

Documentation de mise en place d'un pont Syrano

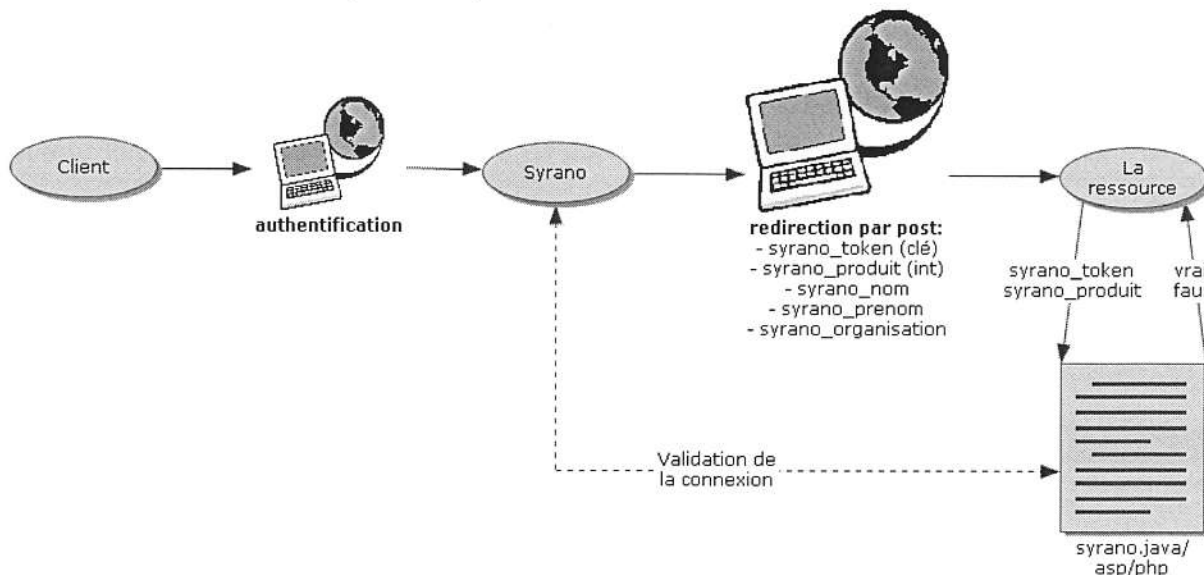
Ce document vise à décrire le processus de mise en place d'un pont SSO entre la plate-forme de gestion des accès de De Marque («Syrano») et la plate-forme de contenu d'un partenaire éditeur. Il est divisé en quatre parties:

1. « Avant de commencer », un résumé du fonctionnement du pont;
2. « Mise en place du pont », des instructions détaillées sur la configuration du pont;
3. « Tester le pont », des instructions pour s'assurer que tout est fonctionnel;
4. « Référence sur les variables Syrano », des informations pour utiliser des fonctionnalités optionnelles du système.

1 Avant de commencer

1.1 Introduction

Le processus de mise en place du pont est résumé dans le schéma suivant:



Dans le schéma, "Syrano" fait référence à la plate-forme d'authentification (Syrano n'est pas un nom commercial, simplement un nom de projet interne, et le "S" n'est pas une coquille... ;-)

1.2 Fonctionnement de Syrano

1.2.1 Envoi par POST et validation de la session

Syrano utilise la méthode POST du protocole HTTP pour envoyer des données à votre application. Nous fournissons une bibliothèque qui contient une fonction pour valider que les informations reçues par votre application sont bel et bien envoyées par le serveur de De Marque (le mécanisme de validation).

1. Compatibilité avec plusieurs types de serveurs

Le mécanisme de validation de session peut être intégré sur tous serveurs Web supportant l'un de ses langages : Java/JSP, PHP, ASP.

1.2.3 Comment fonctionne le mécanisme de validation?

Le mécanisme de validation repose sur la comparaison d'un jeton numérique fourni par la plate-forme appelante (dont on veut valider la légitimité) et d'un jeton hébergé localement, sur votre serveur.

Les paramètres d'entrée de la fonction de validation sont:

- le jeton fourni par la plate-forme appelante;
- votre code privé.

La fonction de validation effectue les actions suivantes:

- récupération du code public (publicHash) sur le serveur de De Marque;
- construction d'un jeton en combinant les codes public et privé;
- comparaison du jeton généré avec celui passé en paramètre d'entrée.

Le paramètre de sortie de la fonction de validation est un booléen qui indique si la plate-forme appelante est légitime (plate-forme de De Marque) ou pas.

1.3 Éléments requis pour mettre en place le pont

Il faut avoir franchi les étapes suivantes pour que la mise en place du pont soit fonctionnelle:

- fournir le URL auquel Syrano doit transmettre les données de connexion (par la méthode POST) sur votre application;
- fournir un identifiant unique pour les produits qui font l'objet d'un pont (un identifiant par produit);
- avoir confirmation que les informations précédentes ont été configurées dans Syrano;
- disposer d'un code privé, fournie par De Marque (voir annexe).

