

Publik - Development #49204

"Application" des modules (applications métiers) Publik

08 décembre 2020 06:03 - Pierre Cros

Statut:	En cours	Début:	08 décembre 2020
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:	Studio	Temps estimé:	0:00 heure
Version cible:	2022	Club:	Non
Patch proposed:	Non		
Planning:	Non		

Description

(description mise à jour pour exposer de manière plus détaillée les interfaces et le plan technique)

Dans Hobo un nouvel écran "Applications", avec des boutons Installer et Créer. (plus tard, un menu kebab, avec une entrée "Catalogues").

Écran de création

Interface avec une petite première section de métadonnées, un nom, une icône, une courte description.

Dans cette interface la liste des types d'objet qui peuvent être intégrés (ça peut être une liste + bouton ajouter comme pour les cellules dans combo), après la sélection d'un type d'objet possibilité de sélectionner l'objet en question (exemple on choisit "Modèles de fiches" puis on peut choisir "Famille").

Lorsque ce choix est fait l'objet est ajouté à la liste des objets de l'application, ainsi que les dépendances de l'objet. (par exemple j'ajoute "Famille" et ça ajoute automatiquement le bloc de champs "Adresse").

Cette étape passe par définir une API qui devrait être similaire entre applications, pour pouvoir :

- lister les types d'objet dispo
- lister les objets d'un type
- récupérer les dépendances d'un objet

Envisager ça comme une API neuve et indépendante, pas s'obliger à exploiter (/api/agendas/ et /api/formdefs/ et autres, qui ne fourniront pas nécessairement toutes les infos souhaitées).

Aparté : dans la liste des types d'objet on pourrait également avoir "données de ces fiches", pour pouvoir dire "prendre les données de la «table» «écoles»". Cela pour dire aussi qu'on peut dévier un peu de ce qu'on considère habituellement comme "objet" et par exemple avoir "hiérarchie de pages" plutôt que des "pages" individuelles.

Quand la même appli est instanciée 2x (cas portail usager / portail agent) les types se retrouvent doublés dans la liste des types d'objets, (ex: "portail usager - page" & "portail agent - page").

On se trouve là avec une liste type :

- wcs / formulaire / signalement (slug)
- wcs / source de données / type-de-signalement (slug) / (dépendance: vrai)
- wcs / modèle de fiche / fiche-d-intervention / (dépendance: vrai)
- authentic / rôle / responsable signalement / (dépendance: vrai)
- combo:portal-agent / page / statistiques-sur-les-signalements

(on peut imaginer cacher les dépendances par défaut, ou les ranger en fin de liste, mais détail).

Sur cet écran, un bouton "Générer" qui va appeler l'API "export" associée à chacun des objets sélectionnés, et tout taper dans un .zip qui pourra être téléchargé.

Également utile un bouton "Rescanner les dépendances" qui repart des objets qui ne sont pas des dépendances et recalcule toutes celles-ci.

Installation

C'est une boîte de dialogue avec un champ fichier et où on peut mettre le zip de l'application. (plus tard ça pourra interroger un ou plusieurs catalogues pour avoir une liste).

Sur validation, il y a découpe du zip et envoi des parties adéquates aux différentes applications.

De manière basique/initiale, ça correspondrait dans chaque application à une API qui serait équivalente à la commande "import_site".

Catalogue

Il devient possible de transformer un Hobo en catalogue, ça pourrait simplement se faire avec un flag "publier dans le catalogue" (et une API qui listerait les applications et un combo qui afficherait ça proprement).

Chronologie

- les API "liste des types d'objet", "liste des objets d'un type", "dépendances d'un objet"
 - en les implémentant dans une ou deux applications (typiquement w.c.s. et combo)
- l'écran de création (modèle "Application" et modèle "Élément d'une application").
- l'action "Générer l'application" (i.e. les API d'export)
- l'installation (i.e. le push vers les API "import-site")
- l'extension pas à pas à de nouveaux types d'objet
- l'exposition sous forme de catalogue
- l'évolution dans les modules pour marquer ce qui vient d'une application / n'est pas éditable / etc.

Demandes liées:

Lié à Hobo - Development #60699: écrans d'assemblage et d'installation d'une ...	Fermé	16 janvier 2022
Lié à w.c.s. - Development #60698: API export/import pour la création et l'in...	Fermé	16 janvier 2022
Lié à Combo - Development #60773: API export/import pour la création et l'ins...	Fermé	18 janvier 2022

Historique

#1 - 02 novembre 2021 10:08 - Frédéric Péters

- *Sujet changé de "Applification" des modules (applications métiers) Publik fait avec des fiches à "Applification" des modules (applications métiers) Publik*

- *Description mis à jour*

#2 - 02 novembre 2021 10:57 - Frédéric Péters

- *Description mis à jour*

#3 - 22 novembre 2021 16:10 - Mikaël Ates (de retour le 29 avril)

- *Lié à Development #56519: Ajouter la possibilité de faire une sauvegarde de page (poser une étiquette sur une version) ajouté*

#4 - 21 décembre 2021 17:22 - Brice Mallet

- *Version cible changé de 2021 à 2022*

#5 - 16 janvier 2022 19:23 - Frédéric Péters

- *Lié à Development #60699: écrans d'assemblage et d'installation d'une application ajouté*

#6 - 16 janvier 2022 19:23 - Frédéric Péters

- *Lié à Development #60698: API export/import pour la création et l'installation d'applications ajouté*

#7 - 18 janvier 2022 14:11 - Frédéric Péters

- *Lié à Development #60773: API export/import pour la création et l'installation d'applications ajouté*

#8 - 28 septembre 2022 12:32 - Brice Mallet

- *Tags mis à applification*

#9 - 05 novembre 2022 09:15 - Frédéric Péters

- *Lié à Development #56519: Ajouter la possibilité de faire une sauvegarde de page (poser une étiquette sur une version) supprimé*

#10 - 13 avril 2023 17:19 - Mikaël Ates (de retour le 29 avril)

- Statut changé de Nouveau à En cours