Publik - Documentation #24603

Améliorer le laïus raccordement d'un portail métier

18 juin 2018 11:20 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	18 juin 2018
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Club:**	Non
Planning:	Non		

Description

Voir https://dev.entrouvert.org/projects/publik/wiki/Mise_en_place_d'un_SSO_avec_un_portail_m%C3%A9tier

La nouvelle page devra:

- être directive, éviter les options,
- se restreindre au protocole OIDC
- parler de SLO et de défédération (suppression de liaison)
- évoquer les parcours
- évoquer les web-services de remonter d'information (basés sur le sub)

Historique

#1 - 18 juin 2018 16:49 - Benjamin Dauvergne

- séparer le cas des portails avec comptes métiers nécessaires (un compte purement déclaratif est impossible) et les portails où un compte déclaratif est possible
- possibilité de raccordement automatique via email: déterminer si c'est juridiquement valable ou pas
- raccordement: 1-1, n-1 ou n-n (un compte publik = un compte portail métier, des comptes publik vers un compte portail métier, ou des comptes publiks vers des comptes portails métier), dépend du domaine (portail famille, favoriser le 1-1 si comptes portail métier personne, sinon n-1 si compte famille/collectivité/groupe/entreprise/association).

Parcours:

- SSO: bouton de connexion sur le portail, initiation du sso via URL /oidc/login/, renvoie vers l'IdP, authentification, authorisation, retour sur le portail,
  - si compte connu fini
  - si compte inconnu:
    - le portail nécessite une liaison avec un compte métier: processus de liaison spécifique au portail:
      - identifiants métiers (ex.: code famille, code personne, secret)
      - raccordement via un identifiant commun validé (numéro de mobile validé, email validé)
      - le lien entre le "sub" Publik et l'identifiant "métier" doit être conservé dans une table à deux colonnes "| sub | id_metier |", en fonction de la cardinalité des liaisons (voir plus haut), prévoir index d'unicité sur (sub, id_metier), (sub) ou (id_metier).
    - le portail ne nécessite pas de liaison avec un compte métier : création sur la base des informations (attributs) fournies par Publik (nom, prénom, email, etc...) et d'éventuelles données supplémentaires demandées via un formulaire sur le portail avant création définitive, la contrainte d'unicité est à voir en fonction de la cardinalité (voir remarque plus haut); on pourra éventuellement y conserver des données ne faisant pas parti du modèle de donnée de l'application métier mais utile dans le cadre de la liaison (adresse email, numéro mobile, etc..)
- SLO:
  - déconnexion depuis publik, le portail est notifié de la déconnexion via le processus de SLO prévu par OIDC, il doit supprimer la session ouverte (si il y en a une en cours)
  - déconnexion depuis le portail: après avoir supprimer sa propre session il doit rediriger l'utilisateur sur l'URL de déconnexion OIDC exposée par Publik
- déliaison:
  - le portail doit permettre dans une page "Mes informations" ou "Mon profile" ou tout autre label adéquat de couper la liaison avec Publik, ou si le compte n'a plus d'intérêt suite à la déliaison doit permettre la suppression pure et simple du compte
    - si compte en liaison avec un compte métier: supprimer la ligne dans la table de liaison
    - si compte autonome: supprimer purement et simplement le compte
  - coté publik il est possible de supprimer l'autorisation, cela supprimera tout accès via web-service aux informations de Publik, mais ne notifiera pas le service; celui-ci pourra par ailleurs venir valider régulièrement que le compte existe encore pour traiter en tâche de fond ces suppressions (voir API de synchronisation sur A2).
- remontée d'information:
  - prévoir des web-services acceptant le sub dans leur URLs comme paramètre (ex. "/api/demandes/?sub=<sub>") les données sont à renvoyer au format JSON de cette forme

```
{
  "err": 0,
  "data": {
    "demandes": [
    ]
  }
}
```

- traiter le cas des factures (/api/factures/?sub=<sub> retourne description standardisée de la facture + lien de détail + paiement)
- traiter le cas des informations de profil (famille, association, entreprise, etc..)

Voir ici si des évolutions sont nécessaires de notre coté (idée: renvoyer dans le profil full de l'utilisateur par authentic les identifiants OIDC vers les services).

