS en demandant à être recontacté pour obtenir un ticket PGT, le serveur CAS renvoie l'identité de l'utilisateur
6. Le serveur CAS rappelle l'application avec le PGT en paramètres

L'application mandataire a maintenant la possibilité de générer des tickets proxy pour l'accès à un service tiers.

## Synoptique des échanges (accès à une application tierce)

L'utilisateur connecté souhaite accéder à une application cliente via l'application proxy




On considère, par souci de simplification, que l'utilisateur est déjà authentifié à l'application mandataire.

1. L'utilisateur accède à l'application mandataire pour demander l'accès à l'application cible (via un lien sur une page par exemple)
2. L'application mandataire évalue la demande (l'utilisateur a-t-il droit d'accéder à l'application cible ?)
3. Si l'utilisateur est autorisé, l'application mandataire requiert un ticket proxy (PT) auprès du serveur CAS en passant le PGT et l'url de la cible en paramètre
4. Le serveur CAS renvoie le ticket proxy et l'application mandataire redirige l'utilisateur vers l'application cible avec le ticket proxy en paramètre.
5. L'application cible valide le ticket reçu auprès du serveur CAS pour vérifier son authenticité et si celui-ci est validé, recevoir des informations sur l'utilisateur.

Notes :

- c'est l'application proxy (ici le catalogue) qui est chargée de faire tous les tests nécessaires pour autoriser ou non l'accès à l'application tierce (site de l'éditeur) demandée (i.e. générer ou non le PT nécessaire)
- l'application tierce ne peut être accédée par l'utilisateur qu'en passant par l'application proxy
- dans Correlyce, le fonctionnement en mode proxy est utilisé pour donner accès aux ressources des éditeurs, cela permet de vérifier si l'utilisateur qui demande la ressource y a bien droit (abonnement) avant de générer le ticket nécessaire.
- L'url cible pour laquelle est demandé le ticket est composée de l'url contenue dans le document ScoLomfr de description de la ressource, à laquelle sont ajoutés deux paramètres :
  - « uai » : Ce paramètre permet de définir la structure courante de l'utilisateur qui

se connecte[6]. Cela concerne certains utilisateurs qui peuvent faire partie de plusieurs établissements ;
- « pf » : Ce paramètre sert à identifier la plate-forme à partir de laquelle l'utilisateur cherche à se connecter[7].

Ces deux paramètres sont à conserver lors de la vérification du ticket CAS.

### Proxy-Granting-Ticket (PGT)

Il est envoyé par le serveur CAS à une application web 'proxy CAS' disposant d'un ST valide. Ce ticket confère au proxy CAS la possibilité de demander au serveur CAS de générer un Proxy Ticket (PT) pour une application tierce et un utilisateur donné.

### Proxy-Ticket (PT)

Il est généré par le serveur CAS à la demande d'un proxy CAS. Il permet d'authentifier l'utilisateur pour un service distant, avec lequel le client web n'a pas d'accès direct. Le service distant l'utilisera comme le ST mais sur une URL différente. CAS dispose de deux services distincts pour la validation de ses tickets :

- /serviceValidate (pour les ST)
- /proxyValidate (pour les PT)

Le PT est lié au service distant et n'est pas rejouable.

Le PT est de la forme « PT-.* », ex : « PT-957-ZuucXqTZ1YcJw81T3dxf ». Selon les implémentations de CAS, le PT peut commencer par « ST- ».

# Implémentation côté éditeur

L'utilisation de CAS est basée sur des standards de l'internet : HTTP(s) et XML et ne nécessite pas l'usage d'un langage de programmation spécifique.

Pour pouvoir prendre en compte les demandes d'accès provenant de Correlyce, deux opérations sont nécessaires :

1. Mettre en place une page d'accès à la ressource
2. Indiquer l'adresse de cette page dans le document ScoLOMFR correspondant

## Mise en place

### Création d'une page d'accès à la ressource

La page d'accès à la ressource doit être une page « dynamique » (dans un langage tel que Java, PHP, ASP, ...) et doit répondre selon le protocole HTTPS.

