Authentic 2 - Development #20699

Fournir dans le contexte des template le service appelant sur toutes la pages

14 décembre 2017 12:52 - Mikaël Ates (de retour le 29 avril)

Statut:	Fermé	**Début:**	14 décembre 2017
Priorité:	Normal	**Echéance:**	
Assigné à:	Valentin Deniaud	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Travail similaire a celui fait dans le CUT lors du login OIDC ou sur la page Mon compte avec le champs next. Inclure SAML et CAS.

Cela permettra notamment la personnalisation de l'interface selon le service ou le partenaire.

Demandes liées:

Lié à Authentic 2 - Development #20700: Pouvoir faire varier les éléments du ...	**Nouveau**	**14 décembre 2017**
Lié à Authentic 2 - Development #23187: Mise à disposition du service d'origi...	**Rejeté**	**15 avril 2018**
Lié à Hobo - Development #57482: Exploiter le paramètre ?service= pour les UR...	**Fermé**	**01 octobre 2021**

Révisions associées

Révision 41aa3734 - 17 septembre 2020 11:39 - Valentin Deniaud

misc: provide origin service in template context (#20699)

Historique

#1 - 14 décembre 2017 12:55 - Mikaël Ates (de retour le 29 avril)

- Lié à Development #20700: Pouvoir faire varier les éléments du thème selon le service appelant ajouté

#2 - 18 avril 2018 08:01 - Mikaël Ates (de retour le 29 avril)

- Lié à Development #23187: Mise à disposition du service d'origine dans le contexte ajouté

#3 - 08 juillet 2019 11:01 - Benjamin Dauvergne

Pour les pages enregistrement et login on reçoit un paramètre &service= avec le slug du service (modèle authentic2.models.Service) ça peut déjà aider à poser un truc en session repris ensuite par un template context processor introduisant une variable "a2_current_service" (et en mettant un classe"service-{{ service.slug }}" sur body et aussi un data-service-slug="{{ service.slug }}" data-service-name="{{ service.name }}" dans les templates de base par exemple).

Viennent ensuite les next, je dirai de mettre ça dans good_next_url(request, next_url) qui est déjà le point d'entré dans normalement (!) toutes les vues pour la gestion des URLs de next. Pour chaque type de service (SAML,OIDC,CAS) comme on déduit les origines acceptables il faut faire le processus inverse, d'une origine retrouver un service et le mettre en session.

#5 - 24 août 2020 17:23 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

J'ai commencé à regarder, ça n'a pas l'air si terrible comme ticket. Le plus dur pour l'instant c'était de comprendre que dans le plan de Benj, le premier paragraphe parle des briques Publik, et le second des SP tierces (et que next ne veut pas dire ?next mais n'importe quoi qui peut servir à remonter au service, typiquement ?redirect_uri). Si j'ai bon sur tout ça ça devrait rouler !

#6 - 24 août 2020 17:31 - Benjamin Dauvergne

Tu peux t'inspirer du code existant dans authentic2-cut http://git.entrouvert.org/authentic2-cut.git/tree/src/authentic2_cut/middlewares.py (qui ne gère pas le coté SAML et CAS); au passage ne pas casser ce code là non plus, de ce coté il suffit de ne pas mettre de clé service_slug ou cut_domain dans la session et ça devrait suffire; à mon avis le plus simple ce serait un propriété service sur request qui soit extrait depuis la requête soit depuis la session (si on arrive sur des pages où on a plus le next ou redirect_uri).

#7 - 25 août 2020 15:12 - Valentin Deniaud

Effectivement le code de CUT donne une bonne idée de ce qu'il faut faire, je vais donc écrire un nouveau middleware qui ajoute un attribut à request (et puis en bonus les modifs au template de base dont tu parles plus haut).

Par contre je ne comprends pas comment ce ticket répond au cas d'usage où c'est la page gestion du compte qu'on veut faire varier, dans le cas de Toulouse par ex. Ça ne me paraît pas du tout satisfaisant d'avoir « depuis où s'est connecté l'usager » et d'en déduire d'où il vient maintenant.
Exemple simple parmi beaucoup d'autre, deux onglets, un sur le site de la ville un sur la métro. Des deux onglets il peut aller sur « Mon compte », et en fonction du site sur lequel il s'est connecté en dernier on va afficher la page « Mon compte » de la ville ou de la métro, dans les deux onglets, ce qui n'est pas le comportement attendu.

Et aussi,

> Viennent ensuite les next, je dirai de mettre ça dans good_next_url

C'est quoi « ça » ?

