

Authentic 2 - Development #24439

Nouvelle interface de mot de passe

11 juin 2018 17:31 - Anonyme

Statut:	Fermé	Début:	
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	100%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
À développer en une ou plusieurs étapes/tickets de développement, nous visons un nouveau widget qui :			
<ul style="list-style-type: none">est un widget Djangoindique la force du mot de passe selon un dictionnaire de règles (js)teste la validité du mot de passe et la double saisie dès la complétion des champs (js)permet d'afficher le dernier caractère à la frappe (cf. https://css-tricks.com/examples/iPhonePassword/)			
En plus selon le délai de livraison et le test du résultat précédent :			
<ul style="list-style-type: none">propose une version mobile au comportement différent: 1 seul input, mais possibilité de l'afficher en clair			
Les 3 tickets client liés sont : https://dev.entrouvert.org/issues/22486 , https://dev.entrouvert.org/issues/21965 , https://dev.entrouvert.org/issues/21916			
Les mockups sont fournis aussi par le projet CUT.			
Pour tester localement:			
* modifier le code de <code>valid_token()</code> pour accepter tout :			
<pre><pre>request.token = { 'email': 'toto'}</pre></pre>			
* <code>authentic2-ctl runserver 9876</code>			
* <code>http://127.0.0.1:9876/accounts/register/azerty</code>			
Demandes liées:			
Lié à Intégrations graphiques Publik - Bug #24767: forms: ne pas forcer width...		Rejeté	25 juin 2018
Lié à Authentic 2 - Development #24833: Adopter un mode de validation des mot...		Fermé	27 juin 2018
Lié à Authentic 2 - Development #25045: Nouvelle interface de mot de passe po...		Fermé	04 juillet 2018
Lié à Publik - Bug #25408: Les pictos et les messages de la nouvelle interfac...		Fermé	23 juillet 2018

Révisions associées

Révision da0ab04a - 20 juillet 2018 17:07 - Benjamin Dauvergne

templates: output form.media in base template (#24439)

It's the new habit downstream.

Révision 93457ecf - 20 juillet 2018 17:07 - Benjamin Dauvergne

api: do not do CSRF check on validate-password API (#24439)

This API is public.

Révision e32621c1 - 20 juillet 2018 17:07 - Benjamin Dauvergne

api: allow empty password in validate-password (#24439)

Révision 4663e217 - 20 juillet 2018 17:07 - Benjamin Dauvergne

create authentic2.forms package (#24439)

Révision d3655cf1 - 20 juillet 2018 17:07 - Benjamin Dauvergne

move authentic2.widgets to authentic2.forms.widgets (#24439)

Révision 6a44c5f5 - 20 juillet 2018 17:07 - Benjamin Dauvergne

move all password related functions in authentic2.passwords (#24439)

Révision f36b4804 - 20 juillet 2018 17:07 - Benjamin Dauvergne

add new widget and fields for passwords (#24439)

Révision c46822af - 20 juillet 2018 17:07 - Benjamin Dauvergne

use new password fields in registration form (fixes #24439)

Révision 1020d50e - 20 juillet 2018 17:07 - Benjamin Dauvergne

update french translation (#24439)

Révision 031db4a7 - 06 août 2018 11:49 - Benjamin Dauvergne

add jquery to password widgets medias (#24439)

Historique

#4 - 11 juin 2018 20:42 - Benjamin Dauvergne

On est pas obligé de faire exactement la demande du CUT je pense, si on par exemple l'affichage du dernier caractère sur tous les environnements, c'est suffisant, pas besoin de l'affichage en clair.

#5 - 11 juin 2018 21:36 - Mikaël Ates

Benjamin Dauvergne a écrit :

On est pas obligé de faire exactement la demande du CUT je pense, si on par exemple l'affichage du dernier caractère sur tous les environnements, c'est suffisant, pas besoin de l'affichage en clair.

C'est bien ce qui est souhaité pour le CUT également.

#6 - 12 juin 2018 00:13 - Benjamin Dauvergne

Mikaël Ates a écrit :

Benjamin Dauvergne a écrit :

On est pas obligé de faire exactement la demande du CUT je pense, si on par exemple l'affichage du dernier caractère sur tous les environnements, c'est suffisant, pas besoin de l'affichage en clair.

C'est bien ce qui est souhaité pour le CUT également.

Yep j'avais vu tes derniers commentaire, je dis ça juste pour qu'Elias ne prenne pas le mockup au pied de la lettre.

#7 - 12 juin 2018 08:57 - Anonyme

- *Description mis à jour*

J'ai modifié la description selon les remarques.

#8 - 13 juin 2018 18:17 - Anonyme

- *Statut changé de Nouveau à En cours*

Travail en cours sur wip/passwords (pour le moment c'est pas présentable pour relecture svp attendre un peu)

#9 - 14 juin 2018 18:15 - Anonyme

- *Fichier 0001-create-password-validation-in-registration-completio.patch ajouté*

- *Patch proposed changé de Non à Oui*

#10 - 14 juin 2018 18:15 - Anonyme (Notes privées)

J'ai poussé dans wip/passwords le même commit (exception de valid_token()) que le patch ci-joint. Ce n'est pas un patch définitif, je dois encore nettoyer ce qui peut l'être, ajouter des tests, et aussi savoir comment ajouter les traductions pour des messages dans validators.py (j'y arrive pas pour le moment, makemessages et compilemessages ne traduit pas les help_text des widgets PasswordInput, si quelqu'un a un conseil...).

Mais si vous avez des remarques de "design", la direction donnée est bonne ?

Pour tester localement:

- j'ai modifié valid_token() pour accepter tout :

```
request.token = { 'email': 'toto' }
```

- authentic2-ctl runserver 9876
- charger <http://127.0.0.1:9876/accounts/register/azerty>

Le patch présente le principe que je voudrais proposer, afin d'avoir une amélioration du formulaire d'activation de compte pour A2, et prévoir les prochaines personnalisations comme potentiellement dans cut-publik-theme.

- pour activer les fonctions de password.js, il faut ajouter les bons attributs au <form>.
- pour que le JS comprenne quelles input et help_text cibler dans ses event handler, on ajoute des classes depuis le Form (RegistrationCompletionForm)
- les paramètres côté JS et côté serveur viennent du même app_settings (A2_PASSWORD_POLICY_*), passés par data-password-policy
- les classes de Form Django qui, comme RegistrationCompletionForm() veulent reprendre le même principe peuvent simplement reprendre les classes des inputs et des help_text.
- style.css : ajout de classes précises qui seront facilement surchargeables.

#11 - 15 juin 2018 07:58 - Anonyme

- Patch proposed changé de Oui à Non

#12 - 15 juin 2018 07:58 - Anonyme (Notes privées)

J'enlève le "rustine proposé" comme dit dans mon dernier commentaire, c'est un travail en cours que je partageais

#13 - 15 juin 2018 12:08 - Anonyme

- Fichier 0001-add-password-validation-UI-in-registration-view-2443.patch ajouté

- Fichier Capture d'écran 2018-06-15 à 11.40.36-fullpage.png ajouté

- Fichier Capture d'écran 2018-06-15 à 11.38.35-fullpage.png ajouté

- Fichier Capture d'écran 2018-06-15 à 11.38.29-fullpage.png ajouté

- Fichier Capture d'écran 2018-06-15 à 11.36.25-fullpage.png ajouté

- Patch proposed changé de Non à Oui

#14 - 15 juin 2018 12:08 - Anonyme (Notes privées)

Voilà un patch, qui sera suivi d'un autre avec des tests et des traductions.

Le principe reste presque le même, mais désormais il y a presque aucune des options de validation/affichage/égalité des mots de passe du Form RegistrationCompletionForm, car elles sont activées et configurées dans le template src/authentic2/templates/registration/registration_completion_form.html

Je vous commente le template dans le patch proposé.

```
<form method="post" class="showPassword passwordEquality" # classes pour l'activation des fonctionnalités
  data-password-policy='{% autoescape off %}{{ passwordpolicy }}{% endautoescape %}' # json venant d'app
_settings.A2_PASSWORD_*
  data-password-policy-input-name='password1' # un paramètre de la fonction 'validate'
  data-password-show-input-name='password1'> # un paramètre de la fonction 'show password'
```

Ce principe servira à configurer d'autres templates (ou une surcharge de ce même template dans un thème) et d'autres formulaires ayant des PasswordInput si besoin à l'avenir.

Je joins des captures.

Je poursuis le travail sur des tests et des traductions

#15 - 15 juin 2018 12:13 - Frédéric Péters

elles sont activées et configurées dans le template src/authentic2/templates/registration/registration_completion_form.html

Je ne pige pas cette partie, je lis le patch et ça continue à se baser sur `app_settings.A2_PASSWORD_POLICY_MIN_LENGTH` & friends et c'est très bien ainsi, c'est pour moi important que ça reste configurable via les settings, qu'on n'exige pas la mise en place d'un template pour modifier la politique de mot de passe.

#16 - 15 juin 2018 12:17 - Anonyme

- Description mis à jour

#17 - 15 juin 2018 12:23 - Anonyme

Frédéric Péters a écrit :

elles sont activées et configurées dans le template `src/authentic2/templates/registration/registration_completion_form.html`

Je ne pige pas cette partie, je lis le patch et ça continue à se baser sur `app_settings.A2_PASSWORD_POLICY_MIN_LENGTH` & friends et c'est très bien ainsi, c'est pour moi important que ça reste configurable via les settings, qu'on n'exige pas la mise en place d'un template pour modifier la politique de mot de passe.

C'était pas clair "elles".

Ce sont bien les `app_settings` qui sont mis dans le contexte (voir `src/authentic2/registration_backend/views.py`) du template (`registration_completion_form.html`)

Le `<form>` du template utilise ces settings et configure, à son tour, le comportement de `password.js`, ce qui permet de varier les types et la configuration de la validation côté interface.

Tu peux tester : modifie `app_settings.py`, et ça sera appliqué dans l'interface; puis supprime des classes du `<form>` dans le template et tu verras les validations disparaître de l'interface.

#18 - 15 juin 2018 12:29 - Frédéric Péters

Tu peux tester : modifie `app_settings.py`, et ça sera appliqué dans l'interface; puis supprime des classes du `<form>` dans le template et tu verras les validations disparaître de l'interface.

Je ne veux pas modifier de fichier `.py`, encore moins éditer un template.

Je veux comme aujourd'hui pouvoir poser dans le `settings.json` `A2_PASSWORD_POLICY_MIN_LENGTH` (& friends) et que ça ait un impact, sans rien faire d'autre. De manière très curieuse j'ai le code que je lis comme suivant ça (très bien) mais dans le même temps je lis tes commentaires partant dans une direction contraire, invitant à modifier le template.

#19 - 15 juin 2018 12:38 - Benjamin Dauvergne

J'aimerais que le widget soit un peu plus "contenairisé", i.e. `passwordpolicy` passé comme paramètre du widget, fichier `.css` séparé déclaré dans Media, on ne devrait pas communiquer via les templates, uniquement dans les Form et la déclaration des widgets.

#20 - 15 juin 2018 13:39 - Anonyme

Frédéric Péters a écrit :

Tu peux tester : modifie `app_settings.py`, et ça sera appliqué dans l'interface; puis supprime des classes du `<form>` dans le template et tu verras les validations disparaître de l'interface.