#2 - 18 juin 2018 18:42 - Stéphane Laget

- Description mis à jour

#3 - 19 juin 2018 10:47 - Benjamin Dauvergne

- Description mis à jour

#4 - 19 juin 2018 11:28 - Benjamin Dauvergne

Précision pour le SSO (from Thomas & Pierre);

- L'URL https://portail-metier/oidc/login/ doit permettre d'initier le SSO vers Publik de n'importe ou (depuis la homepage du portail, depuis le portail citoyen, depuis A2,etc..), il prendra un paramètre supplémentaire ?next=... pour gérer les liens profonds (ex.: ?next=/mes-factures/).

- On va exclure ici le cas d'un pure service en ligne (et donc toujours besoin d'une liaison sub<->id_metier), on fera éventuellement une autre page pour les pures services en ligne (petites-annonces à Vincennes, democracyOS, l'appli PHP du stagiaire, etc..).

#5 - 19 juin 2018 11:39 - Benjamin Dauvergne

Précision pour les remontées de factures:

- une facture c'est le descriptif suivant obligatoire:
  - une date (émission de la facture)
  - un intitulé
  - un statut (payé, à payer, plus payable)
  - un lien vers le détail/la page de paiement
    - j'exclue volontairement la date de limite de paiement c'est leur problème à afficher dans la page du détail, nous on veut juste savoir si c'est payable ou pas
- une demande c'est:
  - une date de création
  - un intitulé
  - un résumé (optionnel)
  - un statut : NOUVEAU, EN COURS DE TRAITEMENT, EN ATTENTE D'INFORMATION, TRAITÉ
  - dernier commentaire (optionnel)
  - un lien vers le détail/suivi
- un profil c'est une liste blocs contenant des blocs ou des champs et des liens "éditer" pour permettre un peu de généricité:
  - le lien edit_url est toujours optionnel
  - un bloc:

```
{ "type": "block"
  "label": "Mes informations familles",
  "content": [],
}
```

    - un bloc contient soit des champs soit d'autres blocs
  - un champ:

```
{ "type": "field",
  "id": "adresse",
  "label": "Adresse",
  "value": "1 rue du calvaire\n13400 AUBAGNE",
  "format": "pre"
}
```

    - le format pre indique une chaîne pré-formatté (conserver les sauts de ligne)
  - ce système doit permettre d'afficher simplement un profil famille (papa, maman, adresse, les enfants), d'une entreprise ou d'une

> association en laissant à l'application métier le choix de la structure et en proposant des liens d'édition soit au niveau des blocs ou des champs individuels, à nous de soigner le rendu.

#6 - 19 juin 2018 11:53 - Benjamin Dauvergne

Pour toutes les remontées être assertif sur le fait que les données doivent être utilisables telles quelles sans manipulations (donc avoir des valeurs textuelles compréhensibles du premier coup), on ne veut pas justement pas développer de connecteur pour faire du mapping.

#7 - 19 juin 2018 12:21 - Benjamin Dauvergne

Fred pointe notre existant en matière d'API famille et factures ou nos souhaits précédents en matière de gestion des liaisons/fédérations:

- https://dev.entrouvert.org/projects/passerelle/wiki/Connecteur_famille : trop restrictif pour moi, suppose implicitement qu'on va vouloir faire du formulaire autour (d'où présence de donnée métier comme les identifiant métier des différents objets, inutile pour du simple affichage), ne s'étend pas facilement aux autres cas (entreprise, assoce, groupe ad-hoc, intranet, inscription à des activités, etc..), impose un modèle de donnée, ne donne pas beaucoup de liberté de mise en forme (si les adresses ne sont pas découpés comme on le souhaite dans le logiciel métier on est foutu), je ne viserai pas d'être compatible avec ça, demande de toute façon de la cellule JSON ad-hoc pour la présentation donc pas un gros gain de temps de travail pour nous,
- https://dev.entrouvert.org/projects/passerelle/wiki/API_Factures : là par contre pas du tout hostile à viser un compatibilité, il faut juste ajuster ça pour permettre le cas de factures "externes" à Publik, i.e. qui seront payées dans le portail métier, la présence d'un lien detail_url/payment_url devrait suffire comme ajustement
- https://dev.entrouvert.org/issues/13327 API évoqué pour une gestions des liaisons affiché dans Publik mais gérer via WS dans le logiciel métier : un peu anti-SSO, suppose qu'on envoie notre UUID Publik unique, interagit mal avec le fait de conserver les autorisations dans A2 pour OIDC, ne laisse pas la main au portail métier pour gérer le processus de liaison comme il l'entend

Pour expliciter un peu mon objectif ce sont les intégrations à la libre-air pour donner quelque chose de cohérent et d'intégré, éviter les iframes et la multiplication des cellules demandes/factures/etc..

