

Personnalisation du profil

Comment faire ?

- Dans `dev/configuration/config.js` de Sakai-ux, il y a une partie `Profile` représentant les différentes sections et les champs associés. Faites les modifications plutôt dans `dev/configuration/config_custom.js` en redéfinissant tout ou une partie des sections.

Cf. partie `Profile` sur [University of Cambridge Institutional Deployment Details](#)

- On peut utiliser le mot-clé **delete** pour supprimer un élément. Exemple pour le champ *Academic Interests* de *About me* :

```
delete config.Profile.configuration.defaultConfig.aboutme.elements.academicinterests;
```

- Si vous personnalisez les champs du profil de la section *Basic information (Informations de base)*, il y a des modifications supplémentaires à effectuer au niveau de Nakamura : Dans la console web Sling, `Configuration org.sakaiproject.nakamura.user.BasicUserInfoServiceImpl.name` (ou en [éditant le fichier](#) `sling/config/org/sakaiproject/nakamura/user/BasicUserInfoServiceImpl.config`), éditez la liste avec les noms des champs dans *Basic information* qui peuvent être enregistrés.

- Il peut être utile de ne pas autoriser la modification des champs nom et prénom (surtout s'ils sont mis à jour avec LDAP) :

```
// Make first and last name read-only
config.Profile.configuration.defaultConfig.basic.elements.firstName.editable = false;
config.Profile.configuration.defaultConfig.basic.elements.lastName.editable = false;
```

- Certaines configurations sont accessibles via les [forks GitHub de d'autres universités/personnes](#).

- Pour personnaliser les champs du profil avec du code JS, le code des modifications devrait être appelé après le `widgetdisplayprofilesection` qui fait le rendu de la page (côté formulaire) ou alors l'intégrer dans ce widget.

- Pour les listes déroulantes, les options dans `config_custom.js` se font par défaut en statique.

Validation des champs

Dans votre `config_custom.js`, si vous voulez valider votre champ "truc" avec un validateur appelé ici "muche" :

```
"truc": {
  "label": "some label",
  "required": false,
  "display": true,
  "validation": "muche"
}
```

Vous aurez besoin de rajouter au plugin `Validate` de jQuery, une méthode de validation via [addMethod](#), on pourrait mettre ce code à la suite dans `config_custom.js` (why not?)

```
require(["jquery", "jquery-plugins/jquery.validate"], function(jQuery) {
  /**
   * Add some jquery validate methods
   */

  // Exemple avec un validateur "muche"
  $.validator.addMethod("muche", function(value, element){
    return true; // Faites votre validation et renvoyez true si on accepte
  }, "Le message d'erreur si c'est pas bon est celui-là");
});
```

En revanche, il ne me semble pas qu'une validation soit bien effectuée côté serveur. Je n'ai rien trouvé côté code mais je n'ai bizarrement pas pu le confirmer en pratique sur le champ adresse mail.