w.c.s. - Development #14510

ajouter une méthode d'authentification FranceConnect

05 janvier 2017 12:05 - Thomas Noël

Statut:	Fermé	**Début:**	05 janvier 2017
Priorité:	Haut	**Echéance:**	14 mars 2017
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

Avoir une nouvelle méthode FranceConnect dans /backoffice/settings/identification/, avec configuration dans /backoffice/settings/identification/fc/ (clés et association des champs user-wcs/user-FC)

Au retour de l'authentification, un user est créé si nécessaire, et ses champs sont remplis en mode "certifiés" (afin que le prefill ne permette pas la modification).

De plus, toutes les informations FC (identité pivot) sont stockées dans la session dans un dico session_var_franceconnect (ou autre).

Cas d'usage : wcs + authentic avec uniquement comptes backoffice (minint), et donc provisionning des comptes FC dans wcs uniquement.

Demandes liées:

Lié à w.c.s. - Bug #15698: Forcer nouveau numéro de session après s'être loggué	**Fermé**	**30 mars 2017**
Lié à w.c.s. - Development #16144: permettre l'utilisation de compute() en de...	**Fermé**	**03 mai 2017**
Lié à w.c.s. - Development #16145: ajouter un attribut extra_user_variables a...	**Fermé**	**03 mai 2017**

Révisions associées

Révision c1a923c2 - 03 mai 2017 16:07 - Thomas Noël

add FranceConnect authentication method (#14510)

use session.extra_user_variables to store attributes retrieved
during FranceConnect SSO.

Historique

#1 - 05 janvier 2017 22:32 - Benjamin Dauvergne

- Fichier 0001-factorize-WorklowStatusItem.compute.patch ajouté

- Fichier 0002-move-wcs.formdata.flatten_dict-into-qommon.misc.patch ajouté

- Fichier 0003-add-france-connect-authentication-method.patch ajouté

- Fichier 0004-add-an-inspect-view-for-global-substitution-variable.patch ajouté

- Statut changé de Nouveau à En cours

- Assigné à mis à Benjamin Dauvergne

- Patch proposed changé de Non à Oui

- patch1: j'ai besoin de compute pour permettre d'éventuelles transformation à ce qui arrive de France Connect, je l'ai déplacé dans qommon.misc (faudra que je reprenne le widget créé inspiré de celui dans wcs.wf.profile dans la configuration SAML 2.
- patch2: idem pour flatten_dict sachant que l'adresse par exemple est un sous-objet dans les données par France Connect, l'esprit dans w.c.s. étant d'aplatir
- patch3: ajout de la méthode d'authentification France Connect
- patch4: création d'une vue /backoffice/inspect, ça ne montre que les variables de substitution globales qu'on voit déjà dans la vue inspect des formdef mais pour mes tests immédiat ça m'a servi et n'a pris aucun temps, mais pas grave si ça n'est pas repris

J'attache aussi deux screenshots: la vue inspect citée plus haut, pas tellement pour elle même mais pour voir les variables disponibles (apparemment la DGIP, au moins en test, donnerait bien l'adresse), et la vue de configuration de France Connect.

À noter que là c'est vite fait en spécifique France Connect mais que ça pourrait très très rapidement devenir une méthode d'authentification OIDC pour migrer w.c.s. en OIDC avec Authentic, il n'y a que la vue de configuration qui changerait et le fait qu'on gérerait plusieurs IdP; peut-être. Là j'ai

fait du spécifique pour simplifier la configuration (pas besoin de connaître les URLs de FC, on choisit directement la plate-forme). Il manque les URLs de prod France Connect, faut que je les retrouve et j'ajouterai ça avant de pousser.

Faut que je complète la liste des variables disponibles via France Connect, comme aide à la définition des règles de mapping, il faudrait aussi donner la correspondance scopes / attributs.

#2 - 05 janvier 2017 22:32 - Benjamin Dauvergne

- Fichier inspect_screenshot.png ajouté

- Fichier france_connect_settings_screenshot.png ajouté

Les screenshots.

#3 - 05 janvier 2017 22:44 - Benjamin Dauvergne

- Fichier 0003-add-france-connect-authentication-method.patch ajouté

URL de prod ajoutée; à noter la déconnexion n'est pas implémentée (mais pour l'instant w.c.s. ne gère pas de méthode de déconnexion en fonction de la méthode d'authentification, c'est juste cablé sur l'implémentation SAML).

À noter que j'ai ajouté un dico fourre tout à l'objet Session (un peu comme une session Django), je ne sais pas trop si c'est bien mais c'est pratique là, surtout pour ne pas reproduire les problèmes de after_url du passé. Dans l'absolu la façon de gérer sauvegarde ou pas des sessions dans w.c.s. est un peu violente comparée à Django (qui elle par contre laisse facilement faire des erreurs); mais c'est un sujet pour un autre ticket.

#4 - 05 janvier 2017 22:45 - Benjamin Dauvergne

C'est testé (manque les tests dans w.c.s. j'avoue).

#5 - 05 janvier 2017 22:47 - Benjamin Dauvergne

Aussi j'ai ajouté une classe large à mes StringWidget et dans admin.css pour que mes champs prennent toute la largeur (pour des URLs ou des client_id/secret c'est vraiment plus pratique) mais si il existe une autre façon de faire je veux bien.

#6 - 06 janvier 2017 10:00 - Frédéric Péters

- 0001
  - évitons les jeux de piste, modifions les appelants dès maintenant.
- 0002
  - ok
- 0003
  - appelons FranceConnect FranceConnect et pas France Connect.
  - KNOWN_ATTRIBUTES, ne contextualisons pas les chaines passées à gettext quand ce n'est pas nécessaire.
  - comme ça peut être très très rapidement fait, plutôt partant pour dès maintenant faire ça sous forme générique oidc, plutôt qu'avoir plus tard à gérer des migrations. (et avoir les URL de FranceConnect en presets possibles, ou pas (en se disant que du w.c.s. connecté directement à FranceConnect, ça ne doit pas être un cas fréquent, Publik passant par Authentic)).
  - pas de CSS spécifique, l'argument était fait que les URL devraient aller dans des champs prenant toute la largeur, ça peut être fait en appliquant ça sur le champ URLWidget, que tout le monde en profite.
  - ok pour le extra dans la session mais comme il y a déjà un extra_variables dedans qui a une signification particulière, il faudrait un autre nom, peut-être juste extra_attributes ?
- 0004
  - peut être discuté mais ça doit être dans un autre ticket.

Bien sûr il faut des tests.

#7 - 06 janvier 2017 14:31 - Benjamin Dauvergne

- Fichier 0001-factorize-WorklowStatusItem.compute.patch ajouté

- Fichier 0003-add-FranceConnect-authentication-method-fixes-14510.patch ajouté

Frédéric Péters a écrit :

> - 0001
>   - évitons les jeux de piste, modifions les appelants dès maintenant.

ok, c'est corrigé, ça passe les tests dans tests/test_workflows.py

> - 0003

- appelons FranceConnect FranceConnect et pas France Connect.

ok.

- KNOWN_ATTRIBUTES, ne contextualisons pas les chaines passées à gettext quand ce n'est pas nécessaire.

ok.

- comme ça peut être très très rapidement fait, plutôt partant pour dès maintenant faire ça sous forme générique oidc, plutôt qu'avoir plus tard à gérer des migrations. (et avoir les URL de FranceConnect en presets possibles, ou pas (en se disant que du w.c.s. connecté directement à FranceConnect, ça ne doit pas être un cas fréquent, Publik passant par Authentic)).

pas chaud, je suis pour garder une méthode spécifique France Connect avec toute l'aide qu'elle fournit et à coté plus tard une méthode OIDC si nécessaire brute de fonderie. Il y aura de plus en plus w.c.s. connecté directement à FranceConnect notamment à Lyon ou ailleurs pour obtenir le revenu fiscal de référence (et vraisemblablement pas comme une authentification mais uniquement en variable de session). À moins qu'on avance un peu sur un protocole unique de pré-remplissage basé sur OAuth2.

