

Mandaye - Autre #9518

MandayeJS : Structure des "Sites"

05 janvier 2016 19:15 - Josué Kouka

Statut:	Rejeté	Début:	05 janvier 2016
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Non		

Description

Le mandayeJS a actuellement la structure suivante

```
mandayejs
├── locale
│   └── fr
│       └── LC_MESSAGES
├── mandaye
│   ├── management
│   │   └── commands
│   ├── migrations
│   ├── static
│   └── templates
│       ├── mandaye
│       └── registration
└── sites
    ├── montpellier3m
    │   └── static
    │       └── montpellier3m
    │           └── mediatheques
    │               ├── css
    │               ├── fonts
    │               ├── images
    │               └── js
    └── vincennes
        └── static
            └── vincennes
                ├── biblio
                │   ├── css
                │   ├── font
                │   ├── images
                │   └── js
                └── conservatoire
                    ├── css
                    ├── font
                    ├── images
                    └── js
```

Soit un **dossier client** contenant les **statics** de chaque **applications** Dans le cas où il faudrait développer un **webservice** par **application**, le mieux serait de :

- Avoir un seul **urls.py** et un seul **views.py** par client, avec dans **urls.py**, comme dans passerelle, un mécanisme active l'url si un paramètre du type **SITE_API_<VINCENNES_CONSERVATOIRE>_ENABLED = True**
- Avoir dans **sites**, des app django par applications (dueonext, archimed, ...) contenant un **urls.py** et **views.py** en plus des statics pour chaque client utilisant cet application. En résumé j'aurai pas exemple une application django **duonext**, **archimed** ...

?

Historique

#1 - 06 janvier 2016 00:00 - Benjamin Dauvergne

J'ai du rater un truc, pourquoi est-ce qu'on se sert de static différents pour chaque client ? Pour moi il n'y a qu'un JS plus ou moins spécifique par client, il y a plus ? S'il y a du style il faudrait essayer de le garder coté publik-base-them.

De quel web service par client parle-t-on: des proxys de web-service pour récupérer la liste des prêts comme à montpellier ou d'autre chose ?

#2 - 06 janvier 2016 09:53 - Josué Kouka

J'ai du rater un truc, pourquoi est-ce qu'on se sert de static différents pour chaque client ? Pour moi il n'y a qu'un JS plus ou moins spécifique par client, il y a plus ?

Oui il y'a plus. Il y'a :

- Un (ou une liste de) js lié à la barre mandaye par client. Ce js inject le css de la barre mandaye
- Un js lié à la page d'association de compte (si besoiiin)

De quel web service par client parle-t-on: des proxys de web-service pour récupérer la liste des prêts comme à montpellier ou d'autre chose ?

Je parle du **web-service pour récupérer la liste des prêts comme à montpellier**

Ma question de savoir quel serait la meilleure structure pour pouvoir gerer un cas où pour un client on a un web-service par application : 1 web-service pour la mediathèques, 1 web-service pour le conservatoire, ...

#3 - 06 janvier 2016 10:01 - Frédéric Péters

Ma question de savoir quel serait la meilleure structure pour pouvoir gerer un cas où pour un client on a un web-service par application : 1 web-service pour la mediathèques, 1 web-service pour le conservatoire, ...

À cette question, ma réponse c'est qu'on peut zapper le niveau "client" pour avoir juste le niveau "application", i.e. sites/duonet/, sites/whatever/, etc. (et si jamais un duonet se retrouvait ne pas ressembler à l'autre, on ferait duonet-vincennes, etc.). Un seul niveau, donc.

D'un autre côté, traiter le style de la barre Mandaye comme un thème, quelque chose de parallèle, et là il peut y avoir vincennes, montpellier, etc. Et cette partie disparaîtra à un moment, intégrée dans la mécanique générale des thèmes (publik-base-theme et cie).

#4 - 06 janvier 2016 10:09 - Benjamin Dauvergne

Josué Kouka a écrit :

J'ai du rater un truc, pourquoi est-ce qu'on se sert de static différents pour chaque client ? Pour moi il n'y a qu'un JS plus ou moins spécifique par client, il y a plus ?

Oui il y'a plus. Il y'a :

- Un (ou une liste de) js lié à la barre mandaye par client. Ce js inject le css de la barre mandaye

Ça devrait être le même js partout, seule la CSS devrait différer.

- Un js lié à la page d'association de compte (si besoiiin)

Idem.

De quel web service par client parle-t-on: des proxys de web-service pour récupérer la liste des prêts comme à montpellier ou d'autre chose ?

Je parle du **web-service pour récupérer la liste des prêts comme à montpellier**

Ma question de savoir quel serait la meilleure structure pour pouvoir gerer un cas où pour un client on a un web-service par application : 1 web-service pour la mediathèques, 1 web-service pour le conservatoire, ...

Je penche pour un système de plugin à la passerelle, en essayant de garder l'activation coté hobo pour ne pas avoir à faire de page de gestions dans mandaye, on pourrait s'en sortir avec une simple liste de template comme pour combo (user-portal, agent-portal, etc.. ici on aurai imuse-duonext, archimed, etc..).