Je ne veux pas modifier de fichier `.py`, encore moins éditer un template.

Je veux comme aujourd'hui pouvoir poser dans le `settings.json` `A2_PASSWORD_POLICY_MIN_LENGTH` (& friends) et que ça ait un impact, sans rien faire d'autre. De manière très curieuse j'ai le code que je lis comme suivant ça (très bien) mais dans le même temps je lis tes commentaires partant dans une direction contraire, invitant à modifier le template.

Oui on parle de la même chose, tu peux changer `A2_PASSWORD_POLICY_MIN_LENGTH` dans `settings.json`, c'est fait pour

#21 - 15 juin 2018 13:46 - Anonyme

Benjamin Dauvergne a écrit :

J'aimerais que le widget soit un peu plus "contenairisé", i.e. `passwordpolicy` passé comme paramètre du widget, fichier `.css` séparé déclaré dans Media, on ne devrait pas communiquer via les templates, uniquement dans les Form et la déclaration des widgets.

J'avais commencé en mettant plein d'attributs dans le python mais j'ai fini par mettre les paramètres dans le tpl parce que je ne voyais pas d'autre

moyen de personnaliser l'UI dans des projets comme le CUT ou autres ("de l'interface dans l'interface")

Mais je retiens d'encapsuler plus dans le Form : `ctx['passwordpolicy']`, `css`.

#22 - 15 juin 2018 16:06 - Anonyme

Je note ici la nécessité d'ajouter la fonction d'affichage du dernier caractère à la frappe.

Je note aussi l'idée de benj de : "si on veut personnaliser le widget le mieux c'est que le widget se base sur un template; donc template + media(JS + CSS), Il me semble qu'en Django 1.11 ça devient le fonctionnement normal, il faut peut-être voir à ne pas bloquer cette évolution aussi"

Je ne vois pas bien encore pourquoi je modifierais le widget `PasswordInput` ou son fragment d'HTML associé.

Je pense toutefois qu'il y a une confusion sur mon code dans le dernier patch: je ne touche pas aux `<input>` ni aux Widgets django, le template contient toujours intact :

```
{% csrf_token %}
{{ form.as_p }}
```

Je demande seulement au Form django de transmettre les `app_settings` nécessaires. Le JS fait tout le reste en fonction de la présence ou non de classes sur le `<form>`, qui est une balise qu'on aura toujours dans les templates où on injectera des `{{ form }}`.

À suivre je vais voir ce que je peux faire.

#23 - 18 juin 2018 12:44 - Benjamin Dauvergne

Classe sur le form ? Sur le input non ? Sachant qu'avec `.as_p`, `.as_ul` et `.as_table` on est pas certain d'avoir toujours le même arbre HTML en dehors du input, il faut tenter d'éviter au maximum de dépendre de ce qu'il y a autour.

Dans mon idée on aurait le pseudo-code suivant pour un formulaire de changement de mot de passe/définition de mot de passe:

```
old_password = CharField(widget=NewPasswordInput(show_lastchar=True))
new_password1 = CharField(widget=NewPasswordInput(show_lastchar=True, must_not_match="old_password", valida
tions=[...])) <-- là voir comment avoir du code qui fournisse
ce qu'il faut coté validation client et validation serveur, pour indiquer à l'utilisateur ce qui es
t attendu et pour remonter les erreurs rencontrées)
new_password2 = CharField(widget=NewPasswordInput(show_lastchar=True, must_match="new_password1"))
```

J'ai ajouté le `must_not_match` ce n'est pas grand chose et ça me parait logique de pouvoir faire cela.

Ce serait très difficile que `must_match/must_not_match` génère la validation côté serveur (, il faudra continuer à faire cela dans la méthode `clean()` du Form; on atteint la limite de ce que l'API des Form/Field/Widget permet (anecdote: en DRF c'est plus sympa, dans tout Field on peut remonter au `serializer` via `.parent` et donc aux Field adjacents).

#24 - 18 juin 2018 13:32 - Anonyme

Benjamin Dauvergne a écrit :

Classe sur le form ? Sur le input non ? Sachant qu'avec `.as_p`, `.as_ul` et `.as_table` on est pas certain d'avoir toujours le même arbre HTML en dehors du input

Classes sur le form ? : je m'explique, dans mon patch on trouve dans l'HTML modifié :

```
<form method="post" class="showPassword passwordEquality" ...> # classes pour l'activation des fonctionnalit
és
```

OK pour ne pas dépendre de l'arbre HTML, mais seulement de la présence d'un `<form>` et d'`<input>`. C'est ce que je proposais ici : tant que tu mets un `<form>` et `{{ form.as_* }}` dans ton template, le validateur JS fonctionne.

En réfléchissant à ta proposition, je me dis qu'elle demande de modifier les Widget des Fields côté Django, alors que je crois pour l'heure qu'on en a pas besoin. **Mais** je dois ajouter à mon patch l'affichage du dernier caractère tapé, et ça changera peut-être ma vision des choses... à suivre.

#25 - 21 juin 2018 14:32 - Anonyme

Une annonce sur la suite du code pour `password.js`

J'ai essayé d'implémenter l'option 2 présentée ici <https://css-tricks.com/examples/iPhonePassword/>, mais c'est trop complexe et tarabiscoté. Ça rend le code illisible et il faut gérer un nombre inconnu de situations particulières, `ctrl-c`, `del`, `suppr`, `inser`, etc.

Et je propose l'implémentation 1 (sans l'effet flash "wow" mais simplement le dernier caractère humblement affiché sur la droite de l'input)

Je trouve que l'option 2 trop risquée car il faut maintenir une version "fantôme" de l'input originale qui suivent les événements du clavier, et le code démo `jquery.dPassword.js` malgré sa complexité est buggué et non-maintenu. Par exemple, comment savoir quel caractère à supprimé avec la touche `Del` ?. Ensuite une fois le faux "input" visible, il faut dire à tous les autres validateurs de s'adresser à l'input caché, ce qui complexifie énormément le code.

Voilà. Le dernier caractère va s'afficher à côté et l'utilisateur aura presque la même chose à mon avis

#26 - 21 juin 2018 16:06 - Anonyme

- Fichier 0001-add-password-validation-interfacein-registration-for.patch ajouté

Meilleure intégration aux Widgets (avec template pour prévoir django 1.11), aux app_settings, et aux Form django
Ajout de l'affichage du dernier caractère.
J'espère que ça colle aux attentes.

Après côté calendrier, pour une release espérée le 28/6, on pourra faire des "fixes" la semaine de freeze à partir de demain matin si ce patch est accepté.

#27 - 21 juin 2018 16:31 - Frédéric Péters

Ta css alterne les niveaux d'indentation, et après les références à tes caractères de FontAwesome on ajoute en commentaire le nom de l'icône. (je laisse à d'autres les commentaires moins bateau).

#28 - 21 juin 2018 17:09 - Frédéric Péters

À tester la branche, <https://perso.entrouvert.org/~fred/tmp/20180621-wip-passwords.ogv> pour le comportement chez moi. Aussi je m'attendais à avoir ça également sur la page de modification du mot de passe. (et aussi à la saisie sur la page de connexion même si je reconnais l'affaire moins tranchée là) (et côté backoffice).

Côté code, autre commentaire de style en passant, les templates devraient selon moi être rangés sous authentic2/, pas sous registration/ (certains templates y étant pour raison historique).

#29 - 21 juin 2018 17:44 - Anonyme

- Fichier 0001-create-AssistedPassword-AssistedPasswordFormMixin-an.patch ajouté

- correction des remarques précédentes (CSS)
- AssitedPassword mieux documenté
- ajouté AssitedPasswordFormMixin pour prévoir les ajouts futurs (une fois discutés) de ce système ailleurs que lors de la création de compte.
 - PasswordChangeForm, SetPasswordForm ? Il faudrait changer le nom des attributs à new_password1 et 2 en faisant un peu de meta programmation le jour venu

#30 - 21 juin 2018 19:25 - Anonyme

- Fichier 0001-create-AssistedPassword-AssistedPasswordFormMixin-an.patch ajouté

- correction du double bouton "eyes", j'utilise des identifiants uniques maintenant
- homogénéisation des classes css "a2-*" au passage

#31 - 21 juin 2018 23:14 - Mikaël Ates

Je me base uniquement sur la vidéo postée par Fred. Je vois 2 yeux précédés du caractère affiché en clair de manière très furtive. Je trouve que ces yeux ce n'est vraiment pas beau.

Ne peut-on pas afficher le dernier caractère en clair en fin de champs comme cela était demandé ?

Enfin, il faut que le caractère soit affiché en clair plus longtemps.

Pour le message indiquant que le mot de passe respecte la politique de complexité on est aussi assez loin de ce qui était demandé mais peut-être que c'est traité dans un autre ticket.

Je ne sais pas dans quelle mesure la configuration/l'ajustement via le thème va permettre d'obtenir les résultats attendus, à savoir les mockups des 3 tickets CUT liés, mais là ça me paraît vraiment loin.

#32 - 22 juin 2018 08:18 - Frédéric Péters

Sur les doutes, je me demande si ce ticket pourrait être divisé ainsi :

- indication live de la validation des critères (en gros, scénarios 2&4 du document "Éléments interfaces CUT") (et là je pense qu'on pourrait reprendre davantage de l'UI, ne pas essayer de présenter ça sous forme de phrase "Your password must contain..."). (même si ça doit sans doute se limiter à deux éléments, longueur et "3 types de caractères", pour faire avec le code de validation actuel)
- œil d'affichage du mot de passe (scénario 3) (perso j'étais surpris par le côté affichage uniquement le temps où le bouton est pressé, plutôt qu'un truc on/off)
- affichage lors de la frappe du dernier caractère, selon possibilité technique, etc. (qui est chouette etc. mais a plutôt sa place dans le code du navigateur, pour ne pas obliger à jongler comme tu le décris dans le commentaire 25).

#34 - 22 juin 2018 12:24 - Benjamin Dauvergne

Paul Marillonnet a écrit :

Mikaël Ates a écrit :

Ne peut-on pas afficher le dernier caractère en clair en fin de champs comme cela était demandé ?
Enfin, il faut que le caractère soit affiché en clair plus longtemps.

(Je ne fais que passer) Le comportement usuel, je crois, c'est que le dernier caractère est affiché en permanence, non ? (il ne s'affiche plus en clair au moment où il devient l'avant-dernier)

Oui mais c'est quasi-impossible à faire avec un simple input password sans créer notre propre widget complet et gérer les frappes (ce dont parle Elias plus haut). Son implémentation me va mais je verrai plutôt une surimpression avec fading en milieu de page et en gros (genre 20vh en taille) et en 400ms.

Commentaires sur le patch:

Pour la validation des critères, l'implémentation actuelle va rendre vraiment difficile de faire évoluer les critères, je verrai plutôt l'utilisation de requête ajax sur un web-service public de validation, on passerait juste l'URL de ce endpoint au widget.

```
POST /api/password-validate/ HTTP/1.1
Content-Type: application/json
```

```
{"password": "Awhatever"}
200 Ok
```

```
Content-Type: application/json
```

```
{"err": 0, "data": {
  "rules": [
    {
      "label": "un caractère majuscule",
      "validate": true
    },
    {
      "label": "un chiffre",
      "validate": false
    }
  ]
}}
```

Le paramétrage du temps d'affichage du dernier caractère me semble inutile, il faut juste trouver la bonne durée.