Actions à réaliser pour vérifier la validité de l'appelant :

1. vérifier qu'un paramètre nommé « ticket » est bien passé en paramètre, sinon afficher un message d'erreur
2. appeler la page https://www.atrium-paca.fr/connexion/proxyValidate[8] avec 2 paramètres :
   1. « ticket » : le paramètre ticket passé en paramètre URL
   2. « service » : l'URL de la page courante[9]
3. analyser la réponse XML du serveur CAS :

[6] Voir Annexe 5 : Identification de l'établissement courant

[7] Voir Annexe 4 : Identification de la plate-forme appelante

[8] L'URL de validation dépend de la plate-forme appelante, voir annexe « Identification de la plate-forme appelante »

1. erreur d'authentification : afficher une erreur à l'utilisateur
2. authentification réussie : extraire les informations d'identifiant numérique d'utilisateur, son profil, le code UAI de son établissement[10] et donner accès à la ressource

Notes :

- Les librairies clientes CAS disponibles[11] dans de nombreux langages de programmation cachent ces échanges clients-serveurs. Il suffit généralement de leur fournir l'adresse du serveur CAS (https://www.atrium-paca.fr) et d'indiquer que la page est un client proxy CAS.
- Un éditeur / diffuseur qui propose plusieurs ressources à Correlyce n'est pas obligé de mettre en place une page d'accès par ressource mais peut proposer un point d'entrée unique avec un paramètre URL différent par ressource (paramètre « idressource » par exemple qui contient un identifiant unique de ressource dans le système de l'éditeur).

## Informations techniques

### Pratique

- Le serveur CAS Atrium de production est à l'adresse :
  https://www.atrium-paca.fr/connexion
- L'url d'accès à la ressource doit être disponible en https.

### Réponse de validation du ticket proxy

Le tableau ci-dessous décrit la liste des informations disponibles dans le ticket de validation proxy CAS :

| Balise | Description | Commentaire |
|---|---|---|
| Identification de l'utilisateur | | |
| <cas:user> | Identifiant numérique de l'utilisateur<br>Ex : « Uaa00124 » | Balise obligatoire et unique |
| **Informations supplémentaires**<br><br>Les balises autres que <cas:user> sont regroupées dans une balise parente <cas:attributes>, cette convention est implémentée par le serveur CAS de Jasig et supportée nativement par les clients CAS officiels[12]. | | |
| <cas:ENTPersonProfils> | Profil de l'utilisateur,<br>Ex : « National_ELV » pour un élève ou « Catalogue_EDI » pour un éditeur | Balise obligatoire et répétable<br>Utilisation de la nomenclature SDET v4 définie dans l'annexe interopérabilité + profils locaux, cf. annexe « *Annexe 1 :* |

[9] Attention, l'URL de cette page doit être celle inscrite dans le document ScoLOMFR à laquelle sont ajoutés les paramètres « uai » et « pf » passés en paramètres. En effet, le ticket PT généré par CAS prend en compte cette URL et la validation du PT vérifie l'adéquation de l'URL passée lors de la génération du PT et l'url de service fournie.

[10] cf. section « Informations techniques » pour la liste des informations disponibles

[11] Voir la page https://wiki.jasig.org/display/CASC/Home.