- pas de CSS spécifique, l'argument était fait que les URL devraient aller dans des champs prenant toute la largeur, ça peut être fait en appliquant ça sur le champ URLWidget, que tout le monde en profite.

Ça ne corrige pas le problème pour les client_id / client_secret, alors j'ai juste remplacé mon class="large" par des size="80",

- ok pour le extra dans la session mais comme il y a déjà un extra_variables dedans qui a une signification particulière, il faudrait un autre nom, peut-être juste extra_attributes ?

ok.

Patch corrigé attaché.

- 0004
  - peut être discuté mais ça doit être dans un autre ticket.

> Bien sûr il faut des tests.

Sûr.

#8 - 07 janvier 2017 00:36 - Benjamin Dauvergne

- Fichier 0003-add-FranceConnect-authentication-method-fixes-14510.patch ajouté

Et hop un test de login FC.

#9 - 07 janvier 2017 10:45 - Frédéric Péters

- 0001

  ok, c'est corrigé, ça passe les tests dans tests/test_workflows.py

Attention besoin aussi d'un patch pour auquo.

Et le patch ne s'applique pas sur master (4c53edd a modifié une ligne d'import).

- 0003

  pas chaud, je suis pour garder une méthode spécifique France Connect avec toute l'aide qu'elle fournit et à coté plus tard une méthode OIDC si nécessaire brute de fonderie.

Comme tu as écrit que ça pouvait être fait très très rapidement, je trouvais dommage de ne pas profiter de l'occasion, ce moment où le plan est clair. Mais si en fait ce n'est pas si rapide et/ou que tu n'as pas envie, tant pis.

> Ça ne corrige pas le problème pour les client_id / client_secret, alors j'ai juste remplacé mon class="large" par des size="80",

Je parlais d'URL parce que tu parlais d'URL, et donc là quand même, pour les URL, utiliser l'UrlWidget.

- Tests.

Il s'appelle test_saml_login_page. Il faudrait aussi des tests pour la partie UI de l'admin. De manière générale ma "règle" c'est que le nouveau code ait au moins autant de couverture par les tests que le reste de w.c.s. (aujourd'hui 80%) et le nouveau france_connect.py (tiens, renommons le aussi franceconnect.py, que disparaisse l'idée de l'espace) en local je lui compte seulement 61%.

#10 - 07 janvier 2017 19:08 - Benjamin Dauvergne

- Fichier 0001-use-misc.compute-instead-of-kflowStatusItem.compute.patch ajouté

- Fichier 0001-factorize-WorklowStatusItem.compute.patch ajouté

- Fichier 0003-add-FranceConnect-authentication-method-fixes-14510.patch ajouté

Frédéric Péters a écrit :

> - 0001
>
>   ok, c'est corrigé, ça passe les tests dans tests/test_workflows.py
>
> Attention besoin aussi d'un patch pour auquo.

Ok.

> Et le patch ne s'applique pas sur master (4c53edd a modifié une ligne d'import).

J'ai fait un rebase, je n'ai pas vu de conflit, je réattache quand même le patch au cas où.

> - 0003
>
>   pas chaud, je suis pour garder une méthode spécifique France Connect avec toute l'aide qu'elle fournit et à coté plus tard une méthode OIDC si nécessaire brute de fonderie.
>
> Comme tu as écrit que ça pouvait être fait très très rapidement, je trouvais dommage de ne pas profiter de l'occasion, ce moment où le plan est clair. Mais si en fait ce n'est pas si rapide et/ou que tu n'as pas envie, tant pis.
>
> > Ça ne corrige pas le problème pour les client_id / client_secret, alors j'ai juste remplacé mon class="large" par des size="80",
>
> Je parlais d'URL parce que tu parlais d'URL, et donc là quand même, pour les URL, utiliser l'UrlWidget.

De fait il n'y a plus d'URL, puisqu'elles sont en dur maintenant mais je prends note.

> - Tests.
>
> Il s'appelle test_saml_login_page. Il faudrait aussi des tests pour la partie UI de l'admin. De manière générale ma "règle" c'est que le nouveau code ait au moins autant de couverture par les tests que le reste de w.c.s. (aujourd'hui 80%) et le nouveau france_connect.py (tiens, renommons le aussi franceconnect.py, que disparaisse l'idée de l'espace) en local je lui compte seulement 61%.

- renommer france_connect.py en franceconnect.py
- renommer en test_fc_login_page
- ajouter test_fc_settings, montant la couverture à 87%, non couvert principalement les cas d'erreur en provenance de France Connect, et les bizarreries du style field_email ou field_name non définis).

#11 - 07 janvier 2017 19:20 - Frédéric Péters

> J'ai fait un rebase, je n'ai pas vu de conflit, je réattache quand même le patch au cas où.

Désolé c'était 0002.

> setup_profile_environement

Typo.

Le renommage en franceconnect.py a fait disparaitre ce fichier du patch.

#12 - 07 janvier 2017 20:59 - Benjamin Dauvergne

- Fichier 0001-factorize-WorklowStatusItem.compute.patch ajouté

- Fichier 0002-move-wcs.formdata.flatten_dict-into-qommon.misc.patch ajouté

- Fichier 0003-add-FranceConnect-authentication-method-fixes-14510.patch ajouté

- Fichier 0004-add-an-inspect-view-for-global-substitution-variable.patch ajouté

Frédéric Péters a écrit :

> J'ai fait un rebase, je n'ai pas vu de conflit, je réattache quand même le patch au cas où.

Désolé c'était 0002.

Pas vu plus de conflits mais je ré-attache toute la série au cas où.

> setup_profile_environement

Typo.

Renommé en setup_user_profile plus clair.

> Le renommage en franceconnect.py a fait disparaitre ce fichier du patch.

Damned, ré-ajouté.

#13 - 13 janvier 2017 11:02 - Thomas Noël

Testé sur ma machine, montré au client. Ça a l'air pas mal, mais je m'inquiète beaucoup de la création d'un compte wcs-only, qui est complétement inutile pour le client ciblé. Les variables dans la session sont parfaitement suffisantes pour leur usage.

Sans compte, ce qui pourrait éventuellement manquer c'est le mode "readonly" pour les données pré-remplies. D'ailleurs sur ce sujet, avec ce patch, c'est plutôt le bazar (faire comprendre que le mode "readonly" ne marche que pour les comptes franceconnectés et que la config des champs bloqués se fait dans les settings de franceconnect, j'ai eu du mal et j'ai vu que ça paraissait bien tordu au client, et surtout peu souple).

Bref, je ne suis finalement pas convaincu par cette approche "méthode d'authentification".

Qui plus est quand le client voudra vraiment passer en mode Publik pour proposer un portail usager... aïe.

#14 - 13 janvier 2017 11:18 - Frédéric Péters

> Qui plus est quand le client voudra vraiment passer en mode Publik pour proposer un portail usager... aïe.

Oui, utilisons dès maintenant Authentic plutôt qu'ajouter des trucs d'authent dans w.c.s.

#15 - 13 janvier 2017 11:32 - Frédéric Péters

> je m'inquiète beaucoup de la création d'un compte wcs-only

On peut voir pour fournir sur le côté des instructions pour faire un update form_xxx set user_id = null; + delete from users;.

> D'ailleurs sur ce sujet, avec ce patch, c'est plutôt le bazar (faire comprendre que le mode "readonly" ne marche que pour les comptes franceconnectés et que la config des champs bloqués se fait dans les settings de franceconnect, j'ai eu du mal et j'ai vu que ça paraissait bien tordu au client, et surtout peu souple).