Je ne comprends pas ce que vient faire l'oeuil ou le setting pour afficher le mot de passe en clair, il me semblait qu'on avait exclu complètement ce fonctionnement.

Pour l'affichage des critères plutôt qu'en texte je verrai bien une popup en bord de widget tant que celui-ci a le focus, genre ça:

```
Nouveau mot de passe : [XXXX_] <-| .-----\
| OK critère 1 |
| NOK critère 2 |
| OK critère 3 |
\_____;
```

On doit pouvoir gérer le placement du div popup avec la CSS, en gérer la présentation etc.. (pseudo-sélecteur :focus ou :focus-within). Aussi laisser une classe sur le widget quand il perd le focus pour bien marquer qu'il n'est pas valide.

Ça serait bien de jouer dans un fiddle pour simplifier les tests et voir qu'on est bien tous d'accords, j'ai fait ça par exemple :

<https://jsfiddle.net/r91k5quf/43/>

#35 - 22 juin 2018 12:29 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Oui mais c'est quasi-impossible à faire avec un simple input password sans créer notre propre widget complet et gérer les frappes (ce dont parle Elias plus haut). Son implémentation me va mais je verrai plutôt une surimpression avec fading en milieu de page et en gros (genre 20vh en taille) et en 400ms.

Et je n'ai pas du tout fait ça dans mon fiddle (je l'ai fait au début) et c'est mieux comme ça, donc padding 2ex à droite et y afficher le caractère, je reprends font-size et line-height de l'input.

#36 - 22 juin 2018 12:58 - Anonyme

- Fichier 1bb8edd1cc8e4dfb2443d8bb3f75459.mp4 ajouté

- Fichier 0001-create-AssistedPassword-AssistedPasswordFormMixin-an.patch ajouté

Le thème de cette vidéo me semble particulier

C'est un thème Publik tout à fait normal.

#38 - 22 juin 2018 13:21 - Anonyme

(ce commentaire s'est croisé avec le dernier de benj, donc ça n'apporte pas encore toutes les réponses)

Je me rend compte que j'ai très peu écrit ce que faisait ce patch. J'espère que les doutes se lèveront après ces explications.

- La vidéo postée par Fred contient un bug, qui fait un double affichage du bouton "œil", et qui a été réglé dans le patch proposé suivant. Le thème de cette vidéo me semble particulier et provoque un affichage lié au thème. Je joins une autre vidéo de la dernière version "brute" sans thème particulier. On pourrait aussi activer cette fonctionnalité au cas par cas par déploiement, c'est à discuter.
- Chaque fonctionnalité d'assistance au mot de passe peut-être activée ou non pour un tenant et personnalisée par son thème, ou affichée différemment en surchargeant le template. Après s'il n'y a pas assez de configuration, on peut en rajouter.

Fonctionnalités

Le bouton "œil" d'affichage du mot passe en entier.

- il est vrai que c'est en extra, mais c'est devenu commun sur le web, et je le voyais mal ne pas proposer l'option, et le code est trivial.
- fonctionnalité : ce n'est prévu que pour l'affichage du mot de passe entier pendant le temps du clic. Le danger ici c'est que quelqu'un voit le mot passe en clair si on oublie de re-cliquer un 2nde fois pour le cacher.
- configuration : la variable de serveur `A2_PASSWORD_DISPLAY_SHOW_ALL` permet de l'activer ou non sur un tenant (voir `app_settings.py`).
- apparence : le bouton est contenu dans une balise `<i>` avec la classe `"a2-show-password-button"` insérée après le champ `password`. On peut avec CSS d'un thème la personnaliser, y compris changer l'icone œil/barré en surchargeant `a2-show-password-button:after` et `hide-password-button:after` (voir `password.css`)

L'affichage du dernier caractère.

- fonctionnalité : contrairement à ce que dit Paul, il s'affiche et s'efface tout seul. Dans mon commentaire #25, et après une tentative de faire fonctionner l'affichage comme le voudrait le mock-up GL (en lieu et place du dernier "point" du champ mot de passe), j'ai préféré abandonner car c'est pas sûr comme fonctionnement, car :
 - on fabrique un faux champ mot de passe,
 - on met la vraie valeur dans un champ caché, ce qui tord complètement la sémantique HTML (qui permet par exemple de demander à son navigateur de retenir son mot de passe dans un trousseau),
 - on génère à priori un bonne centaine de lignes de code JS difficilement maintenable, le serveur envoie de l'HTML qui est complètement modifié côté client.
 - d'où le choix de l'option d'affichage sur le côté du dernier caractère tapé.
- configuration : la variable de serveur `A2_PASSWORD_DISPLAY_LAST_CHAR_DURATION` si elle est `False` désactive l'interface, et sinon donne la durée en millisecondes d'affichage du caractère, ici par défaut très court 300ms, que j'ai remonté à 1000ms dans ce patch.
- apparence : le `<div>` avec la `a2-password-show-last` est personnalisable mais je propose ici une solution qui je crois sera la plus sûre mais on pourra, dans un thème ou tout de suite besoin, le mettre en gros au milieu par exemple. Il y a un fade-in, pause de `A2_PASSWORD_DISPLAY_LAST_CHAR_DURATION` et fade-out. La proposition faite par Benj dans le <https://jsfiddle.net/r91k5quf/43/> ne fonctionne pas universellement. Un élément HTML en `position: absolute` se positionne par rapport au plus proche parent en `position: relative` et ça on ne le contrôle pas. On ne peut pas forcer l'ajout d'un `position: relative` sur l'élément html parent, car il risquerait de sauter dans un thème ou un autre. D'ailleurs, ne fonctionne pas dans le cas d'un `{{ form.as_p }}` simple, car l'offset est alors complètement faux et le caractère est affiché n'importe où. Quid des autres formes de `{{ form }}`, etc...

L'affichage de la politique de composition d'un mot de passe.

- fonctionnalité : en fonction des options activées par le serveur, on vérifié séparément ou ensemble le nombre mini de "classes" de caractères, et/ou la longueur minimale du mot de passe, et/ou la correspondance avec une expression régulière. Fonctionnalité déjà implémentée côté serveur avant ce patch, et qui sont juste exposées côté UI.
- configuration : ce sont les mêmes variables python qu'avant ce patch `A2_PASSWORD_POLICY_MIN_LENGTH`, `A2_PASSWORD_POLICY_MIN_CLASSES` et `A2_PASSWORD_POLICY_REGEX_ERROR_MSG` qui sont utilisées. Voir dans `validators.py` la fonction `password_help_text` qui compile le template des messages d'indication.
 - je suis allé plus loin dans ce dernier patch et utilise un template django pour l'intégralité du rendu des messages de politique de mot de passe, et qui permettra une surcharge complète de l'HTML dans un serveur avec ses propres bouts de template `authentic` (comme le GLC) : la variable serveur `A2_PASSWORD_POLICY_MESSAGE_TPL` donne le chemin vers le template `authentic2/password_help_text.html` qui remplace l'ancien passage compilé en string python dans `validators.py` donnant encore plus de moyen de surcharger ça dans un thème ou un tenant.
- apparence : Chaque message de politique de mot de passe est dans un `` avec sa propre classe CSS : `a2-min-length-policy`, `a2-min-class-policy`, `a2-regex-policy`. Les icônes "danger" et "ok" avant le texte sont personnalisables avec `password-error:before` et `password-ok:before`. Chaque phrase sera traduite (ici en anglais pour le moment). Je peux essayer la proposition de pop-up de benj qui s'affiche au focus dans le champ de mot de passe, mais ça va prendre du temps et surtout risque de faire disparaître la phrase "neutre" qui est affichée de toute façon au chargement de la page. Ce bout de texte est envoyé par le serveur, et est "décoré" sur place par le JS/CSS.
h2. Vérification d'égalité des mots de passe.

- la fonctionnalité: tant que le 2nd mot de passe est vide, on n'affiche rien, dès qu'on commence à taper le second, la vérification est faite à chaque nouvelle frappe dans le 1er ou le 2nd champs.
- configuration: la variable de serveur
- apparence : les messages "match/not match" sont dans un <div> avec une classe a2-passwords-messages, et chaque message est personnalisable avec la classe a2-passwords-unmatched et a2-passwords-matched. Les icônes "danger" et "ok" avant le texte sont personnalisables avec password-error:before et password-ok:before

#39 - 22 juin 2018 13:45 - Anonyme

- Fichier 0001-create-AssistedPassword-AssistedPasswordFormMixin-an.patch ajouté

Frédéric Péters a écrit :

Le thème de cette vidéo me semble particulier

C'est un thème Publik tout à fait normal.

J'ai trouvé ce qui rendait le patch plus universel : le dernier caractère affiche non pas dans un div, qui obligeait à faire un display:inline, mais dans un b ce qui l'affichera dans le cas normal sur la même ligne que l'input. On contrôle aussi le "helptext" après le champ pour avoir par défaut un saut à la ligne. Mais on peut faire autrement si on travaille la CSS. J'ai mis l'état actuel ici <https://jsfiddle.net/elishowk/w4j30e2v/>

#40 - 22 juin 2018 14:01 - Benjamin Dauvergne

Nouveau fiddle avec uniquement du position: relative,;

<http://jsfiddle.net/bdauvergne/d05es4rq/24/>

#41 - 22 juin 2018 14:53 - Benjamin Dauvergne

Elias Showk a écrit :

Le bouton "œil" d'affichage du mot passe en entier.

- il est vrai que c'est en extra, mais c'est devenu commun sur le web, et je le voyais mal ne pas proposer l'option, et le code est trivial.

Mouais, tant que la base n'est pas là je n'irai pas plus loin, mais bon.

- fonctionnalité : ce n'est prévu que pour l'affichage du mot de passe entier pendant le temps du clic. Le danger ici c'est que quelqu'un voit le mot passe en clair si on oublie de re-cliquer un 2nde fois pour le cacher.

Ok, c'est visible entre le mousedown/mouseup où pendant un temps limité dès le mousedown (genre 800ms) ? Je préfère la deuxième solution, la première est peu "découvrable" d'un point de vue ergonomique, i.e. peu d'utilisateur comprendront qu'il faut rester appuyé.

- configuration : la variable de serveur A2_PASSWORD_DISPLAY_SHOW_ALL permet de l'activer ou non sur un tenant (voir app_settings.py).
- apparence : le bouton est contenu dans une balise <i> avec la classe "a2-show-password-button" insérée après le champ password. On peut avec CSS d'un thème la personnaliser, y compris changer l'icone œil/barré en surchargeant a2-show-password-button:after et hide-password-button:after (voir password.css)

Ok.

L'affichage du dernier caractère.

- fonctionnalité : contrairement à ce que dit Paul, il s'affiche et s'efface tout seul. Dans mon commentaire #25, et après une tentative de faire fonctionner l'affichage comme le voudrait le mock-up GL (en lieu et place du dernier "point" du champ mot de passe), j'ai préféré abandonner car c'est pas sûr comme fonctionnement, car :
 - on fabrique un faux champ mot de passe,
 - on met la vraie valeur dans un champ caché, ce qui tord complètement la sémantique HTML (qui permet par exemple de demander à son navigo de retenir son mot de passe dans un trousseau),
 - on génère à priori un bonne centaine de lignes de code JS difficilement maintenable, le serveur envoie de l'HTML qui est complètement modifié côté client.
 - d'où le choix de l'option d'affichage sur le côté du dernier caractère tapé.