[12] https://wiki.jasig.org/display/CASUM/Attributes

| | | |
|---|---|---|
| | | *codes SDET pour le profil de l'utilisateur* » |
| <cas:ENTPersonStructRattachUAI> | Code UAI de l'établissement, Ex : « 0130068D » | Balise obligatoire et monovaluée correspondant à l'établissement auquel est rattaché administrativement l'utilisateur.<br>Dans le cas des administrateurs et des éditeurs, un code UAI fictif sera utilisé. cf. annexe « *Annexe 2 : Codes UAI* » |
| <cas:ENTStructureTypeStruct> | Type d'établissement<br>Ex : « LYC » | Balise obligatoire et monovaluée<br>Utilisation du code TYPE UAI défini dans la table infocentre N_NATURE_UAI[13] |
| Informations propres aux élèves | | |
| <cas:ENTEleveMEF> | Code MEF du niveau de l'élève<br>ex : « 21231012110 » | Balise répétable<br>Utilisation du code MEF à 11 chiffres défini dans la table infocentre N_MEF[14] |
| <cas:ENTEleveClasses> | Libellé de la classe de l'élève dans l'établissement précédé par le numéro UAI de l'établissement suivi du caractère « $ », ex : « 0040007L$1ère S 3 » | Balise répétable |
| Informations propres aux enseignants | | |
| <cas:ENTAuxEnsMEF> | Codes MEF des niveaux pour lesquels enseigne le professeur. | Balise répétable |
| <cas:ENTAuxEnsClasses> | Liste des noms des classes dans lesquelles intervient l'enseignant, chaque classe est précédée du numéro UAI de l'établisssement suivi de $.<br>Ex : « 0040010P$2DE 3 », « 0040011R$1ERE S 2 » | Balise répétable |

## URL d'accès à la ressource éditoriale

L'adresse URL de la page d'accès à la ressource doit être consignée dans la section « 4.3 localisation » de la description ScoLOMFR correspondante.

Cette information est visible uniquement de l'éditeur de la ressource ou de l'administrateur dans l'application Correlyce et n'est pas divulguée à l'extérieur.

[13] Disponible en ligne http://infocentre.pleiade.education.fr/bcn/workspace/viewTable/n/N_NATURE_UAI

[14] La liste des codes MEF est disponible dans la table N_MEF de l'infocentre disponible en ligne : http://infocentre.pleiade.education.fr/bcn/workspace/viewTable/n/N_MEF

## Test

Les tests d'accès à la ressource peuvent être effectués directement par les éditeurs dans leur interface de gestion des titres :

- cliquer sur le contenu de la colonne « Accès » du titre choisi
- cliquer sur le bouton « tester » dans la nouvelle page
- l'utilisateur est alors redirigé vers l'url d'accès indiquée dans le document ScoLOMFR avec un ticket valide en paramètre

# Annexes

## Annexe 1 : codes SDET pour le profil de l'utilisateur

| Identifiant du profil | Description du périmètre |
|---|---|
| National_ELV | Élève |
| National_TUT | Responsable d'un élève (parent, tuteur légal) |
| National_ENS | Enseignant |
| National_DIR | Personnel de direction de l'établissement |
| National_EVS | Personnel de vie scolaire travaillant dans l'établissement |
| National_ETA | Personnel administratif, technique ou d'encadrement travaillant dans l'établissement |
| National_ACA | Personnel de rectorat, de DRAF, d'inspection académique |
| National_DOC | Documentaliste |
| National_COL | Personnel de collectivité territoriale |

*Tableau des profils utilisateur extrait du document « SDET v4.0 annexe Interopérabilité (mise à jour du 14 décembre 2012 »*[15]

À ces profils nationaux, on ajoute 2 profils locaux :

| Identifiant du profil | Description du périmètre |
|---|---|
| Catalogue_EDI | Éditeur de ressources |
| Catalogue_ADM | Administrateur du catalogue |

## Annexe 2 : Codes UAI

L'UAI, pour Unité Administrative Immatriculée1 est un code unique pour les établissements scolaires. Il est composé de 7 chiffres et une lettre. Les 3 premiers chiffres identifient le département, les 4 chiffres suivants identifient l'établissement de manière unique au sein du département. La lettre finale est une somme de contrôle calculée d'après les 7 premiers chiffres.

