Publik - Project management #6068

Réécriture Mandaye [Éléments de liaison de comptes depuis le portail citoyen]

05 décembre 2014 06:56 - Pierre Cros

Statut:	Fermé	**Début:**	05 décembre 2014
Priorité:	Normal	**Echéance:**	14 décembre 2015
Assigné à:	Josué Kouka	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:		**Club:**	
Planning:			

Description

Demandes liées:

Lié à Mandaye - Development #6779: webservices pour la gestion des liaisons d...	**Fermé**	**19 mars 2015**

Historique

#1 - 05 décembre 2014 06:59 - Pierre Cros

Ne rien prévoir ici pour les agents qui ne verront pas cette page.

#2 - 22 décembre 2014 09:20 - Pierre Cros

- Statut changé de Nouveau à Résolu (à déployer)

J'ai mis les textes dans la page de Vic :
https://dev.entrouvert.org/projects/combo/wiki/Mockup_page_de_bienvenue

Concernant le fait d'utiliser "Mon compte" ou "Profil" ou "Espace personnel" faut harmoniser.

Mon compte : un peu redondant avec le compte citoyen
Profil : un peu technique
Espace personnel : celui qui me plaît le plus sans doute.

Je passe le ticket à resolved mais je ne sais pas qui va se charger de l'implémentation.

#3 - 22 décembre 2014 13:03 - Frédéric Péters

L'implémentation va nécessiter une coordination globale (il sera utile sans doute d'en faire un mail comme je l'avais fait à un moment pour hobo).

Concernant la première étape, la confirmation de l'inscription, peu de questions, ça doit s'intégrer dans le travail fait par Serghei sur la procédure d'inscription (#6087).

Pour le reste, ça a lieu lors de la première connexion au portail citoyen (combo), il faut :

- dans combo la possibilité d'agir sur ce moment de première connexion
- dans combo un bloc de gestion des liaisons
  - ça veut dire que combo doit connaitre d'authentic la liste des liaisons possibles
  - mais il deviendra sans doute utile de faire une distinction entre celles-ci, pour par exemple ne pas avoir une longue liste comprenant tous les wcs communaux…
- l'activation d'une liaison depuis combo :
  - ça veut dire une API dans Authentic qui crée la fédération (sso initié par l'idp dans le cas du saml vers un SP qui ne fait que ça et crée automatiquement les comptes)
  - et qui dans le cas d'un service Mandaye doit fournir de quoi ensuite interroger Mandaye
  - et Mandaye qui doit retourner de quoi créer un formulaire (champs "numéro de famille", "mot de passe")
  - Combo doit présenter le formulaire correspondant, sur une validation passer les données à Mandaye
  - donc webservice mandaye pour prendre les données et les vérifier (tenter une authent sur le service)
- la déliaison depuis combo
  - appel d'un webservice côté authentic pour le faire
  - dans le cas de mandaye, c'est sans doute utile de faire plus que de couper la fédération côté authentic, d'également passer le message à mandaye, qu'il puisse supprimer les credentials de sa db.

#4 - 05 janvier 2015 11:21 - Pierre Cros

- Sujet changé de Rédiger texte pour page login et page d'incription portail citoyen à Éléments de liaison de comptes depuis le portail citoyen

- Echéance changé de 23 décembre 2014 à 31 mars 2015

- Assigné à changé de Pierre Cros à Frédéric Péters

#5 - 06 janvier 2015 17:12 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Nouveau

#6 - 03 mars 2015 16:12 - Frédéric Péters

Côté Authentic, c'est un webservice listant les fédérations (oui/non/créer/supprimer).

#7 - 19 mars 2015 10:34 - Frédéric Péters

Partons de l'idée que les liaisons se font uniquement vers des services déclarés au niveau de Hobo (ça veut dire la possibilité de déclarer des Mandaye).

Pour afficher la liste de on/off :

- Combo interroge authentic pour avoir la liste des fédérations existantes.
- Combo se base alors sur cette liste (en matchant sur le slug donné par hobo, ou sur l'url des metadata) pour afficher des curseurs à on, d'autres à off.