#8 - 05 juillet 2018 22:19 - Benjamin Dauvergne

Coté logout ne pas oublier de parler du profil "frontchannel logout" qu'on gère désormais (SLO initié de l'IdP), ça veut dire un endpoint de logout en plus.

#9 - 18 juillet 2018 14:59 - Benjamin Dauvergne

Voilà v0 du document : Raccordement d'un portail métier

Il me reste :

- relire/reprendre la partie OIDC avec des diagrammes de séquence et revenir à ce que je disais sur le précédent document en parsant l'idtoken plutôt que d'appeler user_info, ça gagne un round-trip
- intégrer les remarques qui seront faites sur la partie remontée
- décrire la roadmap de notre coté pour que ce qui est vaporware ne le soit plus :
  1. pouvoir récupérer le sub servi à un service depuis combo (en améliorant l'API A2 /api/users/xxx?full pour qu'elle retourne toutes les fédérations)
  2. ajouter à la cellule facture de combo le support de régies "externe" i.e. où le paiement ne se fait pas dans lingo (donc surtout les champs payment_url et pdf_url plutôt que d'appeler un web-service de récupération de la facture) en utilisant le sub récupéré en 1. plutôt que le NameID interne à Publik
  3. ajouter à la cellule des demandes w.c.s. la possibilité de récupérer les demandes d'un service distant autre que w.c.s. en utilisant le sub récupéré en 1. plutôt que le NameID interne à Publik

#10 - 18 juillet 2018 15:02 - Frédéric Péters

J'entends la portée "... portail métier" mais on a aujourd'hui des cellules factures qui fonctionnent sans cet impératif et je ne suis pas fan de l'idée d'avoir à gérer plusieurs API, ou d'avoir à migrer quoique ce soit pour prendre en compte des changements. Je n'ai pas fait de recherche exhaustive, peut-être qu'il y a "juste" à autoriser NameID et sub comme synonymes, et adapter combo pour qu'en cas de payment_url ça envoie là plutôt que déclencher un paiement, etc. mais avoir comme accompagnement à ce document ce que ça implique comme changements de notre côté, ce serait super.

L'exemple pour "Remontées d'autres informations" ne correspond pas à la documentation. (content vs text par exemple).

"il doit accepter un seul paramètre à cette url: sub" me semble excessif, on a une facilité pour faire des URL propres qui permettent ça mais ce n'est pas le cas de toutes les applications.

#11 - 18 juillet 2018 16:01 - Benjamin Dauvergne

Frédéric Péters a écrit :

> J'entends la portée "... portail métier" mais on a aujourd'hui des cellules factures qui fonctionnent sans cet impératif et je ne suis pas fan de l'idée d'avoir à gérer plusieurs API, ou d'avoir à migrer quoique ce soit pour prendre en compte des changements. Je n'ai pas fait de recherche exhaustive, peut-être qu'il y a "juste" à autoriser NameID et sub comme synonymes, et adapter combo pour qu'en cas de payment_url ça envoie là plutôt que déclencher un paiement, etc. mais avoir comme accompagnement à ce document ce que ça implique comme changements de notre côté, ce serait super.

Nos commentaires se sont croisés à 3 minutes près, c'est juste avant le tien.

> L'exemple pour "Remontées d'autres informations" ne correspond pas à la documentation. (content vs text par exemple).

Je vais repasser dessus mais j'attends surtout des "ok ça ira" ou "ça non ça permet trop/pas assez, fais plutôt comme cela" avant de rendre ça cohérent.

> "il doit accepter un seul paramètre à cette url: sub" me semble excessif, on a une facilité pour faire des URL propres qui permettent ça mais ce n'est pas le cas de toutes les applications.

Ok, je vais plutôt mettre "Publik rajoutera un seul paramètre à cette URL (qui pourra en contenir d'autres)".