Les utilisateurs de profil « administrateur » et « éditeur », n'ont pas d'établissement scolaire associé. Aussi, préconisons-nous l'utilisation de codes UAI fictifs forgés comme suit :

- « ADM » suivi de 5 chiffres (ex : « ADM00001 ») pour l'établissement fictif associé à un administrateur
- « EDI » suivi de 5 chiffres (ex : « EDI12345 ») pour l'établissement fictif associé à

[15] Document « SDET-Interoperabilite-v4.0_226607.pdf » disponible en ligne sur la page http://eduscol.education.fr/cid56994/sdet-version-4.html.

un éditeur

## Annexe 3 : retour de validation CAS

### Échec de validation

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
        <cas:authenticationFailure code='INVALID_TICKET'>
le ticket 'ST-63-PJraQneKSfeXgrl4pvn20sXlFKQHcLWuMrU-20' est inconnu
        </cas:authenticationFailure>
</cas:serviceResponse>
```

*exemple de retour en cas d'échec de validation du ticket proxy*

Note : la présence de la balise <cas:authenticationFailure> indique que la validation a échoué.

### Validation réussie

Note : la présence de la balise <cas:authenticationSuccess> indique que la validation du ticket a fonctionné.

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
<cas:authenticationSuccess>
           <cas:user>Uam00010</cas:user>
     <cas:attributes>
                <cas:ENTPersonProfils>National_ELV</cas:ENTPersonProfils>


<cas:ENTPersonStructRattachUAI>0131313Z</cas:ENTPersonStructRattachUAI>

     <cas:ENTStructureTypeStruct>LYC</cas:ENTStructureTypeStruct>
                <cas:ENTEleveMEF>21231012110</cas:ENTEleveMEF>
                <cas:ENTEleveClasses>0131313Z$Terminale STG
3</cas:ENTEleveClasses>
           </cas:attributes>
           <cas:proxies>
                <cas:proxy>https://www.atrium-
paca.fr/.../receptor</cas:proxy>
           </cas:proxies>
</cas:authenticationSuccess>
</cas:serviceResponse>
```

*exemple de retour pour un utilisateur élève*

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
<cas:authenticationSuccess>
           <cas:user>Uib00006</cas:user>
           <cas:attributes>
```

```
                <cas:ENTPersonProfils>National_ENS</cas:ENTPersonProfils>


<cas:ENTPersonStructRattachUAI>0130151U</cas:ENTPersonStructRattachUAI>

     <cas:ENTStructureTypeStruct>LP</cas:ENTStructureTypeStruct>
                <cas:ENTAuxEnsMEF>2463140122</cas:ENTAuxEnsMEF>
                <cas:ENTAuxEnsMEF>2112220711</cas:ENTAuxEnsMEF>
                <cas:ENTAuxEnsClasses>0130151U$Tle Pro Compta
2</cas:ENTAuxEnsClasses>
                <cas:ENTAuxEnsClasses>0130151U$1ère Biochimie Génie Bio
A</cas:ENTAuxEnsClasses>
           </cas:attributes>
           <cas:proxies>
                <cas:proxy>https://www.atrium-
paca.fr/.../receptor</cas:proxy>
           </cas:proxies>
</cas:authenticationSuccess>
</cas:serviceResponse>
```

*exemple de retour pour un utilisateur enseignant*

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
<cas:authenticationSuccess>
           <cas:user>Uza00006</cas:user>
           <cas:attributes>
                <cas:ENTPersonProfils>Catalogue_EDI</cas:ENTPersonProfils>


<cas:ENTPersonStructRattachUAI>EDI54673</cas:ENTPersonStructRattachUAI>

     <cas:ENTStructureTypeStruct>ENTR</cas:ENTStructureTypeStruct>
           </cas:attributes>
           <cas:proxies>
                <cas:proxy>https://www.atrium-
paca.fr/.../receptor</cas:proxy>
           </cas:proxies>
</cas:authenticationSuccess>
</cas:serviceResponse>
```

*exemple de retour pour un utilisateur éditeur*

## Annexe 4 : Identification de la plate-forme appelante

La normalisation de la réponse de validation du ticket proxy CAS constitue une étape importante pour la standardisation de l'accès aux différents plate-formes de type Correlyce.