#8 - 25 août 2020 15:31 - Frédéric Péters

C'est l'idée que Toulouse mettrait dans les liens "Mon compte" un ?next=https://la-metropole/ et qu'ainsi, hors SSO, on aurait une indication "à jour" du retour souhaité. (ceci est ma lecture de l'affaire, à confirmer par Benjamin)

#9 - 25 août 2020 16:54 - Benjamin Dauvergne

Frédéric Péters a écrit :

> C'est l'idée que Toulouse mettrait dans les liens "Mon compte" un ?next=https://la-metropole/ et qu'ainsi, hors SSO, on aurait une indication "à jour" du retour souhaité. (ceci est ma lecture de l'affaire, à confirmer par Benjamin)

Cette partie là sort du cadre du titre du ticket, sauf à permettre de définir pour un service une URL unique de retour, si ce n'est pas suffisant il faut prévoir de stocker en session le service mais aussi la dernière URL de retour reçu (en espérant qu'elle gère bien un retour simple, pour OIDC ça n'est pas certain, sans paramètre ?code= ou ?error= certaines URL de callback ne fonctionnent juste pas, coté SAML c'est pire).

C'est là ou je n'ai rien à proposé de clair, coté GLC on gère une base maison qui fait le lien entre les URLs et du paramétrage de thème et une URL de retour par défaut. Ici disons qu'on va se retrouver dans le contexte du template avec une variable service, ça nous donne son slug service.slug et le slug de son ou service.ou.slug. Est-ce qu'on en fait automatiquement des classes à poser sur <body> ? Est-ce qu'on prévoit comme sur GLC (mais plutôt dans le backoffice que dans les settings) GLC un moyen d'attacher du paramétrage aux OU ou service genre URL du logo, couleur, etc.. ?

En fait le focus du ticket est un peu étroit il en faudrait un chapeau qui aille au delà des histoires de thème, c'est plus une histoire de contexte de navigation, ça inclut faire varier le thème à la marge (logou, couleur, titre) et pouvoir proposer un lien retour pertinent, soit par défaut soit adapté au contexte d'origine. Pour l'URL de retour je suis toujours frileux à l'idée de la stocker en session, c'est le plus simple ça marche, mais si on ouvre la page mon compte depuis deux services différents ça fait des trucs bizarres.

#10 - 25 août 2020 17:01 - Benjamin Dauvergne

Valentin Deniaud a écrit :

> Par contre je ne comprends pas comment ce ticket répond au cas d'usage où c'est la page gestion du compte qu'on veut faire varier, dans le cas de Toulouse par ex. Ça ne me paraît pas du tout satisfaisant d'avoir « depuis où s'est connecté l'usager » et d'en déduire d'où il vient maintenant.
> Exemple simple parmi beaucoup d'autre, deux onglets, un sur le site de la ville un sur la métro. Des deux onglets il peut aller sur « Mon compte », et en fonction du site sur lequel il s'est connecté en dernier on va afficher la page « Mon compte » de la ville ou de la métro, dans les deux onglets, ce qui n'est pas le comportement attendu.

Comme le dit Fred ça demande à ce que chaque appelle à la page "Mon compte" se fasse avec un paramètre ?next=... et qu'on arrive à transformer le domaine de cette URL en service, d'où ma référence à good_next_url() qui fait ça sans renvoyer le service, juste en disant Oui/Non « c'est une URL validée ».

Pour rebondir sur mon commentaire d'avant, en imaginant qu'on ait un paramétrage "URL de retour par défaut" au niveau service/OU, on pourrait aussi gérer un paramètre ?service=<ou-slug> <service-slug> comme la page de login (ça arrive dans #45672).

> > Viennent ensuite les next, je dirai de mettre ça dans good_next_url
>
> C'est quoi « ça » ?

Mal exprimé, derrière good_next_url() il y a un hook implémenté par les IdPs SAML et OIDC pour vérifier si par hasard l'URL ne correspondrait pas à un service, il faudrait récupérer ça pour lui faire répondre à la question générique "à quel service correspond cette URL ?" plutôt que "est-ce que cette URL correspond à au moins un service ?".

Si on reste concentré sur ce ticket, c'est tout ce qui est demandé ici : depuis /accounts/?next=... en déduire un service et le mettre en session et dans le contexte du template (à mon avis sans le reste ça ne répond pas au problème d'URL de retour, pour ça il faut du paramétrage en plus, c'est la discussion que je propose ici ou ailleurs).

#13 - 25 août 2020 18:06 - Valentin Deniaud

OK donc on se retrouverait à mettre non pas ?next=https://la-metropole/ mais ?next=https://la-metropole/accounts/mellon/login/ pour que ça marche, ça fera un peu bizarre, ou alors se contenter de regarder le domaine.