Je ne comprends pas trop; on pourrait retirer toute config et que les attributs de FranceConnect soient tous toujours bloqués, ça rendrait les choses plus claires ?

La souplesse attendue ce serait pour tel formulaire avoir du préremplissage bloqué et pour tel autre avoir du préremplissage éditable ? (ça se bricolerait aujourd'hui avec du préremplissage python pour le mode éditable; alternativement, si c'est ça la souplesse cherchée, on pourrait peut-être dupliquer le type de préremplissage et avoir "Profil" et "Profil sécurisé" ?)

#16 - 13 janvier 2017 11:39 - Thomas Noël

Frédéric Péters a écrit :

> Je ne comprends pas trop; on pourrait retirer toute config et que les attributs de FranceConnect soient tous toujours bloqués, ça rendrait les choses plus claires ?

Ouaip.

> La souplesse attendue ce serait pour tel formulaire avoir du préremplissage bloqué et pour tel autre avoir du préremplissage éditable ? (ça se bricolerait aujourd'hui avec du préremplissage python pour le mode éditable; alternativement, si c'est ça la souplesse cherchée, on pourrait peut-être dupliquer le type de préremplissage et avoir "Profil" et "Profil sécurisé" ?)

En fait la souplesse ça serait qu'on puisse aller un tout petit peu plus loin dans la notion de préremplissage, enfin disons que les choses soient explicites : avoir un préremplissage qui devient bloquant parce qu'un attribut est vérifié, c'est pas facile à expliquer.

Si on avait une case à cocher au niveau du champ « [ ] S'il y a pre-remplissage alors c'est en fait un remplissage (read-only) », ça pourrait retirer de la magie, et permettre du pré-remplissage habituel (non bloqué) sur un attribut vérifié sans avoir à passer par du Python.

(je vais pas le sortir "explicit is better than implicit", mais tu me suis)

#17 - 13 janvier 2017 11:56 - Frédéric Péters

> > Je ne comprends pas trop; on pourrait retirer toute config et que les attributs de FranceConnect soient tous toujours bloqués, ça rendrait les choses plus claires ?
>
> Ouaip.

Ça me va tout à fait que ça ne soit pas paramétrable.

> > La souplesse attendue ce serait pour tel formulaire avoir du préremplissage bloqué et pour tel autre avoir du préremplissage éditable ? (ça se bricolerait aujourd'hui avec du préremplissage python pour le mode éditable; alternativement, si c'est ça la souplesse cherchée, on pourrait peut-être dupliquer le type de préremplissage et avoir "Profil" et "Profil sécurisé" ?)
>
> En fait la souplesse ça serait qu'on puisse aller un tout petit peu plus loin dans la notion de préremplissage, enfin disons que les choses soient explicites : avoir un préremplissage qui devient bloquant parce qu'un attribut est vérifié, c'est pas facile à expliquer.
>
> Si on avait une case à cocher au niveau du champ « [ ] S'il y a pre-remplissage alors c'est en fait un remplissage (read-only) », ça pourrait retirer de la magie, et permettre du pré-remplissage habituel (non bloqué) sur un attribut vérifié sans avoir à passer par du Python.

Peut-être, dans un autre ticket en priorité basse ? (il y a déjà eu par ailleurs l'ajout de classe css "readonly" pour faire apparaitre un champ comme réel mais non éditable, ça rejoindrait ça, de manière propre).

#18 - 17 février 2017 11:03 - Benjamin Dauvergne

- Fichier 0001-factorize-WorklowStatusItem.compute.patch ajouté

- Fichier 0002-move-wcs.formdata.flatten_dict-into-qommon.misc.patch ajouté

- Fichier 0003-add-FranceConnect-authentication-method-fixes-14510.patch ajouté

- Fichier 0004-add-an-inspect-view-for-global-substitution-variable.patch ajouté

Rebasé sur master.

#19 - 15 mars 2017 01:58 - Thomas Noël

- Echéance mis à 14 mars 2017

- Priorité changé de Normal à Haut

#20 - 15 mars 2017 23:13 - Frédéric Péters

Tout le code de workflow a été modifié pour faire suite à :

```
-    @classmethod
-    def compute(cls, var, do_ezt=True, raises=False):
```

Mais il y a aussi de l'appel à self.compute() côté auquotidien; je serais pour garder la méthode compute, par compatibilité avec le code existant; voire

même rien du tout, et quand il y a besoin du compute ailleurs, que ça soit le code à l'endroit actuel qui soit appelé, après tout c'est une méthode de classe, ça passera.

> 0004-add-an-inspect-view-for-global-substitution-variable.patch

Déjà noté que sa place n'était pas dans ce ticket.

```
            methods.insert(0,
                    ('fc', _('Delegated to France Connect'), 'fc'))
```

FranceConnect. Mais je ne sais pas pourquoi ça se fait dans un bloc "if lasso is not None".

#21 - 16 mars 2017 16:26 - Benjamin Dauvergne

- Fichier 0001-factorize-WorklowStatusItem.compute-.-14510.patch ajouté

- Fichier 0002-replace-use-of-self.compute-.-by-misc.compute-.-1451.patch ajouté

- Fichier 0003-move-wcs.formdata.flatten_dict-into-qommon.misc-1451.patch ajouté

- Fichier 0004-add-FranceConnect-authentication-method-14510.patch ajouté

J'ai remis WorflowStatusItem.compute() qui appelle misc.compute().

Ensuite j'ai séparé le commit en deux on peut ne pas appliquer le deuxième, celui qui remplace les usages de self.compute() par misc.compute().

Enfin j'ai corrigé l'ajout de la méthode d'authentifcation dans les choix, ce n'est plus contrôlé par un if lasso.

#22 - 28 mars 2017 09:29 - Frédéric Péters

> J'ai remis WorflowStatusItem.compute() qui appelle misc.compute().

Donc le 0002-... n'est plus nécessaire, right ? Je préférerais du coup sans.

```
        # XXX: FranceConnect website also refer to adress and phones attributes but we don't know
        # what must be expected of their value.
```

Le XXX pourrait ici être remplacé par "Note:".

```
        # XXX: should we fail login if user info is not sufficient as with SAML 2.0 login ?
        # XXX: should we implement an authentication less mode, it would just store FC user informations into
the session variables
```

Les XXX ici ne demandent-ils pas des réponses ?

(attention concertation avec Thomas à avoir, il me dit travailler sur le sujet aujourd'hui)

#23 - 28 mars 2017 09:44 - Thomas Noël

- Fichier 0001-add-FranceConnect-authentication-method-14510.patch ajouté

Une version simplifiée qui ne change pas les places de compute ou flatten_dict (mais juste ajoute la possibilité d'un context à compute).

Avec aussi une correction du « self.extra_attributes = self.extra_attributes = {} » (dans wcs.qommon.sessions)

Et le XXX qui devient Note pour "adress and phones attributes"

#24 - 28 mars 2017 09:47 - Thomas Noël

Frédéric Péters a écrit :

```
        # XXX: should we fail login if user info is not sufficient as with SAML 2.0 login ?
        # XXX: should we implement an authentication less mode, it would just store FC user informations into
the session variables
```

> Les XXX ici ne demandent-ils pas des réponses ?

- should we fail login if user info is not sufficient as with SAML 2.0 login ?

Ca voudrait dire quoi ici , "is not sufficient" ?

- should we implement an authentication less mode, it would just store FC user informations into the session variables

Ca m'irait bien, mais alors le reste devient incohérent à mon avis (FranceConnect ne serait plus une méthode d'authent au sens attendu dans wcs, où on provisionne un compte). Donc non, ça me semble pas jouable, là, maintenant.

#25 - 28 mars 2017 10:27 - Frédéric Péters

> Ca voudrait dire quoi ici , "is not sufficient" ?