Lorsqu'une demande de liaison avec un service non-mandaye a lieu, il y a :

- appel à authentic pour que celui-ci crée silencieusement la fédération saml avec le service.

Lorsqu'une demande de laison avec un service mandaye a lieu dans combo, il y a :

- GET d'un webservice mandaye qui produit les éléments de formulaire nécessaire; ça peut consister en texte d'intro + liste de champs (libellé et type (string/password)).
- combo affiche ce formulaire et il est complété par l'usager.
- POST vers un webservice mandaye pour créer la liaison (en fournissant le NameID de l'utilisateur, qui est partagé par les différents services); Mandaye valide les données et retourne que c'est ok, ou pas. (et quand c'est ok enregistre l'association NameID/infos).
- pour combo, si c'est ok, on reprend comme plus haut, appel à authentic pour que celui-ci crée silencieusement la fédération saml avec mandaye.

Demande de déliaison :

- si mandaye, POST vers celui-ci pour qu'il oublie les infos.
- POST vers authentic pour supprimer la fédération.

#8 - 19 mars 2015 10:42 - Frédéric Péters

- Projet changé de Produits Entr'ouvert à Publik

#9 - 19 mars 2015 10:47 - Frédéric Péters

- Lié à Development #6779: webservices pour la gestion des liaisons depuis combo ajouté

#10 - 19 mars 2015 19:49 - Benjamin Dauvergne

J'en étais resté à: plus de fédérations, identifiant unique opaque pour tous les services mais jamais affiché, quand est-ce qu'on a changé d'avis ?

#11 - 20 mars 2015 00:03 - Frédéric Péters

On n'a pas changé d'avis, ça doit juste être une histoire de vocabulaire parce que je ne sais pas ce que mandaye peut attendre; quel est exactement ton soucis par rapport à ce que je notais ?

#12 - 20 mars 2015 22:14 - Benjamin Dauvergne

Pendant un instant je me suis dit qu'on pourrait se passer des WS de créaton/suppression de fédération au niveau d'authentic si l'identifiant était globalement unique, mais en fait non donc je retire ma remarque on doit faire comme tu dis pour couvrir tous les cas.

#13 - 20 avril 2015 12:30 - Frédéric Péters

- Sujet changé de Éléments de liaison de comptes depuis le portail citoyen à Réécriture Mandaye [Éléments de liaison de comptes depuis le portail citoyen]

#14 - 20 avril 2015 12:32 - Frédéric Péters

- Echéance changé de 31 mars 2015 à 15 mai 2015

#15 - 11 mai 2015 13:12 - Frédéric Péters

- Echéance changé de 15 mai 2015 à 15 septembre 2015

#17 - 15 septembre 2015 14:53 - Frédéric Péters

- Echéance changé de 15 septembre 2015 à 23 novembre 2015

#18 - 02 novembre 2015 09:50 - Frédéric Péters

- Assigné à changé de Frédéric Péters à Josué Kouka

Lors de l'eoday :

- Josué regarde ça, y Django-mellon à ajouter
- On va s'en servir pour le SSO conservatoire Vincennes (échéance 1 mois , Josué + Mik et Benj)

#19 - 02 novembre 2015 12:54 - Frédéric Péters

Pour permettre le suivi, les trois propositions d'approche notées par Benjamin, elles pourraient être copiées/collées dans ce ticket ?

