Dans la dernière livraison sur test, il y a les fixes de #2621 et #5103.

J'ai ajouté une nouvelle api pour les table scripts, avec un typage plus fort des colonnes (ce qui permet d'avoir une grid kendo et un export excel correct)

L'api est la:

<https://github.com/Vinciguerra/Domos/tree/%40next/domos/dms-script/src/main/java/com/domos/js/display/typed>

Dans le script, la nouvelle api est accessible avec out() (table() pour l'ancienne version).

L'idée est d'utiliser cette api à la place de l'ancienne (DJsDisplayTable).

A noter que vous pouvez désormais définir des groupes de colonnes (#5103).

Un exemple de script est sur test/domosDev: composant NeoTable dans le dashboard NeoDashboard.

Pour info, le script de ce composant a été généré à partir d'une version typescript.

Le source code orignal est la:

<https://github.com/Vinciguerra/Domos/blob/%40next/ts-api/src/table.component.ts>

J'ai ajouté des(quelques) interfaces typescript pour typer fortement les scripts (table.component.ts est typé à 100%).

Notamment, la description entière de la nouvelle api est disponible dans ce fichier (il manque tous les commentaires (TODO)):

<https://github.com/Vinciguerra/Domos/blob/%40next/ts-api/src/domos/display/typed/DJsTypedTable.ts>

Le projet typescript entier est la:

<https://github.com/Vinciguerra/Domos/tree/%40next/ts-api>

Si vous ouvrez le dossier avec Visual Studio Code, vous pouvez écrire vos scripts avec du typage fort, autocompletion et cie

(ce qui fait gagner beaucoup de temps lors de l'écriture des scripts).

(a priori, il vous faut node.js. "npm install" pour installer les dépendances et "npm run build" pour générer le javascript (l'output est dans lib). Demandez moi si vous voulez essayer).

Bien évidemment, les interfaces qui mappent l'api sont encore très incomplètes.

Mon idée pour le futur est de les générer ou de les enrichir au fur et à mesure, et à terme d'offrir la possibilité d'écrire directement les scripts en typescript dans domos.