Quand on n'a pas le nom ou pas l'email on crée pas de compte côté SAML.

#26 - 28 mars 2017 11:19 - Benjamin Dauvergne

Thomas Noël a écrit :

> - should we implement an authentication less mode, it would just store FC user informations into the session variables
>
> Ca m'irait bien, mais alors le reste devient incohérent à mon avis (FranceConnect ne serait plus une méthode d'authent au sens attendu dans wcs, où on provisionne un compte). Donc non, ça me semble pas jouable, là, maintenant.

C'est comme tu veux, c'est ton projet le minint, si ça te va comme cela, go.

#27 - 28 mars 2017 16:20 - Thomas Noël

- Fichier 0001-add-FranceConnect-authentication-method-14510.patch ajouté

J'ai donc ajouté un rejet des FranceConnect sans nom ou sans mail (ça devrait juste absolument jamais arriver) :

```
diff --git a/wcs/qommon/ident/franceconnect.py b/wcs/qommon/ident/franceconnect.py
index 9f1cc49e..b0ddbd07 100644
--- a/wcs/qommon/ident/franceconnect.py
+++ b/wcs/qommon/ident/franceconnect.py
@@ -446,9 +446,15 @@ class FCAuthMethod(AuthMethod):
             user = pub.user_class(sub)
             user.name_identifiers = [sub]

-        # XXX: should we fail login if user info is not sufficient as with SAML 2.0 login ?
-        # XXX: should we implement an authentication less mode, it would just store FC user informations into
 the session variables
         self.fill_user_attributes(user, user_info)
+
+        if not (user.name and user.email):
+            # we didn't get useful attributes, forget it.
+            logger.error('failed to get name and/or email attribute from FranceConnect')
+            session.message = ('error',
+                               _('FranceConnect authentication failed: missing name or email'))
+            return redirect(next_url)
+
         user.store()
         session.set_user(user.id)
         return redirect(next_url)
```

#28 - 29 mars 2017 00:10 - Benjamin Dauvergne

Ok, si tu peux juste ajouter un test qui vérifie que ça marche justement pas sans email ou sans nom.

#29 - 29 mars 2017 15:55 - Thomas Noël

- Fichier 0001-add-FranceConnect-authentication-method-14510.patch ajouté

Lors de l'ajout de ce texte, deux découvertes :

- faux-positif dans le test initial qui cherche à ré-utiliser un usager existant : en fait ça foire bien avant, le gars n'est pas loggué, parce qu'il faut reconstruire nonce, tokens, state après app.get('/login/'):

```
@@ -160,6 +160,13 @@ def test_fc_login_page():
     # Login existing user
```

```
     resp = app.get('/logout')
     resp = app.get('/login/')
+    assert resp.status_int == 302
+    assert resp.location.startswith('https://fcp.integ01.dev-franceconnect.fr/api/v1/authorize')
+    qs = urlparse.parse_qs(resp.location.split('?')[1])
+    state = qs['state'][0]
+    id_token['nonce'] = qs['nonce'][0]
+    token_result['id_token'] = '.%s.' % base64url_encode(json.dumps(id_token))
```

- si on utilise [email] et qu'un email n'est pas fourni, alors la valeur après compute est "[email]". ezt fait comme ça, il ne touche pas aux variables qui n'existent pas. Donc faut indiquer [email ""] ou [if-any email][email][end], c'est un poil relou, oui oui.

```
--- a/tests/test_fc_auth.py
+++ b/tests/test_fc_auth.py
@@ -77,17 +77,17 @@ FC_CONFIG = {
     'user_field_mappings': [
         {
             'field_varname': 'prenoms',
-            'value': '[given_name]',
+            'value': '[given_name ""]',
             'verified': 'always',
         },
...
```

#30 - 29 mars 2017 16:11 - Frédéric Péters

> si on utilise [email] et qu'un email n'est pas fourni, alors la valeur après compute est "[email]". ezt fait comme ça, il ne touche pas aux variables qui n'existent pas. Donc faut indiquer [email ""] ou [if-any email][email][end], c'est un poil relou, oui oui

(notre modif à ezt parce que de base ezt lève une exception, cf #1270).

Amené par cette note, quand même, je note que la config de FranceConnect du coup elle demande à l'admin d'aller remplir un mapping des champs, de lire quelque part qu'il faut taper prenoms → [given_name]. Je trouve ça plutôt pas terrible mais comme on n'utilisera pas ce code, je peux laisser tomber.

Et encore, dans les tests,

```
pub.cfg['users']['field_name'] = ['_prenoms', '_nom']
```

J'imagine que ça échoue, ça devrait être ['prenoms', 'nom'] vu que plus haut dans PROFILE c'est prenoms et nom les champs.

```
methods.insert(0, ('fc', _('Delegated to France Connect'), 'fc'))
```

J'avais déjà noté que "France Connect" s'écrivait "FranceConnect", reste encore ce bout à corriger.

#31 - 29 mars 2017 16:45 - Thomas Noël

- Fichier 0001-add-FranceConnect-authentication-method-14510.patch ajouté

Frédéric Péters a écrit :

> Amené par cette note, quand même, je note que la config de FranceConnect du coup elle demande à l'admin d'aller remplir un mapping des champs, de lire quelque part qu'il faut taper prenoms → [given_name]. Je trouve ça plutôt pas terrible mais comme on n'utilisera pas ce code, je peux laisser tomber.

Si la config n'est pas faite on pourrait se lancer dans un mapping en utilisant les noms "classiques". Mais là, bon, je fatigue un peu.

> Et encore, dans les tests, (...)
> J'imagine que ça échoue, ça devrait être ['prenoms', 'nom'] vu que plus haut dans PROFILE c'est prenoms et nom les champs.

Non parce que via check_hobo (update_profile) ce sont bien des « _foobar » qui servent de clé dans le formdef du profil utilisateur.

> J'avais déjà noté que "France Connect" s'écrivait "FranceConnect", reste encore ce bout à corriger.

Hop.

J'ai renforcé les tests en testant les messages posés dans la session par le callback, genre :

```
+    session_id = app.cookies.values()[0].strip('"')
+    session = get_session_manager().session_class.get(session_id)
+    assert session.user is None
+    assert 'FranceConnect authentication failed: missing name or email' in str(session.display_message())
```

#32 - 29 mars 2017 17:01 - Frédéric Péters

> Non parce que via check_hobo (update_profile) ce sont bien des « _foobar » qui servent de clé dans le formdef du profil utilisateur.

Ok j'avais zappé que les tests passaient par hobo.

Les tests ne passent pas sur le cas où le profil est absent, i.e. ces lignes :

```
        if not user_formdef or not users_cfg.get('field_name'):
            fields.append(('__name', _('Name'), '__name'))
        if not user_formdef or not users_cfg.get('field_email'):
            fields.append(('__email', _('Email'), '__email'))
```

(et les autres plus loin faisant référence à ces double underscores)

Ne passent pas non plus sur la gestion d'erreur, le code dans cette branche :

```
        if 'code' not in request.form:
```

Je peux laisser passer, par fatigue, les tests sur l'absence de profil, mais les retours d'authentification ratée, ça m'irait quand même vraiment bien qu'ils soient testés.