Je note aussi qu'il va sûrement falloir distinguer le cas « avoir le service au SSO » et « avoir le service quand l'utilisateur est déjà connecté », la moulinette qui trouve un service à partir d'une URL présente dans request.GET n'est pas adaptable au SSO SAML où on est bien obligé d'aller voir l'assertion.

#14 - 02 septembre 2020 17:34 - Valentin Deniaud

Je m'auto réponds :

Valentin Deniaud a écrit :

> OK donc on se retrouverait à mettre non pas ?next=https://la-metropole/ mais ?next=https://la-metropole/accounts/mellon/login/ pour que ça marche, ça fera un peu bizarre, ou alors se contenter de regarder le domaine.

N'importe quoi, il suffit de regarder le domaine.

> Je note aussi qu'il va sûrement falloir distinguer le cas « avoir le service au SSO » et « avoir le service quand l'utilisateur est déjà connecté », la moulinette qui trouve un service à partir d'une URL présente dans request.GET n'est pas adaptable au SSO SAML où on est bien obligé d'aller voir l'assertion.

Non, sur la page de login comme dit plus haut on a ?service qui est mis par les backends saml et oidc, on se sert de ça.

#15 - 02 septembre 2020 18:04 - Valentin Deniaud

Benjamin Dauvergne a écrit :

> Pour rebondir sur mon commentaire d'avant, en imaginant qu'on ait un paramétrage "URL de retour par défaut" au niveau service/OU, on pourrait aussi gérer un paramètre ?service=<ou-slug> <service-slug> comme la page de login (ça arrive dans #45672).

À coder le truc je pense qu'il faudrait tout à fait faire ça maintenant, en laissant de côté cette histoire d'URL de retour dont on ne se sert pas : du côté de la personnalisation de l'inté graphique, niet, et du côté du bouton de retour on fera {% if "tm" in service.slug %}url métropole{% else %}url ville{% endif %}.

Ça permet d'introduire tranquillement le concept d'url de retour par défaut par la suite, quand un vrai cas d'usage se présentera.

#16 - 03 septembre 2020 11:23 - Benjamin Dauvergne

Valentin Deniaud a écrit :

> Ça permet d'introduire tranquillement le concept d'url de retour par défaut par la suite, quand un vrai cas d'usage se présentera.

Toutafé.

#17 - 03 septembre 2020 17:45 - Valentin Deniaud

- Fichier 0001-misc-provide-origin-service-in-template-context-2069.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Et donc tout ça pour ce micro patch.

Si on veut vraiment les data-service-slug="{{ service.slug }}", il faudra penser à les ajouter dans publik-base-theme.

Benjamin Dauvergne a écrit :

> Pour les pages enregistrement et login on reçoit un paramètre &service=

Je n'ai pas réussi à reproduire le cas où on avait ce paramètre ajouté sur la page d'enregistrement.
Dans hobo on voit un variables['idp_registration_url'] += '?%s' % urlencode({'next': variables['portal_user_url']}) où l'on pourrait sûrement ajouter un paramètre service si c'est de ça dont il est question.

#18 - 17 septembre 2020 11:29 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

#19 - 17 septembre 2020 11:41 - Valentin Deniaud

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

```
commit 41aa3734e3e98a9ffe43ea99c52aaa71b8c43f36
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date:   Tue Aug 25 10:07:49 2020 +0200

    misc: provide origin service in template context (#20699)
```

#20 - 18 septembre 2020 08:16 - Frédéric Péters

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

#21 - 01 octobre 2021 09:55 - Frédéric Péters

C'est relancé #35468#note-16 pour que ça soit exploité et je ne capte en fait pas trop, il manque encore des morceaux pour que ça marche pour de vrai (i.e. que je puisse faire {% if service.whatever %} dans un gabarit authentic2/accounts.html custom) ?

Genre ajouter des bouts ?service=??? dans hobo/multitenant/settings_loaders.py, à :

```
                variables['idp_account_url'] = service.get('base_url') + 'accounts/'
                variables['idp_registration_url'] = service.get('base_url') + 'accounts/register/'
```

#22 - 01 octobre 2021 10:00 - Benjamin Dauvergne

- Lié à Development #57482: Exploiter le paramètre ?service= pour les URLs "Mon compte" et d'enregistrement ajouté

Fichiers

0001-misc-provide-origin-service-in-template-context-2069.patch	5,21 ko 03 septembre 2020	Valentin Deniaud