Oui, mais donc j'aimerais que ce soit à l'intérieur de l'input et pas sur le côté, idem pour l'oeuil. On doit garder l'aspect contenant du widget (tout ce qui concerne le mot de passe se trouve dans le cadre).

- configuration : la variable de serveur A2_PASSWORD_DISPLAY_LAST_CHAR_DURATION si elle est False désactive l'interface, et sinon donne la durée en millisecondes d'affichage du caractère, ici par défaut très court 300ms, que j'ai remonté à 1000ms dans ce patch.

Juste True/False pour A2_PASSWORD_DISPLAY_LAST_CHAR l'instant et on trouve la valeur qui va bien, je ne veux pas qu'on joue avec ça. Si c'est en standard et que ça ne reste pas affiché on peut même ne pas rendre ça configurable.

- apparence : le <div> avec la a2-password-show-last est personnalisable mais je propose ici une solution qui je crois sera la plus sûre mais on pourra, dans un thème ou tout de suite besoin, le mettre en gros au milieu par exemple. Il y a un fade-in, pause de A2_PASSWORD_DISPLAY_LAST_CHAR_DURATION et fade-out. La proposition faite par Benj dans le <https://jsfiddle.net/r91k5quf/43/> ne fonctionne pas universellement. Un élément HTML en position: absolute se positionne par rapport au plus proche parent en position: relative et ça on ne le contrôle pas. On ne peut pas forcer l'ajout d'un position: relative sur l'élément html parent, car il risquerait de sauter dans un thème ou un autre. D'ailleurs, ne fonctionne pas dans le cas d'un {{ form.as_p }} simple, car l'offset est alors complètement faux et le caractère est affiché n'importe où. Quid des autres formes de {{ form }}, etc...

J'ai proposé une version en pure relative mais on peut aussi retrouver en JS le plus proche parent "relative" et soustraire son offset à l'offset du champ password, donc ce n'est pas vraiment un problème.

L'affichage de la politique de composition d'un mot de passe.

- fonctionnalité: en fonction des options activées par le serveur, on vérifié séparément ou ensemble le nombre mini de "classes" de caractères, et/ou la longueur minimale du mot de passe, et/ou la correspondance avec une expression régulière. Fonctionnalité déjà implémentée côté serveur avant ce patch, et qui sont juste exposée côté UI.

J'aimerais que ça reste complètement coté serveur.

- configuration : ce sont les mêmes variables python qu'avant ce patch A2_PASSWORD_POLICY_MIN_LENGTH, A2_PASSWORD_POLICY_MIN_CLASSES et A2_PASSWORD_POLICY_REGEX_ERROR_MSG qui sont utilisées. Voir dans validators.py la fonction password_help_text qui compile le template des messages d'indication.
 - je suis allé plus loin dans ce dernier patch et utilise un template django pour l'intégralité du rendu des messages de politique de mot de passe, et qui permettra une surcharge complète de l'HTML dans un serveur avec ses propres bouts de template authentic (comme le GLC): la variable serveur A2_PASSWORD_POLICY_MESSAGE_TPL donne le chemin vers le template authentic2/password_help_text.html qui remplace l'ancien passage compilé en string python dans validators.py donnant encore plus de moyen de surcharger ça dans un thème ou un tenant.

On ne souhaite pas ça, des messages corrects du suffiront.

- apparence : Chaque message de politique de mot de passe est dans un avec sa propre classe CSS: a2-min-length-policy, a2-min-class-policy, a2-regex-policy. Les icônes "danger" et "ok" avant le texte sont personnalisables avec password-error:before et password-ok:before. Chaque phrase sera traduite(ici en anglais pour le moment). Je peux essayer la proposition de pop-up de benj qui s'affiche au focus dans le champ de mot de passe, mais ça va prendre du temps et surtout risque de faire disparaître la phrase "neutre" qui est affichée de toute façon au chargement de la page. Ce bout de texte est envoyé par le serveur, et est "décoré" sur place par le JS/CSS.

Le help_text doit rester un help text i.e. il indique la règle, la popup indique les erreurs pendant la frappe, pas plus.

Vérification d'égalité des mots de passe.

- la fonctionnalité: tant que le 2nd mot de passe est vide, on n'affiche rien, dès qu'on commence à taper le second, la vérification est faite à chaque nouvelle frappe dans le 1er ou le 2nd champs.
- configuration: la variable de serveur

Je ne vois pas d'utilité à ce que ce soit configurable, on va le vouloir partout.

- apparence : les messages "match/not match" sont dans un <div> avec une classe a2-passwords-messages, et chaque message est personnalisable avec la classe a2-passwords-unmatched et a2-passwords-matched. Les icônes "danger" et "ok" avant le texte sont personnalisables avec password-error:before et password-ok:before

Nouveau fiddle avec oeuil, position absolue (parce qu'oeuil) et mousedown / mouseup, mouseleave pour affichage du mot de passe en clair.

<http://jsfiddle.net/bdauvergne/d05es4rq/49/>

#42 - 22 juin 2018 18:28 - Anonyme

- Fichier 0001-create-AssistedPassword-AssistedPasswordFormMixin-24.patch ajouté

Pris en compte les retours de benj.

Ce patch n'est pas encore finalisé, il faut faire un peu de CSS pour afficher à droite une du champ la popup de validation (qui fait appel à une api désormais)

Poussé aussi sur wip/passwords

#43 - 23 juin 2018 08:03 - Benjamin Dauvergne

- Au niveau du JS il faut préférer les handler sur body de la forme:

```
$('#body').on('wvtv', 'form', function () { });
```

plutôt qu'un handler direct `$('#form').on('wvtv', function () { });`; ceci pour se laisser de la marge niveau dynamisme (si un formulaire est introduit dynamiquement dans le DOM).

- Tout le code validation devrait sauter maintenant qu'il est dans le web-service.
- Au niveau du web-service la forme générale est historiquement différente des autres applications, c'est `{result: 1, ...}` et pas `{err:0, ...}`, il faut faire comme cela.
- `get_validation_errors()` c'est mal nommé, plutôt `password_validation()` et il faut en faire une fonction qui retourne les erreurs mais aussi les validations qui passent
- `AssistedPasswordFormMixin` ne sert à rien, à dégager, on déclare les widgets directement sur les Form
- les fichiers media sont à déclarer au niveau du widget
- ça peut être simplifiant de sous classer le widget en 2 autres "pré-configurés", `CheckPasswordWidget` (pour le login, et la confirmation dans `password2`), `NewPasswordWidget` (pour `password1`)
- virer la bidouille dans `registration_backend/views.py` qui crashe les tests
- le deuxième template n'est pas nécessaire, uniquement celui pour le `render()` du widget
- je suis toujours pour ne pas ajouter de settings concernant les IHMs, one size fits all

#44 - 25 juin 2018 14:28 - Anonyme

Notes de la réunion de ce matin

Nouvelles fonctionnalités mot de passe.

* Elias fait des vidéos, mais Mik voudrait voir "en vrai".

** avoir *des* instances de dev par branche "wip"

*** [Fred] de mon côté je pense qu'on peut gentiment poster vers un seul dépôt "wip" et de manière temporaire on installe le paquet wip en question sur le serveur SaaS dev. (#24749)

* soluce "coup d'oeil", sans paquet ni déploiement : visibilité des machines des dev via le VPN (=> fixer les IP sur le vpn, Thomas doit regarder mais n'y croit pas des masses)

* un autre ticket A2 pour faire évoluer la validation de mot de passe côté serveur => y a-t-il le type X de caractère ? oui|non

* côté <https://dev.entrouvert.org/issues/24439> : Elias s'approche de ce fonctionnement (condition de présence d'un type de caractère: oui|non) et laisse le fonctionnement actuel "nombre mini de classes de caractères"

* proposer une alternative à l'oeil pour avoir un bouton on|off

J'ajoute que mon calendrier de travail d'ici le 06 juillet (en congé la semaine suivante) me fait **demander de l'aide pour le second ticket qui s'est profilé** après cette réunion (modification de la politique des mots de passe A2 (dire Vrai/Faux sur la présence d'un type de caractère dans le mot de passe, à la place de Vrai/Faux sur la présence de 3 classes au moins de caractères)

#45 - 25 juin 2018 18:33 - Anonyme

- Lié à Bug #24767: forms: ne pas forcer width: 100% sur les inputs ajouté

#46 - 27 juin 2018 18:32 - Benjamin Dauvergne

- Lié à Development #24833: Adopter un mode de validation des mots de passe basé sur une liste de critères simples ajouté

#47 - 27 juin 2018 18:32 - Benjamin Dauvergne

Ok je prends [#24833](#).

#48 - 28 juin 2018 11:53 - Benjamin Dauvergne

Pensée du jour: attention à l'usage des préfixes par les Form `Form(prefix='myform-')` lorsqu'on indique à `CheckPasswordWidget` le nom du champ à vérifier (il faudra ajouter le préfixe, sauf que je ne sais pas si les widget le reçoivent, j'ai l'impression que Form leur passe directement le nom du champ pré-transformé).

#49 - 29 juin 2018 15:07 - Anonyme

- Fichier `0001-create-assisted-password-input-widgets-24438.patch` ajouté

- Fichier `connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_(2).png` ajouté

- Fichier `connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_(Galaxy S5).png` ajouté

- Fichier `connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_(Galaxy S5) (1).png` ajouté

- Fichier `connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_(Galaxy S5) (2).png` ajouté

- Fichier `connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_(Galaxy S5) (3).png` ajouté

- Fichier `connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_(Galaxy S5) (4).png` ajouté

- Statut changé de En cours à Solution proposée

Benjamin Dauvergne a écrit :

Pensée du jour: attention à l'usage des préfixes par les Form Form(prefix='myform-') lorsqu'on indique à CheckPasswordWidget le nom du champ à vérifier (il faudra ajouter le préfixe, sauf que je ne sais pas si les widget le reçoivent, j'ai l'impression que Form leur passe directement le nom du champ pré-transformé).

Je n'ai pas trouvé comment prefix est utilisé dans la base actuelle de code. avec ModelForm en plus, on ne peut pas passer un "prefix=". Je n'ai pas encore trouvé comment utiliser BaseForm.add_prefix() pour préfixer les identifiants des champs dans les nouveaux Inputs widgets proposés.

Enfin, j'ai corrigé tout ce qui avait été listé avant cette dernière remarque. voici un nouveau patch et des captures (ici avec le thème GL-SAU). Si je trouve comment faire pour prefix, j'ajouterai ça mercredi 4/7 à mon retour au boulot. D'ici là, pour moi tout le reste est OK. On doit voir comment phaser le ticket [#24833](#) pour que la première modif puisse être testé en recette quand même si c'est ok pour vous.

#50 - 29 juin 2018 15:18 - Frédéric Péters

(tu peux pousser tes modifications dans la branche ?)

#51 - 29 juin 2018 16:09 - Benjamin Dauvergne

Elias Showk a écrit :

Je n'ai pas trouvé comment prefix est utilisé dans la base actuelle de code. avec ModelForm en plus, on ne peut pas passer un "prefix=". Je n'ai pas encore trouvé comment utiliser BaseForm.add_prefix() pour préfixer les identifiants des champs dans les nouveaux Inputs widgets proposés.