#33 - 29 mars 2017 19:28 - Benjamin Dauvergne

J'ai fait ce surpatch qui teste les éléments signalés par Fred mais:

- session.message n'est pas vraiment utilisable, il n'est pas affiché par la homepage, ça interagit avec la gestion foireuse des sessions (dès qu'un truc est stocké en session la session survit), j'ai donc viré tous les session.message = ('error', ...), je teste les logs directement avec la fixture caplog
- pour parer à ça j'ai ajouté un session.id = None explicite dans le code du callback OIDC, je me dis qu'on devrait faire ça aussi en SAML et en login/mdp (sinon on est vulnérable à une attaque par fixation de session, exemple: je vais sur un poste public, je fais un ?session_var_que_je_connais=xxx (ou n'importe quoi d'autre qui stocke un truc en session), je note l'id de la session, j'attends que quelqu'un se log sur ce poste, je vais sur un autre, je pose le cookie que j'ai noté, magie je suis dans la session de la personne).

```
diff --git a/tests/test_fc_auth.py b/tests/test_fc_auth.py
index 5664b1d..c34edb7 100644
--- a/tests/test_fc_auth.py
+++ b/tests/test_fc_auth.py
@@ -66,6 +66,7 @@ def setup_user_profile(pub):
     # setup an hobo profile
     CmdCheckHobos().update_profile(PROFILE, pub)
     pub.cfg['users']['field_name'] = ['_prenoms', '_nom']
+    pub.cfg['debug'] = {'logger': True}
     pub.user_class.wipe()
     pub.write_cfg()

@@ -106,7 +107,12 @@ def setup_fc_environment(pub):
     pub.write_cfg()

-def test_fc_login_page():
+def get_session(app):
+    session_id = app.cookies.values()[0].strip('"')
+    return get_session_manager().session_class.get(session_id)
+
+
+def test_fc_login_page(caplog):
     setup_user_profile(pub)
     setup_fc_environment(pub)
     app = get_app(pub)
@@ -149,17 +155,28 @@ def test_fc_login_page():
     assert user.name == 'John Doe'

     # Verify we are logged in
-    session_id = app.cookies.values()[0].strip('"')
```

```
-    session = get_session_manager().session_class.get(session_id)
+    session = get_session(app)
     assert session.user == user.id
     assert session.extra_variables['fc_user_given_name'] == 'John'
     assert session.extra_variables['fc_user_family_name'] == 'Doe'
     assert session.extra_variables['fc_user_email'] == 'john.doe@example.com'
     assert session.extra_variables['fc_user_sub'] == 'ymca'
-    assert session.display_message() == ''

-    # Login existing user
     resp = app.get('/logout')
+
+    # Test error handling path
+    resp = app.get('/ident/fc/callback?%s' % urllib.urlencode({
+        'state': state,
+        'error': 'access_denied',
+    }))
+    assert 'user did not authorize login' in caplog.records[-1].message
+    resp = app.get('/ident/fc/callback?%s' % urllib.urlencode({
+        'state': state,
+        'error': 'whatever',
+    }))
+    assert 'whatever' in caplog.records[-1].message
+
+    # Login existing user
     resp = app.get('/login/')
     assert resp.status_int == 302
     assert resp.location.startswith('https://fcp.integ01.dev-franceconnect.fr/api/v1/authorize')
@@ -175,14 +192,10 @@ def test_fc_login_page():
         resp = app.get('/ident/fc/callback?%s' % urllib.urlencode({
             'code': '1234', 'state': state,
         }))
-    new_session_id = app.cookies.values()[0].strip('"')
-    assert session_id != new_session_id, 'no new session created'
-    session = get_session_manager().session_class.get(new_session_id)
+    new_session = get_session(app)
+    assert session.id != new_session.id, 'no new session created'
     assert pub.user_class.count() == 1, 'existing user has not been used'
-    session_id = app.cookies.values()[0].strip('"')
-    session = get_session_manager().session_class.get(session_id)
-    assert session.user == user.id
-    assert session.display_message() == ''
+    assert new_session.user == user.id

     # User with missing attributes
     resp = app.get('/logout')
@@ -207,8 +220,7 @@ def test_fc_login_page():
             'code': '1234', 'state': state,
         }))
     assert pub.user_class.count() == 1, 'an invalid user (no email) has been created'
-    session_id = app.cookies.values()[0].strip('"')
-    session = get_session_manager().session_class.get(session_id)
+    session = get_session(app)
     assert session.user is None
     assert 'FranceConnect authentication failed: missing name or email' in str(session.display_message())

@@ -242,3 +254,53 @@ def test_fc_settings():

     resp = resp.forms[0].submit('submit').follow()
     assert pub.cfg['fc'] == FC_CONFIG
+
+
+def test_fc_settings_no_user_profile():
+    FC_CONFIG = {
+        'client_id': '123',
+        'client_secret': 'xyz',
+        'platform': 'dev-particulier',
+        'scopes': 'identite_pivot',
+        'user_field_mappings': [
+            {
+                'field_varname': '__name',
+                'value': '[given_name ""] [family_name ""]',
+                'verified': 'always',
+            },
+            {
```

```
+                'field_varname': '__email',
+                'value': '[email ""]',
+                'verified': 'always',
+            },
+        ]
+
+    }
+
+    pub.cfg = {}
+    pub.user_class.wipe()
+    pub.write_cfg()
+    app = get_app(pub)
+    resp = app.get('/backoffice/settings/identification/')
+    resp.forms[0]['methods$elementfc'].checked = True
+    resp = resp.forms[0].submit().follow()
+
+    assert 'FranceConnect' in resp.body
+    resp = resp.click('FranceConnect')
+    resp = resp.forms[0].submit('user_field_mappings$add_element')
+    resp = resp.forms[0].submit('user_field_mappings$add_element')
+    resp.forms[0]['client_id'].value = '123'
+    resp.forms[0]['client_secret'].value = 'xyz'
+    resp.forms[0]['platform'].value = 'Development citizens'
+    resp.forms[0]['scopes'].value = 'identite_pivot'
+
+    resp.forms[0]['user_field_mappings$element0$field_varname'] = '__name'
+    resp.forms[0]['user_field_mappings$element0$value'] = '[given_name ""] [family_name ""]'
+    resp.forms[0]['user_field_mappings$element0$verified'] = 'Always'
+
+    resp.forms[0]['user_field_mappings$element2$field_varname'] = '__email'
+    resp.forms[0]['user_field_mappings$element2$value'] = '[email ""]'
+    resp.forms[0]['user_field_mappings$element2$verified'] = 'Always'
+
+    resp = resp.forms[0].submit('submit').follow()
+    assert pub.cfg['fc'] == FC_CONFIG
diff --git a/wcs/qommon/ident/franceconnect.py b/wcs/qommon/ident/franceconnect.py
index b0ddbd0..23e3461 100644
--- a/wcs/qommon/ident/franceconnect.py
+++ b/wcs/qommon/ident/franceconnect.py
@@ -398,7 +398,7 @@ class FCAuthMethod(AuthMethod):
             user.set_attributes_from_formdata(user.form_data)

     AUTHORIZATION_REQUEST_ERRORS = {
-        'access_denied': N_('You refused the connection'),
+        'access_denied': N_('user did not authorize login'),
     }

     def callback(self):
@@ -416,12 +416,8 @@ class FCAuthMethod(AuthMethod):
             # if no error parameter, we stay silent
             if error:
                 # we log only errors whose user is not responsible
-                msg = _(self.AUTHORIZATION_REQUEST_ERRORS.get(error))
-                if not msg:
-                    msg = _('FranceConnect authentication failed')
-                    logger.error('FranceConnect authentication failed with an unknown error: %s',
-                                 error)
-                session.message = ('error', msg)
+                msg = self.AUTHORIZATION_REQUEST_ERRORS.get(error)
+                logger.error(_('FranceConnect authentication failed : %s'), _(msg) if msg else error)
             return redirect(next_url)
         access_token = self.get_access_token(request.form['code'])
         if not access_token:
@@ -451,10 +447,9 @@ class FCAuthMethod(AuthMethod):
         if not (user.name and user.email):
             # we didn't get useful attributes, forget it.
             logger.error('failed to get name and/or email attribute from FranceConnect')
-            session.message = ('error',
-                               _('FranceConnect authentication failed: missing name or email'))
             return redirect(next_url)

         user.store()
         session.set_user(user.id)
+        session.id = None
         return redirect(next_url)
```

#34 - 29 mars 2017 19:29 - Benjamin Dauvergne

- Fichier 0001-tomerge.patch ajouté

Le format-patch.