2 Mise en place du pont

Veuillez suivre les étapes suivantes pour mettre en place le pont.

2.1 Installation

Copiez le fichier Syrano, fourni par De Marque, sur votre serveur d'application.

2.2 Création de la page de connexion

Modifiez la page de connexion (qui correspond à l'URL auquel Syrano transmet les données, mentionné à la section 1.3) selon les trois aspects suivants:

Inclure la bibliothèque Syrano.

Exemple php: `require_once('syrano.php');`

Exemple asp: `<!-- #include file="./syrano.asp" -->`

Exemple jsp: `import Syrano;`

Récupérer la variable `syrano_token`, envoyées par la méthode POST à la page de connexion.

Exemple php: `$syrano_token = $_POST['syrano_token'];`

Exemple asp: `syrano_token = request.Form("syrano_token")`

Exemple jsp: `String syrano_token = new String(
request.getParameter("syrano_token"));`

Note: d'autres variables sont envoyées à la page de connexion. Elles sont décrites à la section 4.

Utiliser la fonction de validation de Syrano avant d'ouvrir la session dans votre page de connexion.

La fonction s'appelle `checkSyranoToken()`.

Exemple php: `checkSyranoToken($syrano_token, $codePrive)` // retourne vrai si la
// session est valide

Exemple asp: `CheckSyranoToken(syrano_token, codePrive)` ` retourne vrai si la
` session est valide

Exemple jsp: `Syrano.checkSyranoToken(syrano_token, codePrive)` // retourne vrai si la
// session est valide

Votre fichier doit donc avoir une structure qui ressemble à ceci :

PHP :

```
require_once('syrano.php');  
....  
$syrano_token = $_POST['syrano_token'];  
$codePrive = "2323424" ; // Remplacer par le code privé qui vous aura  
                        // été remis  
if (checkSyranoToken($syrano_token,$codePrive))  
{  
    // inclure ici le nécessaire pour ouvrir la session  
}  
else  
{  
    echo ('Accès refusé');  
}
```

ASP :

```
<!-- #include file="./syrano.asp" -->  
....  
syrano_token = request.Form("syrano_token")  
codePrive = "2323424" ` Remplacer par le code privé qui vous aura  
                    ` été remis  
If (CheckSyranoToken(syrano_token,codePrive)) Then  
    ` inclure ici le nécessaire pour ouvrir la session  
Else  
    response.write("Accès refusé")  
End If
```

JSP :

```
<%
import Syrano;
...
String syrano_token = new String(request.getParameter("syran_token"));
String codePrive = new String("2323424"); // Remplacer par le code privé
// qui vous aura été remis
if (Syrano.checkSyranoToken(syrano_token, codePrive)) {
    // inclure ici le nécessaire pour ouvrir la session
}
else {
    out.print("Accès refusé");
}
%>
```

3 Tester le pont

Une fois les étapes de la section 2 franchies, vous êtes en mesure de tester le pont. Voici comment procéder:

- 3.1 Assurez-vous d'avoir tous les éléments requis à la section 1.3.
- 3.2 Connectez vous à l'URL de test qui vous a été fourni (voir annexe).
- 3.3 Entrez le nom d'utilisateur et le mot de passe de test qui vous a été fourni (voir annexe).
- 3.4 Vous constaterez rapidement si ça fonctionne (connexion au produit) ou pas.

4 Référence - Variables Syrano

Si votre application requiert de connaître plus d'information sur le client qui est connecté par le pont, différentes variables sont disponibles. Le tableau suivant résume les variables disponibles.

Nom de la variable	Description
syrano_token	Jeton numérique qui a une durée de vie de 30 secondes.
syrano_edProdId	Identifiant unique du produit (selon paramètres que vous nous avez fourni).
syrano_typeConnexion	Différencie un client particulier d'un client institutionnel. (0=particulier, [1,2,3,4,...]=institutionnel)
syrano_typeAbo	Différencie un accès de démo [0] d'un accès abonnement [1]
syrano_nom	Nom de l'utilisateur ou du client ¹
syrano_prenom	Prénom de l'utilisateur ou du client ¹
syrano_organisation	Nom de l'organisation (école, bibliothèque, ...)
syrano_idRPC	Identifiant unique de l'utilisateur
syrano_nomProduit	Nom du site ou de l'Application en ligne
syrano_courriel	Courriel de l'utilisateur ou du client ¹
syrano_dateDebut	Date du début d'abonnement à la ressource
syrano_dateFin	Date de fin d'abonnement à la ressource

Exemple php: `$syranos_token = $_POST['syranos_token'];`

Exemple asp: `syranos_token = request.Form("syranos_token")`

Exemple jsp: `String syranos_token = new
String(request.getParameter("syranos_token"));`

¹Dans certain cas, lorsque l'utilisateur appartient à une institution, nous ne connaissons pas les informations propres à l'utilisateur. Dans ces cas, c'est le nom, le prénom et le courriel de la personne-ressource de l'institution qui sont envoyés.