#20 - 02 novembre 2015 13:01 - Benjamin Dauvergne

Discussion avec Mike et Josué, sans relire le mail initial de Fred, ce qui est une mauvaise idée, on en arrive à l'objectif vouloir tout faire en JS. Voilà le mail initial de Fred sur mandayejs:

```
Yop,

Pour amuser la galerie pendant la réunion tech de demain matin, mais
aussi parce que Dunkerque revient et qu'il y a des adaptations
nécessaires pour la gestion des liaisons depuis Combo, je propose
qu'on évoque Mandaye.

Il a déjà été question dans des discussions passées de maximiser le
javascript et de basculer la gestion sur Django (#4912 par exemple),
et jeudi alors que j'étais parti acheté des bières il m'est venu sans
crier gare que ça pourrait être aussi simple qu'ajouter un chargement
de script js en bout de fichier, et que d'une manière ou d'une autre
il ferait le submit du form de login directement depuis le navigateur,
et ça serait simple et fantastique.

Du coup ce midi j'ai essayé, c'est plutôt facile de laisser Apache
faire le boulot de reverse proxy et qu'il ajoute un appel en bout de
fichiers.

   ExtFilterDefine fixtext mode=output intype=text/html \
      cmd="/bin/sed -e '\$a<script type=\"text/javascript\" src=\"/_mandaye/static/mandaye.js\"></script>'"
   SetOutputFilter fixtext

Et ça semble pareil pour nginx, même si je n'ai pas essayé.

   http://nginx.org/en/docs/http/ngx_http_sub_module.html

Mais voilà, les choses partent bien et puis patatra, un os, ça ne peut
pas marcher, si le submit est fait depuis le navigo il va proposer à
l'usager d'enregistrer le mot de passe, et pas la peine d'essayer
d'empêcher ça, de plus en plus autocomplete="off" est ignoré, pour de
bonnes raisons.

   https://bugzilla.mozilla.org/show_bug.cgi?id=956906

Du coup je suis passé à autre chose, si l'idée du js est bonne, il ne
faut pourtant pas que le login se passe chez l'usager. Ça tombe bien
pour les mini-captures de la page références, j'avais un peu joué avec
PhantomJS, "minimalistic headless WebKit-based with JavaScript API".

   http://git.entrouvert.org/publik-website.git/commit/?id=ea9a92

Ça fonctionne plutôt bien, ça permet de jouer l'affaire ainsi :

   var page = require('webpage').create();
   page.open('https://dev.entrouvert.org/login, function() {
     page.evaluate(function(input) {
       $('#username').val('fred');
       $('#password').val('fred');
       $('#username').parents('form').submit();
     });
   });
```

```
Et comme on a déjà Victor qui développe des grosses applications avec
les workflows de wcs, pourquoi ne pas avoir Pierre qui développe des
connecteurs Mandaye ?

Bref, une fois qu'on a de quoi rejouer de manière fiable un login, on
peut juste extraire les cookies après et les envoyer à l'usager, ni vu
ni connu je t'embrouille, et hop le voilà loggué.

   http://perso.entrouvert.org/~fred/tmp/mandayejs.ogv
   http://git.0d.be/?p=mandayejs.git;a=summary

Oh bien sûr il y a des raccourcis (profiter du fait que redmine
embarque jquery) et des manques (l'absence de logout étant le plus
flagrant), mais au moins ça aura validé certaines pistes.

À demain,

        Frédéric
```

Il en ressort qu'à cause des gestionnaires de mot de passe dans les navigateurs, ce fonctionnement pose problème et donc nécessitera toujours une brique serveur simulant un navigateur pour obtenir un login automatique.

Le nouvel objectif plus raisonnable:

- faire du bout de script autour de phantomjs un module qui irait dans passerelle qu'on puisse appeler pour exécuter un rejeu sur une application configurée
- ce même module pourra produire un script à charger dans la page de l'application (via un injection par un reverse-proxy ou tout bêtement via intervention d'un prestataire sur le thème/template de l'application) qui aura les fonctionnalités suivante:
  - modifier la page, ajouter barre et tutti quanti
  - vérifier l'existence d'une session sur authentic (en accédant à /api/user/ via CORS)
  - lancer un SSO sans protocole si pas de session (juste faire une redirection vers /login?next=<current URL>)
  - afficher une popup demande login/mdp
  - enregistrer/lire ce login mdp dans un web service dans authentic (/api/credentials via CORS)
  - rejouer ce login/mdp via le web-service de rejeu dans passerelle, récupérer le cookie et le poser localement
  - pouvoir détecter qu'une session existe

En plus de désigner l'emplacement des champs login/mdp etc.. la configuration du module passerelle doit permettre:

- d'indiquer les modifications à apporter à HTML
- d'indiquer comment détecter qu'une session existe (peut-être simplement si cookie existe, mais ça peut être plus-compliqué)

Dans un premier temps ça peut simplement être un champ JSON ou directement du javascript, quitte à l'usage à voir ce qu'on peut rendre plus ergonomique.