#37 - 30 mars 2017 10:36 - Frédéric Péters

Et l'id de la session change mais son contenu reste présent, ce qui est nécessaire pour les ?session_var_whatever.

Par contre, au moment où on remplit la session avec les variables de FranceConnect,

```
        session_var_fc_user = {}
        for key in flattened_user_info:
            session_var_fc_user['fc_user_' + key] = flattened_user_info[key]
        session.add_extra_variables(**session_var_fc_user)
```

il serait peut-être judicieux de commencer par virer toutes les clés 'fc_user_' qui seraient déjà présentes.

Et de manière plus générale, il y a peut-être à diviser extra_variables, avoir extra_variables et extra_user_variables, que le second soit systématiquement vidé, lors d'une déconnexion ou d'une nouvelle connexion.

#38 - 30 mars 2017 14:20 - Frédéric Péters

- Lié à Bug #15698: Forcer nouveau numéro de session après s'être loggué ajouté

#39 - 30 mars 2017 22:42 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Et l'id de la session change mais son contenu reste présent, ce qui est nécessaire pour les ?session_var_whatever.
>
> Par contre, au moment où on remplit la session avec les variables de FranceConnect,
>
> [...]
>
> il serait peut-être judicieux de commencer par virer toutes les clés 'fc_user_' qui seraient déjà présentes.

Je penche plus pour un nettoyage de extra_user_variables dans Session.set_user(), de manière à y trouver tout ce qu'on espère d'une nouvelle connexion.

> Et de manière plus générale, il y a peut-être à diviser extra_variables, avoir extra_variables et extra_user_variables, que le second soit systématiquement vidé, lors d'une déconnexion ou d'une nouvelle connexion.

Lors d'une déconnexion normalement la session est détruite et request.session mis à None (donc pas de maintain_session()), ça devrait suffire niveau ménage, mais un accès concurrent est fait possible que la session survive (je commence la requête de logout, une autre requête dans un autre thread navigo commence, la requête de logout finit, puis la deuxième requête refait un maintain_session() et bim la session revient, la seule solution je pense ce serait d'avoir un is_dirty() plus fin qu'un alias à has_info() comme les sessions Django qui ne sont dirty que si il y a eu écriture d'une propriété de l'objet, mais probable que ça casse pas mal de trucs de passer sur ce mode, ce qui est gênant c'est que le système de session génère tout de me beaucoup d'écritures sur disque en l'état).

#40 - 30 mars 2017 23:01 - Benjamin Dauvergne

Donc nouveau diff:

- session.extra_user_variables porte ces données de session liées à l'utilisateur (peut-être le nommer auth_variables ?)
- session.set_user() nettoye cette propriété à chaque login
- c'est écrasé juste après session.set_user() avec la version ici de la dernière authentification
- j'ai viré le session.id = None, puisque ça arrive directement dans set_user() avec #15698

Les nouvelles variables se nommeront session_var_user_fc_{email,given_name,etc.}.

```
diff --git a/wcs/qommon/ident/franceconnect.py b/wcs/qommon/ident/franceconnect.py
index 23e3461..25d3cc2 100644
--- a/wcs/qommon/ident/franceconnect.py
+++ b/wcs/qommon/ident/franceconnect.py
@@ -260,7 +260,8 @@ class FCAuthMethod(AuthMethod):
         state = str(uuid.uuid4())
         session = get_session()
         next_url = get_request().form.get('next') or pub.get_frontoffice_url()
-        session.set_extra_attribute('fc_next_url_' + state, next_url)
+        session.extra_user_variables = session.extra_user_variables or {}
```

```
+        session.extra_user_variables['fc_next_url_' + state] = next_url

         # generate a session id if none exists, ugly but necessary
         get_session_manager().maintain_session(session)
@@ -409,7 +410,8 @@ class FCAuthMethod(AuthMethod):
         session = get_session()
         logger = get_logger()
         state = request.form.get('state', '')
-        next_url = session.pop_extra_attribute('fc_next_url_' + state, '') or pub.get_frontoffice_url()
+        next_url = ((session.extra_user_variables or {}).pop('fc_next_url_' + state, '')
+                    or pub.get_frontoffice_url())

         if 'code' not in request.form:
             error = request.form.get('error')
@@ -417,7 +419,8 @@ class FCAuthMethod(AuthMethod):
             if error:
                 # we log only errors whose user is not responsible
                 msg = self.AUTHORIZATION_REQUEST_ERRORS.get(error)
-                logger.error(_('FranceConnect authentication failed : %s'), _(msg) if msg else error)
+                logger.error(_('FranceConnect authentication failed : %s'),
+                             _(msg) if msg else error)
             return redirect(next_url)
         access_token = self.get_access_token(request.form['code'])
         if not access_token:
@@ -430,8 +433,7 @@ class FCAuthMethod(AuthMethod):
         flatten_dict(flattened_user_info)
         session_var_fc_user = {}
         for key in flattened_user_info:
-            session_var_fc_user['fc_user_' + key] = flattened_user_info[key]
-        session.add_extra_variables(**session_var_fc_user)
+            session_var_fc_user['fc_' + key] = flattened_user_info[key]

         #  Lookup or create user
         sub = user_info['sub']
@@ -451,5 +453,5 @@ class FCAuthMethod(AuthMethod):

         user.store()
         session.set_user(user.id)
-        session.id = None
+        session.extra_user_variables = session_var_fc_user
         return redirect(next_url)
diff --git a/wcs/qommon/sessions.py b/wcs/qommon/sessions.py
index 34d2b6b..033d3eb 100644
--- a/wcs/qommon/sessions.py
+++ b/wcs/qommon/sessions.py
@@ -82,7 +82,8 @@ class Session(QommonSession, CaptchaSession, StorableObject):
     jsonp_display_values = None
     extra_variables = None
     expire = None
-    extra_attributes = None
+    # should only be overwritten by authentication methods
+    extra_user_variables = None

     username = None # only set on password authentication

@@ -117,7 +118,7 @@ class Session(QommonSession, CaptchaSession, StorableObject):
             self.extra_variables or \
             CaptchaSession.has_info(self) or \
             self.expire or \
-            self.extra_attributes or \
+            self.extra_user_variables or \
             QuixoteSession.has_info(self)
     is_dirty = has_info

@@ -151,6 +152,7 @@ class Session(QommonSession, CaptchaSession, StorableObject):
         return None

     def set_user(self, user_id):
+        self.extra_user_variables = None
         QuixoteSession.set_user(self, user_id)
         if str(user_id).startswith('anonymous-'):
             # do not store connection time for anonymous users
@@ -248,16 +250,11 @@ class Session(QommonSession, CaptchaSession, StorableObject):
         if self.extra_variables:
             for k, v in self.extra_variables.items():
```

```
                    d[prefix + k] = v
+           if self.extra_user_variables:
+               for k, v in self.extra_user_variables.items():
+                   d[prefix + 'user_' + k] = v
            return d

-       def set_extra_attribute(self, key, value):
-           if not self.extra_attributes:
-               self.extra_attributes = {}
-           self.extra_attributes[key] = value
-
-       def pop_extra_attribute(self, key, default=None):
-           return (self.extra_attributes or {}).pop(key, default)
-

 class QommonSessionManager(QuixoteSessionManager):
     def start_request(self):
```

Juste pour en parler je pensais que le module SAML pourrait lui aussi y mettre des choses (des données SAML, NameID, id de session SAML, tous les attributs, on pourrait ainsi passer des attributs sans se soucier d'en faire des attributs du profil utilisateur)

#41 - 24 avril 2017 10:26 - Frédéric Péters

Benjamin, tu as quelque part une branche publiée avec les différents diff présents dans des commentaires consolidés dans des commits ?