Il ne l'est quasiment pas, mais clairement ça rend les Form utilisant cette fonctionnalité non réutilisables, il faudra mettre un commentaire là dessus.

Enfin, j'ai corrigé tout ce qui avait été listé avant cette dernière remarque. voici un nouveau patch et des captures (ici avec le thème GL-SAU). Si je trouve comment faire pour prefix, j'ajouterai ça mercredi 4/7 à mon retour au boulot. D'ici là, pour moi tout le reste est OK. On doit voir comment phaser le ticket [#24833](#) pour que la première modif puisse être testé en recette quand même si c'est ok pour vous.

Tu ne pourras rien faire pour préfixe, je viens de relire le code de Django et un widget n'a aucun moyen d'aller voir ça. Au mieux tu pourras gérer ça sur une sous classe de Form spécialisée pour cet usage, mais je ne le ferai pas ici, juste un commentaire précisant cette limitation suffira.

#53 - 04 juillet 2018 12:16 - Anonyme

- Fichier [aab6f794279c44b79542c0487468f689-1530699221194.mp4](#) ajouté

En attendant un patch prochainement qui corrigera les dernières choses (tests, traductions, commentaires), voici une video pour valider l'aspect graphique. Attention cependant, les règles de validation vont évoluer avec le ticket lié que Benjamin a pris, mais le fonctionnement converge vers "respect de la règle X"=> vrai/faux.

#54 - 04 juillet 2018 12:20 - Frédéric Péters

À mon sens les critères doivent être affichés dès le début.

#55 - 04 juillet 2018 12:48 - Anonyme

Benjamin Dauvergne a écrit :

Elias Showk a écrit :

Je n'ai pas trouvé comment prefix est utilisé dans la base actuelle de code. avec ModelForm en plus, on ne peut pas passer un "prefix=". Je n'ai pas encore trouvé comment utiliser BaseForm.add_prefix() pour préfixer les identifiants des champs dans les nouveaux Inputs widgets proposés.

Il ne l'est quasiment pas, mais clairement ça rend les Form utilisant cette fonctionnalité non réutilisables, il faudra mettre un commentaire là dessus.

Ou alors, comme le nom du champ cible ne sert **que** côté password.js, je peux adapter ce code là pour que password2 aille chercher magiquement l'autre "input[type=password]" dans le même form que lui. Je pense que ça couvrira mieux tous les cas d'usages, et évitera d'utiliser l'attribut "name" et donc les préfixes venus de Django.

#56 - 04 juillet 2018 15:02 - Anonyme

- Fichier [e16a3bff245840d184a31bff430a24b0.mp4](#) ajouté

Frédéric Péters a écrit :

À mon sens les critères doivent être affichés dès le début.

Yep c'est mieux comme ça je crois aussi.

Cette démo tourne avec le retrait de l'attribut name pour cibler l'égalité des 2 mots de passe, pour éviter prérequis dans le code des Form Django d'A2 (code que je m'appête à pousser sur la branche wip/passwords)

#57 - 04 juillet 2018 16:36 - Anonyme

Elias Showk a écrit :

Frédéric Péters a écrit :

À mon sens les critères doivent être affichés dès le début.

Yep c'est mieux comme ça je crois aussi.

Cette démo tourne avec le retrait de l'attribut name pour cibler l'égalité des 2 mots de passe, pour éviter prérequis dans le code des Form Django d'A2 (code que je m'appête à pousser sur la branche wip/passwords)

La vidéo ne montre pas le début en fait, désolé : les de règles de mot de passe sont en noir normal sans icône, ils deviennent rouges, gras et avec l'icône danger à la première frappe

#58 - 04 juillet 2018 16:49 - Mikaël Ates

ils deviennent rouges, gras et avec l'icône danger à la première frappe

Est-ce que c'est possible de se mettre à un niveau inférieur sur l'échelle de la terreur, quelque chose qui fasse plus "avertissement" que "danger" ?

Pas exemple avant la frappe c'est grisé, au début de la frappe ça passe en noir (sans gras), quand c'est validé ça passe en vert.

#59 - 04 juillet 2018 18:10 - Anonyme

Mikaël Ates a écrit :

ils deviennent rouges, gras et avec l'icône danger à la première frappe

Est-ce que c'est possible de se mettre à un niveau inférieur sur l'échelle de la terreur, quelque chose qui fasse plus "avertissement" que "danger" ?

Pas exemple avant la frappe c'est grisé, au début de la frappe ça passe en noir (sans gras), quand c'est validé ça passe en vert.

Oui je fais ça

#60 - 05 juillet 2018 12:49 - Anonyme

- Statut changé de Solution proposée à En cours

#61 - 05 juillet 2018 12:53 - Anonyme

- Fichier 0a94b94568d44f60b63fb5cbb99d9023-1530787531991.mp4 ajouté

voilà une version de l'interface améliorée suites aux remarques de Mik.

Même si la vidéo montre l'interface d'un serveur authentic2 "standalone" ça sera la même chose pour un authentic multi-tenant avec un thème, évidemment

#62 - 05 juillet 2018 13:07 - Frédéric Péters

De la vidéo il manque l'information utile (combien de caractères...). (et sur la phrase je serais pour "/contains/must contain/", mais plus encore, je serais pour ne pas verbaliser les critères et plutôt avoir une phrase d'introduction, comme c'est visible dans la maquette CUT ("Pour avoir un mot de passe protégé, veuillez utiliser a minima :").

#63 - 05 juillet 2018 13:10 - Mikaël Ates

Comme il y a la coche verte de succès, peut-être que ce pourrait être une croix lorsque ce n'est pas validé plutôt que le panneau.

Le texte "Your password contains at least characters" devrait être "Your password **must contain** at least **8** characters".

Idem "Your password contains at least classes among" devrait être "Your password **must contain** at least 3 classes among".

#64 - 05 juillet 2018 14:13 - Anonyme

- Fichier 4628a2a9c1bf4ac0b2f10d2e94c0dfc4.mp4 ajouté

#65 - 05 juillet 2018 14:18 - Mikaël Ates

Ok pour cette vidéo pour ma part. C'est possible d'avoir la même en responsive (e.g. 360px de large) ?

#66 - 05 juillet 2018 14:34 - Anonyme

- Fichier 127.0.0.1_9876_accounts_activate_aze_(2).png ajouté

Mikaël Ates a écrit :

Ok pour cette vidéo pour ma part. C'est possible d'avoir la même en responsive (e.g. 360px de large) ?

#67 - 05 juillet 2018 14:46 - Anonyme

- Fichier 0001-create-assisted-password-input-widgets-24438.patch ajouté

- Statut changé de En cours à Solution proposée

Voilà le patch finalisé après les différents aller-retours.

#68 - 05 juillet 2018 14:59 - Frédéric Péters

In order to create a secure password, please comply at least these rules :

- Your password must contain at least 8 characters
- Your password must contain at least 3 classes among...

Je persiste dans mon commentaire sur ces messages, ne pas verbaliser les critères, comme dans les maquettes GLC, ex:

In order to create a secure password, please use at least:

- 8 characters
- 3 different kinds of characters (between lowercase etc.)

Aussi sur le style, c'est jumpy à cause du changement de largeur puce→croix, puis croix→coche; et l'apparition du paragraphe "passwords do not match". (bien sûr tout ça peut être corrigé dans publik-base-theme).

#69 - 05 juillet 2018 16:16 - Anonyme

- Fichier 0001-create-assisted-password-input-widgets-24438.patch ajouté

- Fichier b8e7ef0a3bac4b41bce29f5cc480e82b.mp4 ajouté

Frédéric Péters a écrit :

In order to create a secure password, please comply at least these rules :

- Your password must contain at least 8 characters
- Your password must contain at least 3 classes among...

Je persiste dans mon commentaire sur ces messages, ne pas verbaliser les critères, comme dans les maquettes GLC, ex:

In order to create a secure password, please use at least:

- 8 characters
- 3 different kinds of characters (between lowercase etc.)

C'est OK (les traductions seront dans un second temps)

Aussi sur le style, c'est jumpy à cause du changement de largeur puce→croix, puis croix→coche; et l'apparition du paragraphe "passwords do not match". (bien sûr tout ça peut être corrigé dans publik-base-theme).

J'évite le saut du paragraphe "passwords match". J'ai ajouté le message par défaut : "les deux mots de passes sont identiques", pour suivre la même tournure de phrase que tu proposes.

Pour éviter le saut de puce->icone, j'ai dû remplacer la puce "native" par une icône Font Awesome avec la même taille et la même marge, j'utilises le même système que Font Awesome pour éviter l'effet Jumpy (<https://github.com/FortAwesome/Font-Awesome/blob/v4.7.0/less/fix-width.less>)

Je propose cette icône par défaut <https://fontawesome.com/v4.7.0/icon/hand-o-right>.

Oui c'est une base pour Authentic, et on pourra effectivement re-travailler avec différents clients les tournures et les couleurs.

#70 - 05 juillet 2018 16:31 - Frédéric Péters

(bof pour la main, tu peux pas juste taper un `display: inline-block; width: 1rem; text-align: center ?`)

#71 - 05 juillet 2018 16:45 - Anonyme

- Fichier `0001-create-assisted-password-input-widgets-24438.patch` ajouté

- Fichier `f2b2a688c96b4a1386a1821967593867.mp4` ajouté

Finalement je cache ici le texte "both passwords must match", je ne crois pas que c'était ce qui était attendu.

Je remplace le right-hand par un petit cercle noir.

La CSS que tu proposes est similaires à ce que j'ai utilisé pour éviter l'effet jumpy : l'élément de l'icone ".a2-password-icon" à une grande largeur, un contenu centré mais reste vide, c'est `:before` qui porte l'icône. Ça permet d'avoir une taille fixe et d'éviter tout mouvement du texte placé après.

#72 - 06 juillet 2018 10:31 - Anonyme (Notes privées)

Je suis en train de faire les derniers ajustements et les traductions.

#73 - 06 juillet 2018 12:14 - Benjamin Dauvergne

Au niveau des appels AJAX ce n'est pas du tout ce qu'on avait dit, c'est le web-service qui renvoie les labels pas des classes d'erreur, ce que j'ai posté sur [#24833](#) ça retourne ça:

```
{
  "result": 1,
  "ok": false,
  "checks": [
    {"label": "at least 1 digit", "result": false}
  ]
}
```

#74 - 06 juillet 2018 17:52 - Anonyme

- Fichier `0001-create-assisted-password-input-widgets-24438.patch` ajouté

Benjamin Dauvergne a écrit :

Au niveau des appels AJAX ce n'est pas du tout ce qu'on avait dit, c'est le web-service qui renvoie les labels pas des classes d'erreur, ce que j'ai posté sur [#24833](#) ça retourne ça:

C'est modifié pour coller à cette nouvelle API. J'ai enlevé mon code devenu inutile.

J'ai modifié les messages pour coller plus à la discussion et avoir un wording du type : "8 characters" à la place de "at least 8 characters". (noter qu'il y a une phrase d'intro qui finit par "at least :". J'ai mis à jour les traductions aussi. Si c'est acké, alors n'hésitez pas à pousser pour moi, bien sûr à la suite des 2 patchs de [#24833](#).