Il est peut-être plus utile de tout mettre dans passerelle que de couper les choses en deux entre authentic et passerelle. Les éléments de liaison de compte iraient dans combo à terme et feraient partie de la popup d'accueil aussi.

#21 - 02 novembre 2015 13:04 - Benjamin Dauvergne

On peut ajouter à ce module passerelle le fait de configurer un nginx localement qui s'occuper d'injecter le JS, si nécessaire; pour moi dans beaucoup de cas on pourra tout simplement s'en passer et faire passer l'URL vers passerelle directement dans le template de l'application.

#22 - 02 novembre 2015 13:10 - Benjamin Dauvergne

Accéder à /api/user/ via CORS marchera car authentic n'a pas à faire de redirection vers lui même pour savoir si une session est ouvrable. Par contre sur l'appel CORS à un éventuel /api/credentials/ sur passerelle ça va foirer, car redirection vers authentic pour ouverture de session. Donc on garde tout dans authentic, voir on fait tout ça dans authentic et pas dans passerelle.

#23 - 02 novembre 2015 17:05 - Josué Kouka

Simple draft

```
// authentic2_mandaye/models.py
class Service(object):
     url = URLField
     login_page_url = URLField
     username_accessor = CharField
     password_accessor = CharField
     show_login_bar = BooleanField
     etc...

class Credentials(object):
    service = FK(Service)
    user = FK(User)
    username = CharField
```

```
    password = CharField

// authentic2_mandaye/urls.py
mandaye_pattern = pattern('',
url(r'^mandaye/(?P<service_slug>\w+)/mandaye.js$',views.generate_js, name='mandaye-generate-js'),
url(r'^mandaye/(?P<service_slug>\w+)/replay$',views.replay, name='mandaye-replay'),
)

// authentic2_mandaye/views.py

def replay(request, service_slug):
    get service through slug
    get user creds
    check user creds
    if ok:
        generate cookie
        return response
    if not ok:
        return error response

def generate_js(request, service_pk):
    service = get_object_or_404(Service, slug=service_slug)
return render(request, 'mandayejs/mandaye.js', {'service': service})

...

* une vue /api/mandaye/<slug-du-service>/replay
Verbe: POST
Input: {
     'username: ...,   // optionnel si déjà enregistré
     'password': ...   // idem
}
Output { 'err':0, 'data': {
           'cookies':  {
             'name': ...,
             'value': ...,
              ... },
             ...
            }
          }
        }
Erreur:
    { 'err': 1, 'code': 'no-credentials' }
    { 'err': 1, 'code': 'wrong-credentials' }
    etc...

* une vue http://authentic/api/mandaye/<slug-du-service>/mandaye.js
Verbe: GET
Conent-type: text/javascript

var mandaye_config = {
    'user_endpoint': 'http://authentic/api/user/',
    'replay_endpoint': 'http://authentic/api/mandaye/<slug-du-service>/replay',
    'login_url': 'http://authentic/login/',
    'logout_url': 'http://authentic/logout/',
    'show_login_bar': true,
    'is_logged_hint': [
        'cookie=cookiename',
    ],
    'on_logout': {
        'verb': 'POST',
        'url': '/logout',
    }
    'global_loogut': true,
}

function is_logged_on_idp() {
 // test en AJAX CORS de user_endpoint pour voir si on est loggé
}

function is_logged() {
 // détection si une session est ouverte sur l'application (par exemple en vérifiant qu'un cookie
 // existe ou qu'un certain contenu est dans la page
}
```

```
// appeler par le bouton login de la barre
function login() {
    Action du bout: redirect vers <login_url>?next=<current_url>
}

function replay() {
    Appeler ws replay,
    si pas d'erreur poser le/les cookie(s), recharger la page,
    sinon return false
}

function creer_liaison() {
    while not is_logged():
        afficher popup username / password
        replay
}

// appeler par le bouton logout de la barre
function logout() {
    Action: faire les actions de déconnexion (POST ou GET sur une URL /logout, supprimer un cookie, etc.., ) s
i global_logout rediriger sur logout_url sinon faire une reload sur la homepage
}
if (mandaye_config.show_login_bar) {
   // afficher la barre
  if (is_logged_on_idp()) {
       // affiche nom de l'utilisateur
       if (not is_logged()) {
         if (not sessionStorage.replay_already_done) {
           if (not replay()) {
              sessionStorage.replay_already_done = true;
              // on affiche un bouton créer une liaison
           } // si réussi on log l'utilisateur par un setcookie + reload , voir plus haut
         } else {
           // on affiche un bouton créer une liaison
         }
       } else {
         afficher bouton logout

       }
  } else {
      // affiche bouton de login

  }
}
```