Pour chaque type de "mandaye" on active un certain plugin via le choix du template, et donc par exemple on aurait une URL "`^_manday/ws/*$`", qui passerait la requête à l'application activée par le template choisi:

```
from django.apps import apps

def ws(request):
    if settings.MANDAYE_TEMPLATE:
        try:
            app = apps.get_app_config('mandaye.apps.%s' % settings.MANDAYE_TEMPLATE)
        except LookupError:
            pass
        else:
            return app.ws_view(request)
    raise Http404
```

Ça nous force à maintenir coté hobo la liste des applications disponibles dans mandaye, ça ne me parait pas déconnant.

Mais il faut bien voir qu'il a deux axes de configurations:

- le client, qui ne devrait déterminer que le style (CSS donc),
- et l'application, qui va déterminer le mode d'intégration (nom des champs de la page de login, etc..) et les web-services disponibles.

Il faut éviter de mélanger les deux pour ne pas refaire deux fois la même intégration pour la même application dans deux villes différentes. Si j'indique dans hobo que l'application derrière mandaye est un portail duonet ça devrait faire les bons choix de JS et de web-services automatiquement, ensuite ça devrait récupérer le thème depuis publik-base-theme ou le combo portail usager, sur ce point je ne sais pas trop, l'intégration complète du thème n'étant pas généralement possible.

#5 - 06 janvier 2016 11:13 - Josué Kouka

Il y'a sûrement un point que je n'ai pas saisi. Mais il me semble que le style de la barre mandaye est relatif au client.

Ça devrait être le même js partout, seule la CSS devrait différer.

Il y'a bien un js commun (<http://git.entrouvert.org/mandayejs.git/tree/mandayejs/mandaye/static/mandaye.js>) avec une CSS (<http://git.entrouvert.org/mandayejs.git/tree/mandayejs/mandaye/static/mandaye.css>) par défaut. Le(s) js dont j'ai parlé permet(tent) de styliser cette barre mandaye pour le client en question. Dans le cas de Vincennes, il y'a js (duonet.js) qui ajoute un logo et des icones. Ensuite, il y'a le fait que ce sera la meme barre utilisée pour toutes les applications "mandayisés" du client

Idem pour le style de la page d'association

et l'application, qui va déterminer le mode d'intégration (nom des champs de la page de login, etc..) et les web-services disponibles.

Le mode d'integration est défini dans le settings.json. Ce dont je ne suis pas sûre que ce soit relatif à l'application (duonet, imuse) que l'on mandayise. Encore dans le cas de vincennes, je pense qu'ils ont demandé d'ajouter un champs ou que vincennes est sur une version plus vieille (comparaison <http://new.extranet.duonet.fr/Connect.aspx?key=CV4j27Em0dM=> et <https://duonet.fr/> sachant que <http://new.extranet.duonet.fr/> redirige vers <https://duonet.fr/>).

Pour l'instant je ne vois pas trop comment avoir un thème générique où je ne sais juste pas trop comment faire.

#6 - 06 janvier 2016 11:36 - Benjamin Dauvergne

S'il y a des textes à personnaliser, il faut utiliser des variables (settings nommé TEMPLATE_VARS, alimenté par les variables expotés par hobo, et diffusé par un context_processor dans les templates, voir hobo/context_processors.py et hobo/multitenant/settings_loader.py).

Si ce sont des images, il faut utiliser un fichier CSS, mais en aucun cas les templates ne devraient changer. Si tu entends un problème que ces deux solutions ne supportent pas, expose le ici. Le JS ne doit pas servir à personnaliser la barre, juste à définir son comportement et ce soit donc être le même partout.

Normalement les paramètres d'intégration sont uniquement fonction de l'application (et éventuellement de sa version, mais je n'ai rien contre des /apps/duonet2014, /apps/duonet2015, etc..), tu pourrais avoir des AppConfig de cette forme:

```
class Duonet2014AppConfig(AppConfig):
    AUTH_CHECKER = ...
    CSS = ...
    JS = ...
    etc...
```

ça enlèvera une bonne partie de la configuration manuelle sur ces histoires dans les settings, le choix d'utiliser un module en particulier pourra être fait en fonction soit d'une variable de settings explicite MANDAYE_TEMPLATE soit d'une variable template_name dans settings.TEMPLATE_VARS; comme cela on gère un fonctionnement manuel et via hobo.

#7 - 12 janvier 2016 12:00 - Josué Kouka

Dans TEMPLATE_VARS (client)

```
* SITE_CA_URL
* SITE_SCRIPTS
* SITE_ASSOCIATE_STATIC
* SITE_STATIC_ROOT_PATH
```

Dans AppSettings (application)

```
* SITE_LOGIN_PATH
* SITE_HOME_PATH
* SITE_LOCATORS
* SITE_AUTH_CHECKER
* SITE_AUTH_COOKIES
* SITE_FORM_SUBMIT_ELEMENT
```

#8 - 12 janvier 2016 12:33 - Josué Kouka

Application

- duonet
- archimed
- argpege
- imuse

#9 - 18 juillet 2022 21:24 - Frédéric Péters

- Statut changé de Nouveau à Rejeté