#75 - 06 juillet 2018 17:59 - Anonyme

- Fichier `0001-create-assisted-password-input-widgets-24438.patch` ajouté

- Fichier `0002-update-translation-for-password-validation-24438.patch` ajouté

Même chose mais divisé en 2 commit mettant les traductions à part.

#76 - 06 juillet 2018 18:25 - Anonyme

- Fichier `0001-create-assisted-password-input-widgets-24438.patch` ajouté

- Fichier `0002-update-translation-for-password-validation-24438.patch` ajouté

Et enfin après avoir fixé les tests (pour le changement de texte de messages de validation) cf.

<https://jenkins2.entrouvert.org/job/authentic2-wip/job/wip%252Fnewpassword/5/console>

(Je vous laisse toujours le soin de pousser en mon absence si besoin, tout en faisant passer les patches de [#24833](#) avant svp)

#77 - 08 juillet 2018 12:10 - Frédéric Péters

Le numéro de ticket mentionné dans le message de commit ne correspond pas.

En local je me trouve avec password.js chargé deux fois (parce que champ mot de passe + champ de confirmation ?); ça donne deux fois le même handler posé sur keyup, donc deux fois les appels serveurs de validation.

Je ne pense pas que seulement réagir à keyup soit suffisant (genre aussi réagir à "paste").

À remplir puis vider on se trouve avec les critères qui ne sont pas réinitialisés (parce que `if (!$this.val()) return;`).

(En l'état pas ok pour moi)

#78 - 16 juillet 2018 19:51 - Anonyme

- Fichier `4c939103631a4f71b1dd506c2be79218.mp4` ajouté

- Fichier `0002-update-translation-for-password-validation-24438.patch` ajouté

- Fichier `0001-create-assisted-password-input-widgets-24439.patch` ajouté

Frédéric Péters a écrit :

Le numéro de ticket mentionné dans le message de commit ne correspond pas.

Corrigé, désolé

En local je me trouve avec password.js chargé deux fois (parce que champ mot de passe + champ de confirmation ?); ça donne deux fois le même handler posé sur keyup, donc deux fois les appels serveurs de validation.

C'est pas normal, Django limite à un seul JS ou CSS par lui-même automatiquement (media liés au widget Input), et je n'ai de mon côté qu'un seul script Js et un seul link Css. Et je n'ai pas d'événement déclenchés en double. Peut-être un thème qui inclut par erreur deux fois des assets ?

Je ne pense pas que seulement réagir à keyup soit suffisant (genre aussi réagir à "paste").

J'ai ajouté ça.

À remplir puis vider on se trouve avec les critères qui ne sont pas réinitialisés (parce que `if (!$this.val()) return;`).

J'ai corrigé ça.

(En l'état pas ok pour moi)

Un vidéo en plus pour montrer le tout ! Et hop.

#79 - 16 juillet 2018 20:25 - Frédéric Péters

En local je me trouve avec password.js chargé deux fois (...)

C'est pas normal (...) Peut-être un thème qui inclut par erreur deux fois des assets ?

C'est un thème publik tout à fait normal. (clapotis-les-canards, qui ne touche pas au theme.html).

Ce qui s'y passe, c'est :

```
{% block extra_scripts %}
{{ block.super }}
{{ form.media.js }}
{% endblock %}
```

alors que publik-base-theme pose déjà `{{ form.media }}` de manière générale.

#80 - 17 juillet 2018 09:19 - Anonyme

- Fichier 0002-update-translation-for-password-validation-24438.patch ajouté

- Fichier 0001-create-assisted-password-input-widgets-24439.patch ajouté

Frédéric Péters a écrit :

En local je me trouve avec password.js chargé deux fois (...)

C'est pas normal (...) Peut-être un thème qui inclut par erreur deux fois des assets ?

C'est un thème public tout à fait normal. (clapotis-les-canards, qui ne touche pas au theme.html).

Bien vu, j'ai retiré form.media du template registration, et pour éviter le doublon et en même temps inclure ces fichiers dans un authentic standalone, j'ai rajouté form.media.js et form.media.css dans authentic2/templates/authentic2/base.html. J'ai vérifié que password.js et css ne sont inclus qu'une seule fois en situation d'un thème et d'un serveur a2 standalone.

#81 - 17 juillet 2018 16:12 - Anonyme

- Fichier 0002-update-translation-for-password-validation-24439.patch ajouté

- Fichier 0001-create-assisted-password-input-in-registration-24439.patch ajouté

- Rebase de master pour éviter l'échec dans le test test_user_cannot_change_password
- Avec les numéro de commit dans le message qui est corrigé...

#82 - 18 juillet 2018 10:10 - Anonyme

- Lié à Development #25045: Nouvelle interface de mot de passe pour le changement d'un mot de passe en FO et BO ajouté

#83 - 18 juillet 2018 16:16 - Anonyme

- Fichier 0001-create-assisted-password-input-in-registration-24439.patch ajouté

- Fichier 0002-update-translation-for-password-validation-24439.patch ajouté

Reprise des deux patches en enlevant ce qui a été poussé ce matin pour [#24833](#).

#84 - 18 juillet 2018 18:22 - Benjamin Dauvergne

Merde j'ai pas fait mes remarques, donc:

- renommer A2_PASSWORD_WIDGET_SHOW_ALL_BUTTON en A2_PASSWORD_POLICY_SHOW_PASSWORD_BUTTON pour pas multiplier les namespaces de setting
- j'ai vraiment du mal avec assisted_password.html, je ne vois pas ce qu'on ne peut pas faire via le help_text d'un PasswordField customisé (on peut mettre du HTML avec mark_safe()).

#85 - 18 juillet 2018 18:34 - Benjamin Dauvergne

L'utilisation de app_settings.A2_PASSWORD_WIDGET_SHOW_ALL_BUTTON dans un contexte global ne va pas fonctionner en mode multitenant (les settings varie entre tenants).

#86 - 19 juillet 2018 09:19 - Benjamin Dauvergne

J'ai poussé une branche pour montrer la forme que je souhaite (pas de template qui fait le boulot 2 fois) :

<http://git.entrouvert.org/authentic.git/log/?h=wip/24439-Nouvelle-interface-de-mot-de-passe>

#87 - 19 juillet 2018 11:29 - Anonyme

Merci pour la branche, même si j'ai eu un peu de mal à voir les changement de logique à cause de déplacements de code d'un fichier à l'autre à voir certaines différences (les changements de noms de variables ou classes n'ont pas aidé non plus)

Avec le tag de ce matin, c'est trop tard pour la prochaine mise à jour...

Mais j'ai regardé wip et wip2:

- la question du template assisted_password.html qui ferait le boulot deux fois :
- la logique c'est que plus le template contient de la présentation, plus on sera souples dans la CSS et des templates enfants pour répondre facilement aux évolutions spécifiques; c'est aussi l'idée d'un render() par template dans Django 1.11, ou celle de séparation des responsabilités.
- par exemple, si tu veux un jour changer la formulation de help_text, tu seras obligé de faire du python. Toutes les questions de changement de

"textes" et de "libellés" qui viennent parfois de clients comme juste hier le CD06.

- OK pour la création de Field pour encapsuler plus les Input, même si ce niveau est utile que pour mettre de l'HTML dans du Python comme tu le proposes à helptext.
- OK renommage de A2_PASSWORD_POLICY_SHOW_PASSWORD_BUTTON dans les settings ici **par défaut**, si j'ai bien compris. Même si les settings varient dans les tenants, ils pourront surcharger cette variable pour passer la valeur par défaut à True.
 - et après toutes les discussions sur cette fonction, j'ai presque envie de l'enlever pour arrêter d'en parler et nous alléger la tâche.
- OK pour le renommage des classes css, attributs HTML.
- other_widget_name : on ne peut pas utiliser ça pour éviter la casse du système en cas de Form Django avec préfixe (déjà discuté dans ce ticket).
- getValidation: la sélection de policyContainer ne devrait pas changer à mon avis, le système de sélection par "name" assure qu'un message=un input, donc préférablement ne pas supprimer var policyContainer = \$('#a2-password-policy-helper-' + inputName);
- ok pour ajouter les 2 commits de nettoyage import json, et csrf dans la série de patches du ticket.
- a2-password-policy-container : la présentation des critères au chargement avec l'utilisation de help_text n'est pas identique, ça fait sauter toute la présentation après la première frappe. Les discussions précédentes allaient vers une stabilité visuelle de la présentation, rien qui saute.

#88 - 19 juillet 2018 13:17 - Benjamin Dauvergne

Elias Showk a écrit :

- la logique c'est que plus le template contient de la présentation, plus on sera souples dans la CSS et des templates enfants pour répondre facilement aux évolutions spécifiques; c'est aussi l'idée d'un render() par template dans Django 1.11, ou celle de séparation des responsabilités.

La présentation m'importe peu, uniquement la clarté du code m'importe à ce stade.

- par exemple, si tu veux un jour changer la formulation de help_text, tu seras obligé de faire du python. Toutes les questions de changement de "textes" et de "libellés" qui viennent parfois de clients comme juste hier le CD06.

Je ne pense pas que changer ces formulations ait une importance, il faut en avoir des correctes.

- OK pour la création de Field pour encapsuler plus les Input, même si ce niveau est utile que pour mettre de l'HTML dans du Python comme tu le proposes à helptext.

Franchement cette fonctionnalité est anecdotique,

- OK renommage de A2_PASSWORD_POLICY_SHOW_PASSWORD_BUTTON dans les settings ici **par défaut**, si j'ai bien compris. Même si les settings varient dans les tenants, ils pourront surcharger cette variable pour passer la valeur par défaut à True.
 - et après toutes les discussions sur cette fonction, j'ai presque envie de l'enlever pour arrêter d'en parler et nous alléger la tâche.

Je suis ok pour virer tout cette partie du code, de toute façon le rendu est pas terrible, on l'activerai pas. Comme je le disais au tout début ça prend plus de temps que tu ne le pensais de faire correctement chacune de ces fonctions.

- OK pour le renommage des classes css, attributs HTML.

L'idée c'est d'avoir un espace de nom pour chaque chose et voir ce qui vient de a2 et ce qui est plus général, d'ailleurs au lieu d'utiliser des data-... je serais pour passer plutôt par des classes, c'est plus l'usage, data- c'est uniquement si ça doit porter une valeur.

- other_widget_name : on ne peut pas utiliser ça pour éviter la casse du système en cas de Form Django avec préfixe (déjà discuté dans ce ticket).

Oui mais ton fonctionnement est pire, il recherche par name qui ne marche pas avec les préfixes et fait des suppositions sur l'agencement des inputs dans le form, je préfère encore cela qui est explicite. La structure de Django ne permet de toute façon pas d'interaction entre widget de façon portable.

- getValidation: la sélection de policyContainer ne devrait pas changer à mon avis, le système de sélection par "name" assure qu'un message=un input, donc préférablement ne pas supprimer var policyContainer = \$('#a2-password-policy-helper-' + inputName);

On est toujours certains que le help-text est encapsulé dans un container avec l'input, donc non ça marche aussi.

- ok pour ajouter les 2 commits de nettoyage import json, et csrf dans la série de patches du ticket.
- a2-password-policy-container : la présentation des critères au chargement avec l'utilisation de help_text n'est pas identique, ça fait sauter toute la présentation après la première frappe. Les discussions précédentes allaient vers une stabilité visuelle de la présentation, rien qui saute.