#42 - 24 avril 2017 11:59 - Benjamin Dauvergne

Rebasé et bug corrigés sur les tests (on crée moins de sessions pour rien maintenant)
http://git.entrouvert.org/wcs.git/log/?h=wip/14510-france-connect

#43 - 28 avril 2017 16:58 - Thomas Noël

Branche rebasée, fonctionnement testé, ça semble OK.

Je viens d'y envoyer deux patches :

- tests de ?next=... pour redirection après login :
  http://git.entrouvert.org/wcs.git/commit/?h=wip/14510-france-connect&id=3d8db49bc11ed332774d0643bfcf5e37fe480834
- correction du lien quand on valide les settings :
  http://git.entrouvert.org/wcs.git/commit/?h=wip/14510-france-connect&id=6bd00b9b2df72225fbed678a9d2e7e193674721b

#44 - 28 avril 2017 17:09 - Frédéric Péters

Pas tout à fait rebasée sur origin/master, les deux commits suivant n'ont pas le bon hash.

```
* 9122b467 workflows: don't include uncompleted choice actions (#15887)
* 7ba9fcbc misc: fix "ask for confirmation" setting to work with new buttons (#15920)
```

> Je viens d'y envoyer deux patches :

Ok, seront à intégrer dans le commit général avant de pousser.

J'aurais préféré que la partie "Also add an extra_user_variables" soit dans son propre commit.

```
+           try:
+               value = WorkflowStatusItem.compute(value, context=user_info)
+           except:
+               continue
```

notify_of_exception ?

> FranceConnect authentication failed : %s

Pas d'espace devant le :.

#45 - 29 avril 2017 00:22 - Thomas Noël

Frédéric Péters a écrit :

> Pas tout à fait rebasée sur origin/master, les deux commits suivant n'ont pas le bon hash.

Rebasé avec un cerveau branché, et git-push-force sur la branche.

> Je viens d'y envoyer deux patches :

> Ok, seront à intégrer dans le commit général avant de pousser.

C'est intégré.

> J'aurais préféré que la partie "Also add an extra_user_variables" soit dans son propre commit.

J'ai commencé et puis en fait session.extra_user_variables est directement utilisé pour la gestion du next_url, alors j'ai laissé le mix en un seul coup.

> notify_of_exception ?

Ajouté.

> FranceConnect authentication failed : %s

> Pas d'espace devant le :.

Je plaide non coupable ; corrigé.

#46 - 02 mai 2017 10:36 - Frédéric Péters

> J'aurais préféré que la partie "Also add an extra_user_variables" soit dans son propre commit.

> J'ai commencé et puis en fait session.extra_user_variables est directement utilisé pour la gestion du next_url, alors j'ai laissé le mix en un seul coup.

Je ne comprends pas, pour moi la modif à sessions.py tient d'elle-même, peut venir avant le commit FranceConnect.

```
+            fc_callback = pub.get_frontoffice_url() + '/ident/fc/callback'
+            r += htmltext(_('<p>Callback URL is <a href="%s">%s</a>.</p>')) % (
+                fc_callback, fc_callback)
```

Je n'en ferais pas un lien, vu que cliquer dessus sera juste une redirection vers la page d'accueil.

```
+            r += htmltext('<div><p>See <a '
+                          'href="https://franceconnect.gouv.fr/fournisseur-service#identite-pivot" '
+                          'target="_blank">FranceConnect partners\'site</a> for more '
+                          'informations on available scopes and attributes. Known ones '
+                          'are :<p>')
```

Ici aussi, pas d'espace avant le :, et il faut que ça soit marqué pour traduction.

```
+            r += (htmltext('<table><thead><tr><th>%s</th><th>%s</th></tr></thead><tbody>') %
+                  (_('Attribute'), _('Description')))
```

Poser une classe sur la balise <table>, le style pourrait être celui actuellement défini pour table#substvars.

```
+            for attribute, description in self.KNOWN_ATTRIBUTES:
+                r += htmltext('<tr><td><pre>%s</pre></td><td>%s</td></tr>') % (attribute, _(description))
```

tt plutôt que pre.

```
+    def is_ok(self):
+        fc_cfg = get_cfg('fc', {})
+        for key, title in self.method_admin_directory.CONFIG:
+            if not fc_cfg.get(key):
+                return False
+        return True
```

Vu la tête de celle-ci, les différents champs du formulaire de configuration ne devraient-ils pas être marqués comme obligatoires ?

```
+            return template.error_page(_('FranceConnect support is not yet configured'))
```

Point final.

```
+import sys
+import json
+import urllib
+import uuid
+import hashlib
+import base64
```

Trier.

```
+         return json.loads(data)
```

Partout ailleurs on utilise json_loads, qui assure l'encodage des chaines de caractères.

#47 - 02 mai 2017 16:04 - Thomas Noël

J'ai push-forcé la branche, avec toutes les remarques. J'ai aussi séparé la modification à compute, et ajouté des bêtes tests (sur compute et sur session.extra_user_variables)

Et par rapport à ce que Frédéric Péters a écrit :

> Je ne comprends pas, pour moi la modif à sessions.py tient d'elle-même, peut venir avant le commit FranceConnect.

Bien sûr (je pensais ajouter la modif à session.py après FC, ce qui était idiot).

> Poser une classe sur la balise <table>, le style pourrait être celui actuellement défini pour table#substvars.

J'ai déclaré « table.franceconnect-attrs » dans admin.css

> tt plutôt que pre.

Et comme tt n'existe plus, c'est code

> Vu la tête de celle-ci, les différents champs du formulaire de configuration ne devraient-ils pas être marqués comme obligatoires ?

Exact, required=True ajouté. Au passage j'ai aussi ajouté un bouton cancel.

#48 - 02 mai 2017 16:30 - Frédéric Péters

À regarder maintenant des détails levés par pylint, RadiobuttonsWidget n'est pas utilisé, son import peut être retiré, et le "from .base import AuthMethod" pourrait être descendu sous les "from wcs...".

```
                for field in user_formdef.fields:
                    if field_varname == field.varname:
                        field_id = str(field.id)
                        break
                else:
                    continue
                form_data[field.id] = value
            if field_varname == '__email':
                field_varname = 'email'  # special value for verified email field
```

```
            # Update verified fields
            if field_id:
                if verified == 'always' and field_id not in user.verified_fields:
                    user.verified_fields.append(field.id)
                elif verified != 'always' and field_id in user.verified_fields:
                    user.verified_fields.remove(field.id)
```

Moins un détail ici, ça va d'une part péter si user_formdef.fields est vide (field ne sera pas défini). Et d'autre part, si field_varname est __email il y aura quelque chose dans field_id mais field ne sera pas défini.

#49 - 02 mai 2017 17:35 - Thomas Noël

Frédéric Péters a écrit :

> Moins un détail ici, ça va d'une part péter si user_formdef.fields est vide (field ne sera pas défini).

Je pense que le "else: continue" protège de ça. Non ?

> Et d'autre part, si field_varname est __email il y aura quelque chose dans field_id mais field ne sera pas défini.

Ah, oui, j'ai l'impression qu'il faudrait faire :

```
             # Update verified fields
             if field_id:
                 if verified == 'always' and field_id not in user.verified_fields:
-                    user.verified_fields.append(field.id)
+                    user.verified_fields.append(field_id)
                 elif verified != 'always' and field_id in user.verified_fields:
-                    user.verified_fields.remove(field.id)
+                    user.verified_fields.remove(field_id)
```

Et par ailleurs, juste au dessus :

```
391             if field_varname == '__email':
392                 field_varname = 'email'  # special value for verified email field
```

ne sert à rien du tout... ?

