

Sujet : Front Code Challenge

De : Jonathan Geoffroy <geoffroy.j@sfeir.com>

Date : 03/08/2022 17:04

Pour : Melvin DOUCET <doucet.m@sfeir.com>

Melvin,

Comme nous en avons discuté tous les deux, tu m'as posé une très bonne question: pourquoi utiliser ReactJS plutôt que VanillaJS, alors que ReactJS complexifie ton application.

Je suis d'accord avec toi que ReactJS ajoute un overhead, et qu'il peut être difficile de voir l'intérêt sur un petit projet.

Aussi je te propose un challenge un peu plus gros. Tu peux le développer dans le framework de ton choix, ou en VanillaJS sans aucun autre framework.

Pour ma part, je participerai au challenge, en prenant SolidJS, un framework qui ressemble beaucoup à ReactJS sur son principe.

Je te propose de prendre ce challenge comme objectif cette année chez Sfeir (vu avec Rodrigue). Pour rappel, un objectif n'est jamais obligatoire. Bien entendu, prendre et réussir un objectif est toujours bien vu : en validant ton objectif, tu pourras prétendre à une augmentation de salaire. Enfin, même si tu ne réussis pas ou pas totalement un objectif, si tu y as mis de la bonne volonté, on pourra le valider en partie. Bref, nous en reparlerons très bientôt :)

Pour revenir au challenge : pour t'aider dans le développement, tu trouveras un serveur avec des données à disposition ici : <https://github.com/melvin-exercise/server>

Voici les règles du jeu :

1. Aucune autre library : pas de material-ui, pas de react-table, pas de jquery-select etc. Tous les composants sont à développer par nous même. Les technologies à disposition dans les navigateurs modernes sont en revanche autorisées.
2. Le challenge propose déjà un serveur CRUD sur l'ensemble des données utiles. Il est interdit de changer le modèle de données ou de rajouter des fonctionnalités dans le serveur.
3. Le challenger peut décider de développer toutes les fonctionnalités, ou une partie uniquement. De plus, il peut décider de modifier certaines règles, à condition qu'il puisse montrer en quoi cela est meilleur pour l'utilisateur final (UX plus simple, plus jolie, plus facilement compréhensible etc). Bien évidemment, plus il y a de fonctionnalités, mieux c'est !
4. Le design est laissé libre. Le but du challenge est surtout de comparer le code, pas le design. Cependant, un beau design, c'est toujours mieux également :)

Critère de réussite :

- Qualité / lisibilité du code (ce critère étant parfois subjectif, ou pourra en débattre :))
- Rapidité d'exécution, absence de lags etc
- Compatibilité des browsers "modernes" : chrome / firefox / edge.
- Gestion des erreurs (exemple : le serveur n'est plus joignable, le serveur refuse la modification etc)

Scénario :

Le but du challenge est de développer cette interface :

Exercice

Users

Name	Email	Gender	Address	Company	Roles	Actions
Ada Lovelace	adalovelace@gmail.com	Female	15 Liberty Avenue, Gerber, Nevada	Rooftoria	Contributor	
Grace Hopper	gracehopper@yahoo.lu	Female	31 Fenimore Street, Celeryville, Virginia	Vicon	Guest	
Robert Martin	robertmartin@sfeir.com	Male	4 Clark Street, Bynum, Georgia	Gink	Guest	
John Doe	john@doe.com	Male		Company	3 items selected	

History

- Delete address
- update address
"Old search" => "New search"
- Add address
"Old search"
- Update gender
"Female" => "Male"
- Add gender
"Male"

Comportement des données du tableau

Affichage des données

Le tableau doit contenir l'ensemble des informations sur les utilisateurs, à condition qu'il "match" tous les filtres (filtre1 AND filtre2 AND ...). L'application offre un filtre par colonne, comme illustré plus haut.

Par exemple, si l'utilisateur choisit : Gender=femelle et Companies="Sfeir, Diginetic", alors les lignes affichées doivent être des femmes, soit chez Sfeir soit chez Diginetic.

De plus amples explications sur les filtres sont données plus bas.

Modification d'une ligne

A tout moment l'utilisateur peut modifier une ligne en cliquant sur le bouton associé dans la colonne action.

Dans ce cas, les champs de la ligne se transforment en champs texte ou selectbox en fonction de la colonne.

Pour l'entreprise, toutes les options doivent être sélectionnables (ne pas tenir compte du filtre en haut de la colonne). Une seule entreprise est sélectionnable.

Pour le selectbox gender, seuls "male" et "female" sont sélectionnables (pas de "All").

L'utilisateur peut cliquer sur "valider", auquel cas la ligne doit être mise à jour, en front comme en back.

L'utilisateur peut aussi cliquer sur "cancel", auquel cas la ligne revient à son état initial.

Dans les deux cas la ligne repasse en mode lecture (texte et non plus champs éditables)

A noter : on ne peut éditer qu'une seule ligne à la fois.

De plus, lorsqu'on édite une ligne, les filtres et tris sont désactivés. (les lignes restent triées et filtrées, mais les filtres passent en read-only et l'utilisateur ne peut plus les modifier)

Filtre des données

Pour le champ "entreprise", 0, 1 ou plusieurs valeurs sont sélectionnables. Le champ "match" le filtre si au moins une sélection correspond au champ (option1 OR option2 OR option3)

Pour le champ "Gender", un seul choix est possible. Le choix "All" est activé par défaut, dans ce cas il match toutes les lignes.