#24 - 03 novembre 2015 09:23 - Frédéric Péters

Benjamin Dauvergne a écrit :

> Accéder à /api/user/ via CORS marchera car authentic n'a pas à faire de redirection vers lui même pour savoir si une session est ouvrable. Par contre sur l'appel CORS à un éventuel /api/credentials/ sur passerelle ça va foirer, car redirection vers authentic pour ouverture de session. Donc on garde tout dans authentic, voir on fait tout ça dans authentic et pas dans passerelle.

Je passe vraisemblablement à côté de quelque chose, aveuglé par dix ans de larpe puis mandaye, et ça me préoccupe vraiment.

Dans ma conception, un point fondamental c'est que mandaye doit pouvoir être installé à la périphérie, comme une Red Hat au fond du CDG59, ou une machine au firewalling bizarre chez Ovea. Ça amène pour moi que c'est une application à part entière.

Un autre élément c'est que ça doit tourner sur un nom de domaine propre et donc nécessairement les interactions JS sont facilitées si elles restent sur ce domaine tiers. (je comprends qu'une requête vers le /api/user/ à partir de ce domaine, elle va être soumise aux règles CORS).

À ça j'ajoute aussi un commentaire sur les rythmes de développement et la stabilité, l'impact d'une mise à jour de mandaye est tout différent s'il correspond à une mise à jour d'authentic (à nouveau ici, centralité d'authentic vs périphérie des services mandaye).

Voilà, comme je le notais, ça m'ennuie vraiment de ne pas voir à côté de quoi je passe.

#25 - 03 novembre 2015 12:08 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Benjamin Dauvergne a écrit :
>
> > Accéder à /api/user/ via CORS marchera car authentic n'a pas à faire de redirection vers lui même pour savoir si une session est ouvrable. Par contre sur l'appel CORS à un éventuel /api/credentials/ sur passerelle ça va foirer, car redirection vers authentic pour ouverture de

session. Donc on garde tout dans authentic, voir on fait tout ça dans authentic et pas dans passerelle.

Je passe vraisemblablement à côté de quelque chose, aveuglé par dix ans de larpe puis mandaye, et ça me préoccupe vraiment.

Dans ma conception, un point fondamental c'est que mandaye doit pouvoir être installé à la périphérie, comme une Red Hat au fond du CDG59, ou une machine au firewalling bizarre chez Ovea. Ça amène pour moi que c'est une application à part entière.

Un autre élément c'est que ça doit tourner sur un nom de domaine propre et donc nécessairement les interactions JS sont facilitées si elles restent sur ce domaine tiers. (je comprends qu'une requête vers le /api/user/ à partir de ce domaine, elle va être soumise aux règles CORS).

À ça j'ajoute aussi un commentaire sur les rythmes de développement et la stabilité, l'impact d'une mise à jour de mandaye est tout différent s'il correspond à une mise à jour d'authentic (à nouveau ici, centralité d'authentic vs périphérie des services mandaye).

Voilà, comme je le notais, ça m'ennuie vraiment de ne pas voir à côté de quoi je passe.

Yep j'ai un peu oublié la nécessaire ouverture de flux entre authentic et l'application pour que le service "replay" fonctionne.

Bon rétropédalage, finalement on va tout faire dans le mandayejs actuel, on ajoute django-mellon et on multitenantise avec hobo. On peut simplement se rapprocher du web-service replay tel qu'il est décrit ici et qui est le seul moyen d'avoir des web-services développés ailleurs que dans mandayejs directement, i.e. si possible dans passerelle.