Je n'ai pas suivi cette partie; dans ce cas on garde le visuel initial inline sans la liste, de toute façon ça prend trop de place.

```
blabla : <span>crit1, crit2</span>
```

devient ensuite

```
blabla : <span><span ok>crit1</span>, <span nok>crit2</span></span>
```

#89 - 19 juillet 2018 17:24 - Anonyme

- Assigné à changé de Anonyme à Benjamin Dauvergne

suite à discussion sur jabber, je redonne la main pour finaliser après ce qui a été proposé dans la nouvelle branche de benjamin

#90 - 20 juillet 2018 00:44 - Benjamin Dauvergne

Voilà c'est sûr ma branche, ça peut être relu sauf les 3 derniers patches qui est le début du taf pour mettre ce widget partout.

Pour l'affichage du dernier caractère je suis revenu à un positionnement absolu, en forçant le conteneur du widget en position: relative dans le code JS, je ne vois pas ce que ça pourrait avoir comme conséquences mais bon.

<http://git.entrouvert.org/authentic.git/log/?h=wip/24439-Nouvelle-interface-de-mot-de-passe>

#91 - 20 juillet 2018 08:04 - Frédéric Péters

Il y a dans "add new widget and fields for passwords (#24439)" du bruit de .po.

#92 - 20 juillet 2018 08:38 - Frédéric Péters

- Fichier Screenshot_2018-07-20 Publik local de Fred - Enregistrement.png ajouté

Pour l'affichage du dernier caractère je suis revenu à un positionnement absolu (...)

Absolu mais il semble manquer un right: 0; là pour moi le caractère apparait à gauche lors de la saisie, ce qui rend l'affaire curieuses.

Visuellement pas fan de la boulette affichée avant la saisie derrière les critères; mais le balisage me semble ok pour arriver à ce qui se trouve dans les maquettes CUT, par quelque chose genre :

```
.a2-password-policy-container {
  display: block;
  column-count: 2;
}

.a2-password-policy-rule {
  display: block;
}

// virer les icônes des ::after
// opacity: 0.7; sur .a2-password-ok
```

#93 - 20 juillet 2018 08:53 - Frédéric Péters

Absolu mais il semble manquer un right: 0; là pour moi le caractère apparait à gauche lors de la saisie, ce qui rend l'affaire curieuses.

Ainsi que,

```
// on crée un div placé dans le padding-right de l'input
```

mais plus loin dans le même fichier

```
$input.css({'padding-left': '1.25rem'});
```

à noter aussi dans ce fichier password.js un mélange tabulations/espaces pour les indentations, ce commentaire en français, un console.log() de debug.

#94 - 20 juillet 2018 10:00 - Benjamin Dauvergne

Frédéric Péters a écrit :

Pour l'affichage du dernier caractère je suis revenu à un positionnement absolu (...)

Absolu mais il semble manquer un right: 0; là pour moi le caractère apparaît à gauche lors de la saisie, ce qui rend l'affaire curieuse.

Je me suis rendu compte que ce n'était vraiment pas ergonomique d'afficher le caractère tout à droite, soit on l'affiche au niveau du curseur de saisie et c'est visible, soit tout à gauche, mais tout à droite, selon la longueur du champ ce sera invisible pour pas mal d'utilisateurs qui seront concentrés sur le curseur de saisie.

Visuellement pas fan de la boulette affichée avant la saisie derrière les critères; mais le balisage me semble ok pour arriver à ce qui se trouve dans les maquettes CUT, par quelque chose genre :

[...]

En fait j'ai remis à droite les indicateurs visuels parce que pour le champ de confirmation avoir le picto avant le texte sachant qu'il n'y a pas de texte introductif rendait mal (non alignement des help_text entre les deux champs). Effectivement les boulettes ne servent plus à rien si on les met à droite et je vais les virer en mettant opacity: 0 pour garder l'espacement entre les critères.

Pour le rendu CUT je suis pour proposer celui là d'abord en disant que c'est le rendu qu'on souhaite avoir chez tous nos clients.

#95 - 20 juillet 2018 10:09 - Anonyme

- un oubli dans password.js ligne 79 : \$this undefined, c'est \$input, l'évènement "paste" est cassé.
- le placement à gauche du caractère, pourquoi pas, c'est sympa sur le thème SAU, mais je ne suis pas sûr que le "position: absolute;" fonctionne partout.
- pour le style en ligne des critères, on s'éloigne des maquettes du CUT, et je joins une capture pour illustrer ma difficulté à savoir à quoi est associé la croix rouge, comme le texte "8 caractères" reste noir. L'affichage en 2 colonnes avec un espacement entre les critères permettait de lever le doute.
- les tests ne passent pas, j'ai l'impression que le refactoring de forms en est la cause <https://jenkins2.entrouvert.org/job/authentic2-wip/job/wip%252F24439-Nouvelle-interface-de-mot-de-passe/4/>
- finalement le code du bouton "oeil" a été enlevé, il devra être re-développé pour le CUT ?
- note : concernant les 3 derniers commits, j'avais aussi créé <https://dev.entrouvert.org/issues/25045> si ça peut servir

#96 - 20 juillet 2018 10:14 - Benjamin Dauvergne

Branche mise à jour avec pour password.js correction à l'indentation pour du 4 espaces partout, console.log() retiré et commentaires en anglais sur le placement du span de dernier caractère.

#97 - 20 juillet 2018 10:22 - Mikaël Ates

- finalement le code du bouton "oeil" a été enlevé, il devra être re-développé pour le CUT ?

Non, voir <https://dev.entrouvert.org/issues/22486#note-10>. Le mot de passe tout en clair n'était plus dans le périmètre.

#98 - 20 juillet 2018 10:30 - Frédéric Péters

Je me suis rendu compte que ce n'était vraiment pas ergonomique d'afficher le caractère tout à droite, soit on l'affiche au niveau du curseur de saisie et c'est visible, soit tout à gauche, mais tout à droite, selon la longueur du champ ce sera invisible pour pas mal d'utilisateurs qui seront concentrés sur le curseur de saisie.

Pour ma part vraiment pas fan de cet affichage, ça m'irait mieux que la fonctionnalité ne soit pas activée par défaut et expérimentée côté CUT (mais il ne me semble pas y avoir d'option (mais si on ne veut pas d'option peut bien sûr désactiver au niveau thème)).

```
+     $span.css({
+ ...
+         'font-weight': 'bold',
+ ...
+     $input.parent().css({'position': 'relative'});
+     $input.css({'padding-left': '1.25rem'});
```

Je préfère avoir les CSS dans le fichier CSS, les poser ainsi directement sur l'élément leur donne une priorité trop importante (qui oblige à utiliser le moche !important dans la CSS si jamais on veut passer outre). Je parle uniquement des styles "fixes".

#99 - 20 juillet 2018 10:43 - Frédéric Péters

Sur un ctrl-v, ça affiche une mauvaise indication comme quoi un v est tapé; dans l'objet event : ctrlKey: true.

#100 - 20 juillet 2018 11:08 - Benjamin Dauvergne

Elias Showk a écrit :

- les tests ne passent pas, j'ai l'impression que le refactoring de forms en est la cause
<https://jenkins2.entrouvert.org/job/authentic2-wip/job/wip%252F24439-Nouvelle-interface-de-mot-de-passe/4/>

Il manquait un import, c'est corrigé, j'ai viré un de tes tests qui testait juste qu'on avait certaines classes présentes, c'est pas super utile comme test unitaire.

#101 - 20 juillet 2018 11:14 - Benjamin Dauvergne

Frédéric Péters a écrit :

Je me suis rendu compte que ce n'était vraiment pas ergonomique d'afficher le caractère tout à droite, soit on l'affiche au niveau du curseur de saisie et c'est visible, soit tout à gauche, mais tout à droite, selon la longueur du champ ce sera invisible pour pas mal d'utilisateurs qui seront concentrés sur le curseur de saisie.

Pour ma part vraiment pas fan de cet affichage, ça m'irait mieux que la fonctionnalité ne soit pas activée par défaut et expérimentée côté CUT (mais il ne me semble pas y avoir d'option (mais si on ne veut pas d'option peux bien sûr désactiver au niveau thème)).

C'est à dire ? Tu voulais bien en standard si affiché sur la droite ou bien t'es pas fan de la fonctionnalité en elle même ?

Je préfère avoir les CSS dans le fichier CSS, les poser ainsi directement sur l'élément leur donne une priorité trop importante (qui oblige à utiliser le moche !important dans la CSS si jamais on veut passer outre). Je parle uniquement des styles "fixes".

J'ai réorganisé:

- position: absolute -> posé par JS puisque c'est lui qui positionne de toute façon
- .last-char + input[passwords] { padding-left: 1.25rem } -> remis dans la CSS on pourra varier cet espace
- .last-char { font-weight: bold } mis dans la CSS

J'ai essayé de ne garder dans le JS que ce qui est nécessaire à un bon positionnement absolute.

#102 - 20 juillet 2018 11:14 - Benjamin Dauvergne

Frédéric Péters a écrit :

Sur un ctrl-v, ça affiche une mauvaise indication comme quoi un v est tapé; dans l'objet event : ctrlKey: true.

Uniquement sous FF, mais c'est corrigé.

#103 - 20 juillet 2018 11:19 - Benjamin Dauvergne

Voilà la fonctionnalité d'affichage du dernier caractère est par défaut désactivée, à activer via A2_PASSWORD_POLICY_SHOW_LAST_CHAR=True.

D'autres objections ?

#104 - 20 juillet 2018 11:33 - Frédéric Péters

C'est à dire ? Tu voulais bien en standard si affiché sur la droite ou bien t'es pas fan de la fonctionnalité en elle même ?

En elle-même je l'ai déjà dit, mais dans sa réalisation, je préférerais à droite.

D'autres objections ?

Ça reste plutôt dans les commentaires; que j'avais noté à Elias, le côté qui saute un peu parce que différentes tailles de caractères, ça peut s'arranger en ajoutant dans .a2-password-policy-rule::after ces règles :

```
display: inline-block;
width: 3ex;
text-align: center;
```

(et le padding: 0.5ex; ajouté aux -ok et -nok peut être supprimé).

Reste un côté qui fait sauter les choses ce sont les virgules avant d'entrer dans le champ, a minima, et aussi pour permettre le style de la maquette CUT, il y aurait moyen de les mettre en ?

#105 - 20 juillet 2018 12:30 - Benjamin Dauvergne

Frédéric Péters a écrit :

C'est à dire ? Tu voulais bien en standard si affiché sur la droite ou bien t'es pas fan de la fonctionnalité en elle même ?

En elle-même je l'ai déjà dit, mais dans sa réalisation, je préférerais à droite.

D'autres objections ?

Ça reste plutôt dans les commentaires; que j'avais noté à Elias, le côté qui saute un peu parce que différentes tailles de caractères, ça peut s'arranger en ajoutant dans .a2-password-policy-rule::after ces règles :

[...]

(et le padding: 0.5ex; ajouté aux -ok et -nok peut être supprimé).

Intégré.

Reste un côté qui fait sauter les choses ce sont les virgules avant d'entrer dans le champ, a minima, et aussi pour permettre le style de la maquette CUT, il y aurait moyen de les mettre en ?