Dans le cas où le filtre est un champs texte, une ligne "match" le filtre si il contient

exactement le champs rentré, case insensitive, et trim (le filtre " something " match donc le champs "It's Something Great")

Les données du tableau doivent être filtrées dès qu'un filtre change, c'est-à-dire dès qu'une lettre est tapée dans un champ texte, dès qu'une option est modifiée pour les selectboxes.

Tri des données

Cliquer sur une colonne du tableau permet de modifier le tri du tableau sur cette colonne :

1. premier clic : tri ascendant
2. deuxième clic : tri descendant
3. troisième clic : pas de tri (ordre tel que renvoyé par le backend).

Chaque colonne possède son système de tri :

- Gender : Female en premier pour le tri ascendant
- Rôles : l'ordre donné par le serveur correspond au tri ascendant
- Les autres champs textes : ordre alphabétique, case-insentive, après trim.

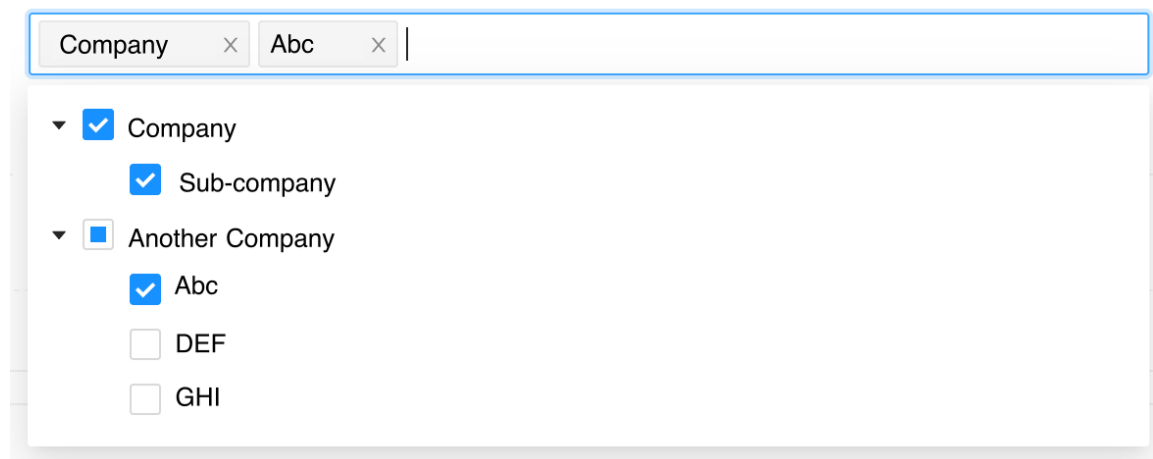
A noter : la colonne "Actions" n'est ni triable, ni filtrable.

Comportement des filtres

Multi-select récursif : Companies (Moyen)

Le serveur permet de récupérer l'ensemble des entreprises. Dans ce modèle, une entreprise peut avoir des filiales ("sous-entreprises").

Le filtre "companies" doit refléter ce modèle en proposant une arborescence récursive de ses options. L'idée est de reproduire un comportement similaire à celui ci : <https://ant.design/components/tree-select/> (voir "multiselection").



Interdépendance des sélections (Difficile)

Les filtres textes sont des champs libres, et l'utilisateur est libre de taper ce qu'il souhaite. Cela peut donc résulter en un tableau vide, car aucune ligne ne match ce champ

En revanche, les filtres selectboxes ne proposent que des options qui sont dans le tableau. Si un filtre est utilisé, les selectboxes doivent réduire leurs options disponibles, sauf s' il s'agit d'une option déjà activée auparavant (pour que l'utilisateur puisse la décocher facilement).

Par exemple :

1. L'utilisateur choisit les entreprises "A", "B", "C". Le filtre continue de permettre la sélection des autres entreprises.
2. Les données du tableau sont filtrées, on ne retrouve plus que des utilisateurs pour les entreprises "A", "B" ou "C".
3. Les autres selectboxes (Rôles, Gender), peuvent avoir des choix réduits (par exemple: uniquement "Admin" si il n'y a plus que des admin
4. L'utilisateur filtre sur le nom "Grace". Il n'y a plus que des utilisateurs de la company "A" pour cet exemple.
5. Les filtres se réduisent encore. Le filtre companies affichera "A" car il reste des données ; "B" et "C" car ils étaient sélectionnées auparavant. En revanche, les autres entreprises ne sont plus disponibles.

Attention : le filtre entreprise étant un arbre récursif, les valeurs parentes doivent également être gardées si elles sont sélectionnées, ou si un enfant au moins est sélectionné.

A noter : un pattern plus "user-friendly" peut aussi être proposé. Dans ce cas, on demandera à d'autres personnes de tester l'application pour voir ce qu'ils préfèrent.

Enfin, la croix tout à droite (voir maquette) permet de réinitialiser tous les filtres. Le tableau de données affiche alors de nouveau tous les utilisateurs.

Historique (difficile)

Le panel de droite propose un historique des filtres. Chaque fois que l'utilisateur modifie un filtre, une ligne s'ajoute dans l'historique.

Chaque ligne doit permettre à l'utilisateur de comprendre la modification effectuée.

L'utilisateur peut ensuite cliquer sur une ligne pour repositionner les filtres à ce point dans le temps.

A noter : pour améliorer l'expérience utilisateur, on peut ne rajouter qu'une seule ligne pour les filtres de texte, plutôt qu'une ligne pour chaque lettre.

Bonne chance, bon courage ! Et surtout : j'espère que tu t'amuseras bien si tu acceptes de relever le défi :)

A bientôt,
Jonathan