#12 - 18 juillet 2018 16:19 - Benjamin Dauvergne

Frédéric Péters a écrit :

> J'entends la portée "... portail métier" mais on a aujourd'hui des cellules factures qui fonctionnent sans cet impératif et je ne suis pas fan de l'idée d'avoir à gérer plusieurs API, ou d'avoir à migrer quoique ce soit pour prendre en compte des changements.

Si je prends l'API de cellules w.c.s. sans y toucher ça veut dire:

- recevoir du paramètre NameID
- obliger à supporter notre système de signature de web-service
- retourner {"data":[{"url":..,"title":, "form_number":.., "status":...}..], pour l'instant ne pas savoir afficher la date de la demande (on pourrait le faire pour w.c.s on l'a dans "receipt_time", c'est pas un gros changement, mais si on l'affiche pas actuellement je suppose que c'est parce qu'on n'y trouve pas beaucoup d'intérêt), ni de résumé ou d'autres informations, ou alors il faut prévoir que pour les demandes venant d'un certain slug on puisse fournir un template différent de celui de base.

Je veux bien pousser vers une voie médiane, garder le format w.c.s. (donc virer ce qui dépasse dans ma proposition), par contre permettre déclarer ces sources de demandes de manière différente (à voir comment, ce serait bien qu'on ait pas à se répéter pour déclarer un portail métier), que ça passe sub (c'est vraiment important si on veut pouvoir faire de la fédération et pas continuer à distribuer un identifiant unique), que l'authentification passe par un truc standard ou on a pas besoin de fournir du code et des explications.

Coté facture, on a juste pas ce qu'il faut donc il y a seulement deux alternatives :

- on complète notre API et au passage on la simplifie,
- on oublie l'objectif d'intégration, un portail métier il a des cellules différentes pour ses demandes, ses factures, etc.. (et ça réduit ma doc coté remontée d'information au web-service générique)

Une dernière chose: pour la remontée d'information générique, je suis allé vers un truc qui mixe présentation et données mais je peux encore aller vers un truc purement sémantique; ce que je me suis dit c'est que les changements de wording seront pour l'éditeur du portail métier et plus pour nous, tout en évitant de recevoir du pure HTML dégueulasse qui casse notre intégration. Nous au pire on aura un peu de style, on pourra éventuellement proposé des classes toutes faites (les grid-x-x, tout ça).

#13 - 18 juillet 2018 16:35 - Frédéric Péters

> Si je prends l'API de cellules w.c.s. sans y toucher ça veut dire:

Soyons clair, mes craintes sur les évolutions ne concernent pas w.c.s., on peut décider d'y gérer de l'auth http, de gérer un ?sub=, etc. Par contre les remontées familles/factures actuelles… De là, avoir une idée des changements attendus, que les évolutions demandées à nos logicielles soient posées et mesurées, et qu'on ne se trouve pas à demander à quelqu'un assemblant son portail à choisir entre "factures (format 1)" et "factures (format 2)".

Ensuite, à revenir sur le sub, "pas continuer à distribuer un identifiant unique", et donc "pouvoir récupérer le sub servi à un service depuis combo (en améliorant l'API A2 /api/users/xxx?full pour qu'elle retourne toutes les fédérations)"; ça m'ennuie d'avoir à caler des appels supplémentaires, ça serait concevable de passer l'info dans le provisionning, faire en sorte qu'on en dispose en local ?

#14 - 18 juillet 2018 17:55 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Ensuite, à revenir sur le sub, "pas continuer à distribuer un identifiant unique", et donc "pouvoir récupérer le sub servi à un service depuis combo (en améliorant l'API A2 /api/users/xxx?full pour qu'elle retourne toutes les fédérations)"; ça m'ennuie d'avoir à caler des appels supplémentaires, ça serait concevable de passer l'info dans le provisionning, faire en sorte qu'on en dispose en local ?

J'étais plus dans l'idée de faire en sorte que ça aille vite peut-être en ayant directement un /api/users/<uuid>/links/<service-slug>/ et en faisant du cache coté combo.

#15 - 26 janvier 2022 09:35 - Benjamin Dauvergne

- Statut changé de Nouveau à Fermé

- Assigné à Benjamin Dauvergne supprimé

- Planning mis à Non

- Club mis à Non

C'est Thomas qui a repris tout ça.