Pour aller plus loin, nous ajoutons automatiquement un paramètre URL supplémentaire « pf » lors de l'accès à la ressource éditeur[16]. Sa valeur permet d'identifier de manière unique la plate-forme appelante.

Ainsi, côté éditeur, une seule page d'accès proxy CAS sera nécessaire pour l'ensemble des plate-formes, la valeur du paramètre « **pf** » déterminant l'URL de validation du ticket proxy CAS.

À ce jour, les plate-formes recensées sont les suivantes :

| Valeur du paramètre « pf » | URL de validation CAS correspondante |
|---|---|
| atrium-paca | https://www.atrium-paca.fr/connexion/proxyValidate |
| recette.atrium-paca | https://recette.atrium-paca.fr/connexion/proxyValidate |
| preprod.atrium-paca | https://preprod.atrium-paca.fr/connexion/proxyValidate |

## Annexe 5 : Identification de l'établissement courant

Dans la réponse de validation du ticket proxy CAS, le contenu de la balise <ENTPersonStructRattachUAI> contient le code UAI de l'établissement de rattachement administratif de l'utilisateur. Pour un élève, c'est l'établissement dans lequel il étudie. Pour un enseignant, c'est l'établissement dans lequel il enseigne.

Certains enseignants interviennent dans plusieurs établissements, certains élèves suivent des cours dans un établissement autre que le leur. Au sein du catalogue, ces utilisateurs ont la possibilité de choisir l'établissement pour lequel ils agissent. Lors de la connexion à une ressource éditoriale, l'information d'établissement de rattachement administratif ne reflète pas forcément l'établissement courant choisi au sein du catalogue. Dans ce cas particulier, on pourra se reporter au paramètre URL « **uai** » ajouté automatiquement à l'URL de service et portant l'information de l'établissement courant.

Ce paramètre est présent systématiquement lors de l'appel à la ressource mais ne diffère du contenu de la balise <ENTPersonStructRattachUAI> que dans le cas des utilisateurs agissant dans un établissement autre que celui auquel ils sont rattachés.

[16] En plus du paramètre « ticket » contenant le ticket proxy CAS

## Annexe 6 : Exemple d'échanges par urls

Contexte :

- Votre url d'accès à la ressource est : https://monserveur/mapage?id=23
- L'utilisateur est connecté sur la plate-forme de production Correlyce. Le code plate-forme correspondant est donc « atrium-paca »
- L'utilisateur est connecté sur l'établissement dont le code UAI est « 1234567Z »

### Cinématique

L'utilisateur abonné clique sur le titre de la ressource, Correlyce constitue l'URL complète d'accès à l'éditeur :

*https://monserveur/mapage?uai=1234567Z&pf=atrium-paca*

Correlyce demande un ticket proxy CAS pour cette URL auprès du serveur et redirige l'utilisateur vers l'éditeur avec le paramètre ticket :

*https://monserveur/mapage?uai=12344567Z&pf=atrium-paca&ticket=PT-345-Lyg0BdLkgdrBO9W17bXS*

La page d'accès de l'éditeur détermine l'URL du serveur CAS à contacter d'après le paramètre pf. Elle appelle donc la page de validation du ticket proxy avec le ticket (PT-345-Lyg0BdLkgdrBO9W17bXS) et l'URL de service (https://monserveur/mapage?uai=12344567Z&pf=atrium-paca) encodée en paramètre :

*https://www.atrium-paca.fr/connexion/proxyValidate?ticket=PT-345-Lyg0BdLkgdrBO9W17bXS&service=https%3A%2F%2Fmonserveur%2Fmapage%3Fuai%3D1234567Z%26pf%3Datrium-paca*

Le serveur CAS valide le coupe (URL de service, ticket) et renvoie les informations XML sur l'utilisateur connecté.

### Détail des échanges[17] :

1. [SERVEUR] Correlyce demande un ticket auprès du serveur CAS pour l'url :

   *https://monserveur/mapage?uai=1234567Z&pf=atrium-paca*

2. [SERVEUR] CAS génère le ticket correspondant et le renvoie à Correlyce

[17] Les échanges préfixés [SERVEUR] ne sont pas visibles de l'utilisateur : Ils sont exécutés côté serveur. Ceux préfixés [NAVIGATEUR] correspondent à une redirection du navigateur de l'utilisateur courant.