#50 - 02 mai 2017 17:49 - Frédéric Péters

> > Moins un détail ici, ça va d'une part péter si user_formdef.fields est vide (field ne sera pas défini).
>
> Je pense que le "else: continue" protège de ça. Non ?

Ouaip.

```
391             if field_varname == '__email':
392                 field_varname = 'email'  # special value for verified email field
```

En effet, sert à rien.

```
return template.error_page(_('FranceConnect support is not yet configured'))
```

Autre occurence où manque le point final.

#51 - 02 mai 2017 18:19 - Thomas Noël

Nouveau git push avec le . final.

#52 - 02 mai 2017 22:35 - Frédéric Péters

J'ai pushé dans la branche une série de commits sur d'autres commentaires que je me suis fait,

- dcf7fd01 move fc to last place : plutôt qu'avoir FranceConnect placé en première méthode d'authentifcation.
- 4e8a48b9 invert submitted/not submitted code parts to match what's done recently : depuis un certain temps je préfère cette forme, qui permet notamment d'avoir un niveau d'indentation en moins pour le gros de la méthode.
- b1132d31 remove target=_blank, parce que ça ne devrait jamais exister.
- d2809436 avoid markup in translatable strings, properly open <p>, pour éviter des bugs amenés par la traduction et parce qu'un <p> manquait.
- 6f301946 don't duplicate css, pour vraiment partager le code existant.

Mon idée est de laisser juger de leur pertinence, puis qu'ils soient fusionnés dans le commit FranceConnect.

+ le patch de #16140.

#53 - 03 mai 2017 10:38 - Thomas Noël

Frédéric Péters a écrit :

> J'ai pushé dans la branche une série de commits sur d'autres commentaires que je me suis fait,
>
> - dcf7fd01 move fc to last place : plutôt qu'avoir FranceConnect placé en première méthode d'authentifcation.
> + le patch de #16140.

Ack

- 4e8a48b9 invert submitted/not submitted code parts to match what's done recently : depuis un certain temps je préfère cette forme, qui permet notamment d'avoir un niveau d'indentation en moins pour le gros de la méthode.

Ca marche, ack.

- b1132d31 remove target=_blank, parce que ça ne devrait jamais exister.

Je me disais aussi. Ack.

- d2809436 avoid markup in translatable strings, properly open <p>, pour éviter des bugs amenés par la traduction et parce qu'un <p> manquait.

Ca roule, ack.

- 6f301946 don't duplicate css, pour vraiment partager le code existant.

Ack, of course (quel nul je fais...)

> Mon idée est de laisser juger de leur pertinence, puis qu'ils soient fusionnés dans le commit FranceConnect.

Je m'occupe de fusionner tout ça, merci.

#54 - 03 mai 2017 11:17 - Thomas Noël

- Lié à Development #16144: permettre l'utilisation de compute() en dehors des variables de substitution ajouté

#55 - 03 mai 2017 11:34 - Thomas Noël

- Lié à Development #16145: ajouter un attribut extra_user_variables aux sessions ajouté

#56 - 03 mai 2017 11:35 - Thomas Noël

- Fichier 0001-add-FranceConnect-authentication-method-14510.patch ajouté

Voici le patch sur master, isolé, pour dernière (?) relecture.

#57 - 03 mai 2017 15:58 - Frédéric Péters

Hop, ack.

#58 - 03 mai 2017 16:08 - Thomas Noël

- Statut changé de En cours à Résolu (à déployer)

```
commit c1a923c2d07badeeaa3cae33ff19ea0a6465602c
Author: Thomas NOEL <tnoel@entrouvert.com>
Date:   Tue May 2 15:56:50 2017 +0200

    add FranceConnect authentication method (#14510)

    use session.extra_user_variables to store attributes retrieved
    during FranceConnect SSO.
```

#59 - 23 décembre 2018 14:51 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Solution déployée

Fichiers

0001-factorize-WorklowStatusItem.compute.patch	2,93 ko	05 janvier 2017	Benjamin Dauvergne
0002-move-wcs.formdata.flatten_dict-into-qommon.misc.patch	1,7 ko	05 janvier 2017	Benjamin Dauvergne
0003-add-france-connect-authentication-method.patch	20,9 ko	05 janvier 2017	Benjamin Dauvergne
0004-add-an-inspect-view-for-global-substitution-variable.patch	2,45 ko	05 janvier 2017	Benjamin Dauvergne
inspect_screenshot.png	400 ko	05 janvier 2017	Benjamin Dauvergne
france_connect_settings_screenshot.png	311 ko	05 janvier 2017	Benjamin Dauvergne
0003-add-france-connect-authentication-method.patch	21,3 ko	05 janvier 2017	Benjamin Dauvergne
0001-factorize-WorklowStatusItem.compute.patch	13,4 ko	06 janvier 2017	Benjamin Dauvergne
0003-add-FranceConnect-authentication-method-fixes-14510.patch	20,8 ko	06 janvier 2017	Benjamin Dauvergne

0003-add-FranceConnect-authentication-method-fixes-14510.patch	25,5 ko	06 janvier 2017	Benjamin Dauvergne
0001-use-misc.compute-instead-of-kflowStatusItem.compute.patch	2,61 ko	07 janvier 2017	Benjamin Dauvergne
0001-factorize-WorklowStatusItem.compute.patch	13,4 ko	07 janvier 2017	Benjamin Dauvergne
0003-add-FranceConnect-authentication-method-fixes-14510.patch	9,78 ko	07 janvier 2017	Benjamin Dauvergne
0002-move-wcs.formdata.flatten_dict-into-qommon.misc.patch	1,72 ko	07 janvier 2017	Benjamin Dauvergne
0001-factorize-WorklowStatusItem.compute.patch	13,4 ko	07 janvier 2017	Benjamin Dauvergne
0003-add-FranceConnect-authentication-method-fixes-14510.patch	27,9 ko	07 janvier 2017	Benjamin Dauvergne
0004-add-an-inspect-view-for-global-substitution-variable.patch	2,45 ko	07 janvier 2017	Benjamin Dauvergne
0001-factorize-WorklowStatusItem.compute.patch	13,4 ko	17 février 2017	Benjamin Dauvergne
0002-move-wcs.formdata.flatten_dict-into-qommon.misc.patch	1,72 ko	17 février 2017	Benjamin Dauvergne
0003-add-FranceConnect-authentication-method-fixes-14510.patch	27,9 ko	17 février 2017	Benjamin Dauvergne
0004-add-an-inspect-view-for-global-substitution-variable.patch	2,45 ko	17 février 2017	Benjamin Dauvergne
0001-factorize-WorklowStatusItem.compute-.-14510.patch	4,83 ko	16 mars 2017	Benjamin Dauvergne
0002-replace-use-of-self.compute-.-by-misc.compute-.-1451.patch	8,97 ko	16 mars 2017	Benjamin Dauvergne
0003-move-wcs.formdata.flatten_dict-into-qommon.misc-1451.patch	1,71 ko	16 mars 2017	Benjamin Dauvergne
0004-add-FranceConnect-authentication-method-14510.patch	27,8 ko	16 mars 2017	Benjamin Dauvergne
0001-add-FranceConnect-authentication-method-14510.patch	28,7 ko	28 mars 2017	Thomas Noël
0001-add-FranceConnect-authentication-method-14510.patch	28,8 ko	28 mars 2017	Thomas Noël
0001-add-FranceConnect-authentication-method-14510.patch	30,3 ko	29 mars 2017	Thomas Noël
0001-add-FranceConnect-authentication-method-14510.patch	30,8 ko	29 mars 2017	Thomas Noël
0001-tomerge.patch	7,51 ko	29 mars 2017	Benjamin Dauvergne
0001-add-FranceConnect-authentication-method-14510.patch	32,2 ko	03 mai 2017	Thomas Noël