Par contre mandayejs est lui même une application multi-tenant à 2 niveau, en dehors du multi-tenant d'hobo (i.e. dans un tenant mandayejs il y a encore de tenants, les sites mandayisés). Ça me semble compliquer à gérer, je serai pour rendre l'application elle-même "mono-tenant" i.e. ça ne mandayise qu'une application par tenant; à ce titre on a pas vraiment besoin d'un /manage on peut gérer complètement la configuration depuis l'objet class Mandaye(Service) de hobo.

Donc tâches:

- ajouter django-mellon
- finir packaging python et debian (avec nécessaire intégration de hobo)
- développer le modèle Service dans Hobo permettant de configurer l'URL du service, l'URL de la page de login, username_accessor, password_accessor, choses qui seront passés à hobo_deploy,
- créer un hobo_agent spécifique capable de générer la configuration apache ou nginx nécessaire et de les poser dans le /var/lib/mandayejs/nginx.d/ et /var/lib/mandayejs/apache.d/

#26 - 03 novembre 2015 12:09 - Benjamin Dauvergne

Et on verra pour avoir un web-service replay ensuite, ce n'est pas le plus compliqué.

#27 - 03 novembre 2015 12:16 - Frédéric Péters

Et pour assurer Josué sur ce point, tout ce plan me va bien (maintenant qu'on a du multitenant abouti, ça ne sert vraiment à rien de dupliquer le job dans mandaye).

#28 - 19 novembre 2015 01:54 - Josué Kouka

Y'aurait il moyen d'utiliser **casperjs**, qui est basé sur **phantomjs** ?
J'ai rencontré quelques soucis avec **phantomjs** seul (sans casper):

- persitence des sessions malgré un appel de phantomjs avec l'option --cookies-file
- problème de navigation entre plusieurs page, i.e : [stackoverflow](http://stackoverflow.com/questions/30504217/phantomjs-resource-redirection-causes-an-request-error), [github](https://github.com/ariya/phantomjs/issues/12750)

Je viens de faire quelques tests et **casperjs** semble ne pas avoir les memes soucis, le code ressemblerait à ça :

```
system = require('system');

casper = require('casper').create();

input = JSON.parse(system.stdin.read(2000));

casper.start(input.url, function(){
    this.evaluate(function(input){
        if ($(input.username_id).length == 0){
            console.log(JSON.stringify({'result': 'ok'}));
        }

        $(input.username_id).val(input.username_value);
        $(input.password_id).val(input.password_value);
        var submit = $(input.username_id).parents('form').find('input[type=submit]');
        $(submit).click();
        //this.click(submit);
```

```
    }, input);
});

casper.then(function(){
    this.capture('clogin.png');
    this.echo(JSON.stringify({'result': 'ok', 'cookies': this.page.cookies}));
});

if (input.page){
    casper.thenOpen(input.page, function(){
        this.capture('cpage.png');
        this.echo(JSON.stringify({'result': 'ok', 'cookies': this.page.cookies, 'content': this.page.content})
);
    });
}

casper.run(function(){
    this.exit();
});
```

Le gros hic c'est qu'il est encore en beta (le module a été sorti de phantomjs)

#29 - 19 novembre 2015 09:23 - Frédéric Péters

Si tu as chez toi une installation qui fonctionne de bout en bout avec casperjs, très bien, on verra ensuite pour assurer un paquet.

Cependant les deux URL que tu pointes concernent phantomjs 2.0, qui est dans mon souvenir une réécriture, et qui ne correspond pas à la version sur laquelle on s'était dit qu'il fallait bosser, la version qui est actuellement packagée et dans nos dépôts. (1.9.0) Version avec laquelle je n'ai pas rencontré de problème particulier.

Mais je le redis une fois, **si tu as chez toi une installation qui fonctionne de bout en bout avec casperjs, très bien**.

#30 - 23 novembre 2015 12:34 - Pierre Cros

- Echéance changé de 23 novembre 2015 à 14 décembre 2015

#31 - 26 novembre 2015 16:23 - Josué Kouka

- Statut changé de Nouveau à En cours

#32 - 21 décembre 2015 13:13 - Frédéric Péters

- Statut changé de En cours à Fermé