3. [SERVEUR] Correlyce reçoit le ticket « PT-345-Lyg0BdLkgdrBO9W17bXS »
4. [NAVIGATEUR] L'utilisateur est redirigé vers

> *https://monserveur/mapage?uai=1234567Z&pf=atrium-paca&ticket=PT-345-Lyg0BdLkgdrBO9W17bXS*

5. [SERVEUR] La page d'accès à la ressource demande la validation du ticket auprès du serveur CAS :

> *https://www.atrium-paca.fr/connexion/proxyValidate&ticket=PT-345-Lyg0BdLkgdrBO9W17bXS&service=https%3A%2F%2Fmonserveur%2Fmapage%3Fuai%3D1234567Z%26pf%3Datrium-paca*

6. [SERVEUR] La page d'accès reçoit la réponse XML du serveur CAS et l'analyse
7. L'utilisateur accède à la ressource demandée

## Annexe 7 : Client proxy CAS : exemples de code

Cette annexe présente des exemples de code permettant de réaliser la page de validation du ticket proxy CAS

### Script d'accès exemple en PHP

```
<?php
   /*
================================================================================
    * = Constantes
    *
================================================================================
*/
   // La liste des URLS de validation CAS en fonction
   // du paramètre "pf"
   // Ce tableau pourra être complété lorsque d'autres
   // plate-formes de type Correlyce adopteront ce type de
   // connexion
   $CAS_VALIDATION_URLS = array(
      'atrium-paca'
         => 'https://www.atrium-paca.fr/connexion/proxyValidate',
      'recette.atrium-paca'
         => 'https://recette.atrium-paca.fr/connexion/proxyValidate',
      'preprod.atrium-paca'
         => 'https://preprod.atrium-paca.fr/connexion/proxyValidate',
   );

   // L'adresse de la ressource que vous voulez mettre à disposition de Correlyce
   define('SERVICE_URL', 'https://demo.tech.fr/correlycev2/');

   // Un peu de style
   $CSS = <<<EOS
      xmp { font: courier; background-color: #e0e0e0; padding: 1em;
         width: 70em; }
EOS;

   /*
================================================================================
    * = Fonctions
    *
================================================================================
*/
   /**
    * Affichage d'une page d'erreur
    */
   function showError($errtitle, $errmsg, $xmlcontent='') {
      global $CSS;

      if ($xmlcontent != '') {
         $xmlcontent = "<h3>Contenu de la réponse :</h3><xmp>$xmlcontent</xmp>";
      }

      $html =<<<EOH
```

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset=UTF-8>
    <title>Accès interdit : $errtitle</title>
    <style>
        $CSS
    </style>
</head>
<body>
    <h1>Accès interdit</h1>
    <h2>$errtitle</h2>
    <p>$errmsg</p>
    $xmlcontent
</body>
</html>
EOH;
        die($html);
    }

    /**
     * Formatage du contenu XML pour le rendre plus lisible
     */
    function formatXMLContent($xmlcontent) {

        // Suppression des lignes vides
        $xmlcontent = preg_replace("/^\n+|^[\t\s]*\n+/m", "", $xmlcontent);

        // Correction de la sur-indentation
        $xmlcontent = preg_replace("/\t{5}/", "\t\t\t", $xmlcontent);

        // Remplacement des tabulations par 4 espaces
        $xmlcontent = str_replace("\t", '    ', $xmlcontent);

        return $xmlcontent;
    }

    /*
================================================================================
     * = Programme principal
     *
================================================================================
*/
    // Récupération du ticket
    $ticket = @$_GET['ticket'];

    // Pas de ticket passé en paramètre => refuser l'accès
    if ($ticket == '') {
        showError("Erreur n°1", "ticket non présent");
    }

