Programmation Fonctionnelle : des concepts aux applications web (NFP119)

Interface Utilisateur (UI)

Dans ce TP, nous utiliserons la bibliothèque de composants Grommet (basée sur React) pour concevoir une interface utilisateur premettant d'afficher les données du TP précédent. La documentation de Grommet est disponible en ligne :

https://v2.grommet.io

Cette bibliothèque permet le développement d'une interface utilisateur moderne, adaptative (*responsive*) et accessible. Un outil en ligne associé, Grommet Designer, permettant la conception visuelle (*visual designer*) d'un prototype est aussi disponible :

https://designer.grommet.io

Nous l'utiliserons pour la suite de ce TP :

1. Ouvrir l'exemple Card dans Grommet Designer.

2. Cliquer sur Repeater à gauche et, à droite, donner à count la valeur 1.

3. Cliquer sur Card Box à gauche, puis sur Box ⋁ et reset. Ensuite, à droite :

   – cliquer sur background ⋁ puis color et choisir white

   – cliquer sur round et choisir medium

   – cliquer sur margin et choisir aussi medium

4. Cliquer à nouveau sur "Card Box" à gauche, puis sur Box ⋁ et show code... en haut à droite. La fenêtre qui s'ouvre vous affiche le code React/Grommet correspondant à l'objet actuellement selectionné.

5. Cliquer sur Card ⋁ puis share à gauche, puis Generate code, et vous obtenez le code complet de l'application. Selectionner et copier ce code.

6. Créer une application React/Grommet en suivant le tutorial en ligne (cf Annexe).

7. Remplacer le contenu de App.tsx par le code copié auparavant. Tester l'application.

8. Supprimer le thème dans App.tsx et tester à nouveau l'application. Supprimer le paramètre full et tester à nouveau l'application.

9. Cliquer sur Card ⋁ puis share à gauche, puis configure. Dans Data, conserver data comme Name, ajouter comme Source :

   https://jsonplaceholder.typicode.com/users

10. Cliquer sur Repeater à gauche et, à droite, donner à dataPath à la valeur data (mais laisser la valeur de count à 1).

11. Cliquer sur le nom de la première carte et, à droite :
    - donner à name la valeur Name, puis remplacer le texte par {name}
    - donner à name la valeur Email, puis remplacer le texte par {email}
    - remplacer le texte de la Description par {company.catchPhrase}
12. Dans l'application, créer un fichier Users.ts contenant les déclarations de type et de la constante users du TP précédent, et ajouter à la fin les lignes suivantes :

```
export type { Geo, Address, Company, User }

export { users }
```

13. Ajouter les imports correspondants dans App.tsx :

```
import type { Geo, Address, Company, User } from './Users'

import { users } from './Users'
```

14. Encadrer la Box correspondant à une carte par :

```
{users.map(({ name, email, company }: User) =>

    ...

)}
```

où ". . . " est remplacé par le code de cette Box.

15. Revenir à Grommet Designer et ajouter un Accordion dans le bas d'une carte (juste au dessous de la Description), puis une AccordionPanel à l'intérieur. Donner la valeur Address au label et remplir le panel avec des champs Text contenant l'adresse de l'utilisateur.
16. Déplacer le code pour la Box correspondant à une carte dans un ficher Card.tsx. Le composant fonctionnel (FC) contiendra alors (en plus des autres imports) :

```
import { FC } from 'react'

const Card: FC<{ name: string, email: string, company: Company }> =

    ({ name, email, company }) => {

        return (

            ...

        )

    }

export default Card;
```

17. Modifier aussi le code de App.tsx ainsi :

```
{users.map(({ name, email, company }: User) =>

    <Card name={name} email={email} company={company} />

)}
```

Annexe

Le tutoriel en ligne ci-dessous décrit comment créer une application React/Grommet :

https://cedric.cnam.fr/sys/crolard/enseignement/USAL3A/grommet-cards.html

L'utilisation de create-react-app nécéssite au moins nodejs 14. Pour les versions antérieures de nodejs, il est toutefois possible de compiler une archive pré-existante d'un projet, avec la commande suivante :

npm install

Alternativement, il est aussi possible d'installer yarn (à la racine du compte) :

npm install –prefix . -g yarn

et de compiler ensuite le projet ainsi :

yarn install

Visual Studio Code doit normalement reconnaitre les deux configurations (npm ou yarn).