J'ai viré les virgules et je laisse l'espace du ::after avec opacity: 0 gérer l'espacement.

#106 - 20 juillet 2018 15:48 - Benjamin Dauvergne

- Fichier 0002-api-do-not-do-CSRF-check-on-validate-password-API-24.patch ajouté

- Fichier 0003-api-allow-empty-password-in-validate-password-24439.patch ajouté

- Fichier 0001-templates-output-form.media-in-base-template-24439.patch ajouté

- Fichier 0004-create-authentic2.forms-package-24439.patch ajouté

- Fichier 0007-add-new-widget-and-fields-for-passwords-24439.patch ajouté

- Fichier 0006-move-all-password-related-functions-in-authentic2.pa.patch ajouté

- Fichier 0009-update-french-translation-24439.patch ajouté

- Fichier 0005-move-authentic2.widgets-to-authentic2.forms.widgets-.patch ajouté

- Fichier 0008-use-new-password-fields-in-registration-form-fixes-2.patch ajouté

Voilà ce qui doit être poussé, j'ai mis mes autres patches d'intégrations des nouveaux champs sur une autre branche.

#107 - 20 juillet 2018 16:21 - Anonyme

Benjamin Dauvergne a écrit :

Voilà ce qui doit être poussé, j'ai mis mes autres patches d'intégrations des nouveaux champs sur une autre branche.

- Sur la page password/change : les conditions sont collées les unes aux autres, peut-être une solution

```
<span class="a2-password-policy-container"><span class="a2-password-policy-rule">8 caractères</span><span class="a2-password-policy-rule">1 minuscule</span><span class="a2-password-policy-rule">1 chiffre</span><span class="a2-password-policy-rule">1 majuscule</span></span>
```

- dans password.css, l'unité padding: 1rex; est pas valide (sous chromium ici, peut-être partout). Les écarts entre les conditions sont un peu trop élevés au chargement, je mettrai sur ::before width: 2ex à la place de 3ex.
- dans password.js, encore du code commenté et des console.log oubliés.
 - il manque aussi un "use strict"; au début du code pour éviter certains bugs. Mais alors il faudra surement mettre les fonction de plus haut niveau dans window, par exemple window.a2_password_check_equality
- plus une question qu'un problème ici : ValidatePasswordAPI est publique avec CsrfExemptSessionAuthentication, mais est-ce qu'on n'a pas de risques d'abus sans contrôle de Referer ou autre, pour vérifier que ça vient de chez nous ?

- je ne suis pas l'auteur de d1350bbab85da968b4cdee00c7fb667e4ad53260, ça doit être le résultat d'un "amend" quand tu as repris le patch.

#108 - 20 juillet 2018 17:03 - Benjamin Dauvergne

Elias Showk a écrit :

Benjamin Dauvergne a écrit :

Voilà ce qui doit être poussé, j'ai mis mes autres patches d'intégrations des nouveaux champs sur une autre branche.

- Sur la page password/change : les conditions sont collées les unes aux autres, peut-être une solution

[...]

Je ne comprends pas la solution.

- dans password.css, l'unité padding: 1rex; est pas valide (sous chromium ici, peut-être partout). Les écarts entre les conditions sont un peu trop élevés au chargement, je mettrai sur ::before width: 2ex à la place de 3ex.

Mis 1.5ex a vu de nez.

- dans password.js, encore du code commenté et des console.log oubliés.

Vire.

- il manque aussi un "use strict"; au début du code pour éviter certains bugs. Mais alors il faudra surement mettre les fonction de plus haut niveau dans window, par exemple window.a2_password_check_equality

Bof. Je fais pas assez de JS pour trouver ça interessant.

- plus une question qu'un problème ici : ValidatePasswordAPI est publique avec CsrfExemptSessionAuthentication, mais est-ce qu'on n'a pas de risques d'abus sans contrôle de Referer ou autre, pour vérifier que ça vient de chez nous ?

Quels abus ? Des gens qui abusent de verifier le format de leur mot de passe ? En plus sans CORS c est pas evident a faire.

- je ne suis pas l'auteur de d1350bbab85da968b4cdee00c7fb667e4ad53260, ça doit être le résultat d'un "amend" quand tu as repris le patch.

Je corrige.

Modif pousses sur la branche.

#109 - 20 juillet 2018 17:06 - Anonyme

- Statut changé de Solution proposée à Solution validée

Ack sur le contenu mis à jour wip/24439-Nouvelle-interface-de-mot-de-passe

#110 - 20 juillet 2018 17:10 - Benjamin Dauvergne

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

- % réalisé changé de 0 à 100

Appliqué par commit [authentic2|c46822af588a4fed0fefce8953e808f579b49565](#).

#111 - 23 juillet 2018 09:33 - Mikaël Ates

- Lié à Bug #25408: Les pictos et les messages de la nouvelle interface de mot de passe ne s'affichent pas correctement sur le thème de démo ajouté

#112 - 13 décembre 2018 22:34 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

Elements_interfaces_CUT.pdf	1,07 Mo	11 juin 2018	Anonyme
0001-create-password-validation-in-registration-completio.patch	11,4 ko	14 juin 2018	Anonyme

0001-add-password-validation-UI-in-registration-view-2443.patch	11,2 ko	15 juin 2018	Anonyme
Capture d'écran 2018-06-15 à 11.38.35-fullpage.png	78,6 ko	15 juin 2018	Anonyme
Capture d'écran 2018-06-15 à 11.38.29-fullpage.png	80,1 ko	15 juin 2018	Anonyme
Capture d'écran 2018-06-15 à 11.40.36-fullpage.png	80,7 ko	15 juin 2018	Anonyme
Capture d'écran 2018-06-15 à 11.36.25-fullpage.png	78,1 ko	15 juin 2018	Anonyme
0001-add-password-validation-interfacein-registration-for.patch	16,8 ko	21 juin 2018	Anonyme
0001-create-AssistedPassword-AssistedPasswordFormMixin-an.patch	17,4 ko	21 juin 2018	Anonyme
0001-create-AssistedPassword-AssistedPasswordFormMixin-an.patch	17,8 ko	21 juin 2018	Anonyme
1bb8edd1cc8e4dfeb2443d8bb3f75459.mp4	181 ko	22 juin 2018	Anonyme
0001-create-AssistedPassword-AssistedPasswordFormMixin-an.patch	20 ko	22 juin 2018	Anonyme
0001-create-AssistedPassword-AssistedPasswordFormMixin-an.patch	20,1 ko	22 juin 2018	Anonyme
0001-create-AssistedPassword-AssistedPasswordFormMixin-24.patch	23,9 ko	22 juin 2018	Anonyme
0001-create-assisted-password-input-widgets-24438.patch	23,2 ko	29 juin 2018	Anonyme
connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_ (2).png	82,5 ko	29 juin 2018	Anonyme
connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_ (Galaxy295).png	92,5 ko	29 juin 2018	Anonyme
connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_ (Galaxy295).png	82,5 ko	29 juin 2018	Anonyme
connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_ (Galaxy295).png	82,5 ko	29 juin 2018	Anonyme
connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_ (Galaxy295).png	92,5 ko	29 juin 2018	Anonyme
connexion-authentic.dev-eshowk.ddns.entrouvert.org_accounts_activate_aze_ (Galaxy295).png	82,5 ko	29 juin 2018	Anonyme
aab6f794279c44b79542c0487468f689-1530699221194.mp4	373 ko	04 juillet 2018	Anonyme
e16a3bff245840d184a31bff430a24b0.mp4	234 ko	04 juillet 2018	Anonyme
0a94b94568d44f60b63fb5cbb99d9023-1530787531991.mp4	197 ko	05 juillet 2018	Anonyme
4628a2a9c1bf4ac0b2f10d2e94c0dfc4.mp4	307 ko	05 juillet 2018	Anonyme
127.0.0.1_9876_accounts_activate_aze_ (2).png	45,9 ko	05 juillet 2018	Anonyme
0001-create-assisted-password-input-widgets-24438.patch	31,6 ko	05 juillet 2018	Anonyme
0001-create-assisted-password-input-widgets-24438.patch	32 ko	05 juillet 2018	Anonyme
b8e7ef0a3bac4b41bce29f5cc480e82b.mp4	168 ko	05 juillet 2018	Anonyme
0001-create-assisted-password-input-widgets-24438.patch	31,8 ko	05 juillet 2018	Anonyme
f2b2a688c96b4a1386a1821967593867.mp4	404 ko	05 juillet 2018	Anonyme
0001-create-assisted-password-input-widgets-24438.patch	38,3 ko	06 juillet 2018	Anonyme
0002-update-translation-for-password-validation-24438.patch	12,9 ko	06 juillet 2018	Anonyme
0001-create-assisted-password-input-widgets-24438.patch	24,9 ko	06 juillet 2018	Anonyme
0002-update-translation-for-password-validation-24438.patch	12,9 ko	06 juillet 2018	Anonyme
0001-create-assisted-password-input-widgets-24438.patch	29,1 ko	06 juillet 2018	Anonyme
4c939103631a4f71b1dd506c2be79218.mp4	430 ko	16 juillet 2018	Anonyme
0002-update-translation-for-password-validation-24438.patch	12,9 ko	16 juillet 2018	Anonyme
0001-create-assisted-password-input-widgets-24439.patch	30,5 ko	16 juillet 2018	Anonyme
0002-update-translation-for-password-validation-24438.patch	12,9 ko	17 juillet 2018	Anonyme
0001-create-assisted-password-input-widgets-24439.patch	30,5 ko	17 juillet 2018	Anonyme
0002-update-translation-for-password-validation-24439.patch	12,9 ko	17 juillet 2018	Anonyme
0001-create-assisted-password-input-in-registration-24439.patch	30,3 ko	17 juillet 2018	Anonyme
0002-update-translation-for-password-validation-24439.patch	8,49 ko	18 juillet 2018	Anonyme
0001-create-assisted-password-input-in-registration-24439.patch	24,9 ko	18 juillet 2018	Anonyme
Screenshot_2018-07-20 Publik local de Fred - Enregistrement.png	4,64 ko	20 juillet 2018	Frédéric Péters
0002-api-do-not-do-CSRF-check-on-validate-password-API-24.patch	1,5 ko	20 juillet 2018	Benjamin Dauvergne
0003-api-allow-empty-password-in-validate-password-24439.patch	786 octets	20 juillet 2018	Benjamin Dauvergne
0001-templates-output-form.media-in-base-template-24439.patch	2,36 ko	20 juillet 2018	Benjamin Dauvergne
0004-create-authentic2.forms-package-24439.patch	2,63 ko	20 juillet 2018	Benjamin Dauvergne
0007-add-new-widget-and-fields-for-passwords-24439.patch	14,6 ko	20 juillet 2018	Benjamin Dauvergne
0006-move-all-password-related-functions-in-authentic2.pa.patch	3,6 ko	20 juillet 2018	Benjamin Dauvergne
0009-update-french-translation-24439.patch	15,4 ko	20 juillet 2018	Benjamin Dauvergne
0005-move-authentic2.widgets-to-authentic2.forms.widgets-.patch	13,9 ko	20 juillet 2018	Benjamin Dauvergne
0008-use-new-password-fields-in-registration-form-fixes-2.patch	2,78 ko	20 juillet 2018	Benjamin Dauvergne