    // Récupération de l'identifiant de plate-forme
    $pf = @$_GET['pf'];

    // Pas d'identifiant => refuser l'accès
    if ($pf == '') {
        showError("Erreur n°2", "identifiant de plate-forme (pf) absent");
    }
```

```
// Identifiant de plate-forme inconnu => refuser l'accès
if (!in_array($pf, array_keys($CAS_VALIDATION_URLS))) {
    showError("Erreur n°3", "identifiant de plate-forme inconnu : ["
        .htmlspecialchars($pf)."]");
}

// URL de validation CAS
$casValidationURL = $CAS_VALIDATION_URLS[$pf];

// On constitue l'URL de service en partant de l'URL
// d'appel et en retirant le paramètre ticket
$paramStr = $_SERVER['QUERY_STRING'];
if (($tidx = strpos($paramStr, '&ticket')) !== false) {
    $paramStr = substr($paramStr, 0, $tidx);
}
$serviceURL = SERVICE_URL.'?'.$paramStr;

// Validation du ticket reçu
$validationURL = $casValidationURL.'?ticket='.$ticket
    .'&service='.urlencode($serviceURL);

// Retour de validation CAS
if (($xmlcontent = file_get_contents($validationURL)) === false) {
    showError('Erreur n°4', "Impossible d\'accéder au serveur CAS pour la validation '
        . 'du ticket reçu<br/>URL = '.$validationURL);
}

// Formatage du contenu XML pour une meilleure lisibilité
$xmlcontent = formatXMLContent($xmlcontent);

// Test d'analyse XML du contenu
try {
    $dom = new DOMDocument();
    $dom->loadXML($xmlcontent);
} catch (Exception $e) {
    showError('Erreur n°5', "La réponse de validation n'est "
        .'pas un contenu XML valide', $xmlcontent);
}

$rootElt = $dom->documentElement;
if ($rootElt->tagName != 'cas:serviceResponse') {
    showError('Erreur n°6', "La réponse de validation est incorrecte",
        $xmlcontent);
}

// Recherche du premier élément
foreach ($rootElt->childNodes as $node) {
    if ($node->nodeType == XML_ELEMENT_NODE) {
        break;
    }
}

// Erreur CAS
if ($node->tagName == 'cas:authenticationFailure') {
    $errmsg = $node->getAttribute('code');
    showError('Erreur n°7', "CAS a rencontré une erreur : « $errmsg »",
        $xmlcontent);
```

```
    }

    // Récupération des infos utilisateurs
    $uaiParam = @$_GET['uai'] != '' && $_GET['uai'] != 'null' ? $_GET['uai'] : '';

    // Identifiant utilisateur
    $user = $dom->getElementsByTagName('user')->item(0)->textContent;

    // Infos complémentaires
    $attElt = $dom->getElementsByTagName('attributes')->item(0);
    $attChildren = $attElt->getElementsByTagName('*');

    // On constitue une table de hachage avec les informations
    // complémentaires
    $map = array();
    for ($i=0; $i<$attChildren->length; $i++) {
        $attChild = $attChildren->item($i);
        $key = $attChild->tagName;
        $val = $attChild->textContent;

        if (!isset($map[$key])) {
            $map[$key] = array();
        }
        $map[$key][] = $val;
    }

// Et on affiche la page de succès
?>
<!DOCTYPE HTML>
<html>
<head>
    <meta charset=UTF-8>
    <title>Accès autorisé !</title>
    <style><?php echo $CSS; ?></style>
</head>
<body>
    <h1>Accès autorisé !</h1>
    <h2>Identification numérique de l'utilisateur</h2>
    <p><?php echo $user; ?></p>
    <?php if ($uaiParam != '') { ?>
    <p>Établissement courant : <?php echo $uaiParam; ?></p>
    <?php } ?>

    <h2>Informations complémentaires</h2>
    <ul>
<?php
    foreach ($map as $k => $v) {
        echo "<li>$k : ".implode(', ', $v)."</li>\n";
    }
?>
    </ul>

<h3>Réponse XML :</h3>
<xmp>
<?php echo $xmlcontent; ?>
</xmp>
</body>
</html>
```

**Notes**

Ce fichier d'exemple nécessite que PHP soit installé avec les modules openssl, domxml et la directive allow_url_fopen=On.

## Servlet d'accès exemple en Java

```
package fr.tech.corree.cas;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.URLDecoder;
import java.util.Map;
import java.util.Properties;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLSession;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.jasig.cas.client.authentication.AttributePrincipal;
import org.jasig.cas.client.validation.Assertion;
import org.jasig.cas.client.validation.Cas20ProxyTicketValidator;
import org.jasig.cas.client.validation.TicketValidationException;

public class CASProxySampleServlet extends HttpServlet {
	/** Serial version UID. */
	private static final long serialVersionUID = -266131795934829680 7L;

	// Les différentes URLs de serveurs CAS à compléter
	// au fur à mesure de la connexion à d'autres plate-formes
	private static final Properties casServerUrls = new Properties();
	static {
		casServerUrls.setProperty("atrium-paca",
				"https://www.atrium-paca.fr/connexion");
		casServerUrls.setProperty("recette.atrium-paca",
				"https://recette.atrium-paca.fr/connexion");
		casServerUrls.setProperty("preprod.atrium-paca",
				"https://preprod.atrium-paca.fr/connexion");
	};

	// L'url du service à adapter
	private static final String SERVICE_URL
		= "https://dev.tech.fr/casclientdemo/";

	public void doGet(HttpServletRequest request, HttpServletResponse response)
			throws IOException {
```

```
		final PrintWriter pw = response.getWriter();

		// Pas de ticket => pas d'accès
		String ticket = request.getParameter("ticket");
		if (ticket == null) {
			pw.println("Acces interdit [1] -- pas de ticket");
			return;
		}

		// Pas de paramètre pf, pas d'accès
		String pf = request.getParameter("pf");
		if (pf == null) {
			pw.println("Acces interdit [2] -- pas de paramètre pf");
			return;
		}

		// Identifiant de plate-forme inconnu
		if (!casServerUrls.containsKey(pf)) {
			pw.println("Accès interdit [3] -- paramètre pf inconnu");
			return;
		}

		// On crée le validateur CAS à partir de l'URL de la plate-forme
		String casServerUrlPrefix = casServerUrls.getProperty(pf);
		Cas20ProxyTicketValidator pv
			= new Cas20ProxyTicketValidator(casServerUrlPrefix);
		pv.setAcceptAnyProxy(true);


		try {
			// On récupère l'ensemble des paramètres sauf "ticket"
			String decodedQueryString =
URLDecoder.decode(request.getQueryString(), "UTF-8");
			int idx;
			if ((idx = decodedQueryString.indexOf("&ticket=")) != -1) {
				decodedQueryString = decodedQueryString.substring(0,
idx);
			}
			String serviceURL = SERVICE_URL + '?' + decodedQueryString;

			Assertion a = pv.validate(ticket, serviceURL);
			AttributePrincipal principal = a.getPrincipal();

			pw.println("Accès autorisé");
			pw.println("Identifiant utilisateur = "+principal.getName());
			String uaiParam = request.getParameter("uai");
			if (uaiParam != null && !"null".equals(uaiParam)) {
				pw.println("Établissement courant = " + uaiParam);
			}
			pw.println("Informations complémentaires :");
			for (String key :
((Map<String,String>)principal.getAttributes()).keySet()) {
				pw.println(key+" : "+principal.getAttributes().get(key));
			}

		} catch (TicketValidationException tve) {
```

```
                    pw.println("Accès interdit -- erreur de validation du ticket CAS :"
                            + tve.getMessage());
                }
            }
        }
```

**Note**
Cet exemple nécessite l'utilisation de la librairie CAS Java[18].

[18] https://wiki.jasig.org/display/CASC/CAS+Client+for+Java+3.1