

Atelier 2 :

Les components

Objectifs

Cet atelier vise à manipuler une des notions fondamentales dans Angular, à savoir les components. A la fin de cet atelier vous serez capable de :

1. Comprendre en pratique le rôle des components,
2. Utiliser les paramètres composant un component
3. Faire interagir des components différents.

Enoncé

Les components définissent des zones de responsabilité dans notre interface utilisateur (UI) qui nous permettent de réutiliser ces ensembles de fonctionnalités de l'UI. Nous en avons déjà construit un avec le component de liste de produits.

Dans Angular, un component est principalement composé de trois choses :

- **Une classe de type component** : qui gère les données et les fonctionnalités. Dans l'atelier précédent, les données de produit et la méthode `share()` ont été définies pour vous dans la classe de type composant.
- **Un template HTML** : qui détermine ce qui est présenté à l'utilisateur. Dans l'atelier précédent, nous avons modifié le template HTML de la liste de produits pour afficher le nom, la description et un bouton "Partager" pour chaque produit.
- **Styles spécifiques du component** : qui définissent l'apparence. La liste de produits ne définit aucun style.

Une application Angular est composée d'une arborescence de components, dans laquelle chaque component Angular a un objectif et une responsabilité spécifiques.

Actuellement, notre application a trois components :

- **app-root** (cadre rouge) est le shell de l'application. C'est le premier component à charger et le parent de tous les autres components. Vous pouvez le considérer comme la page de base.
- **app-top-bar** (cadre orange) est le nom du magasin et le bouton de paiement.
- **app-liste-produit** (cadre vert) est la liste de produits que nous avons modifiée l'atelier précédent.

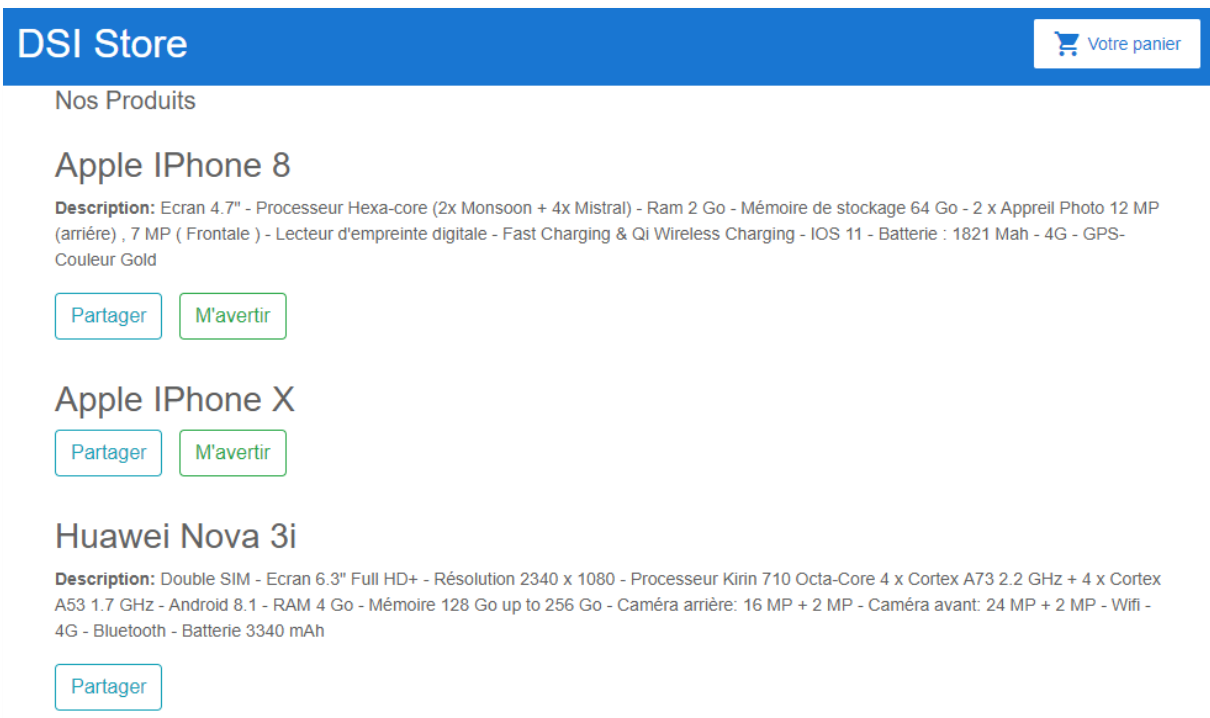


Travail à faire

Nous allons créer une nouvelle fonctionnalité d'alerte. La fonctionnalité d'alerte prendra un produit en tant qu'entrée. Il vérifiera ensuite le prix du produit et, si le prix est supérieur à 3000 DT, un bouton "M'avertir" apparaîtra pour permettre aux utilisateurs de s'inscrire pour recevoir des notifications lorsque le produit sera en promotion.

1. Nous commençons par générer un nouveau composant qu'on nomme alertes-produit.
2. Dans le fichier alertes-produit.component.ts, et à l'intérieur de la classe AlertesProduitComponent, nous devons définir une propriété nommée produit avec le décorateur @Input, ce décorateur indiquera à Angular que la valeur de cette propriété sera passée depuis son composant parent. Il ne faut pas oublier d'importer la classe input depuis le @angular/core.

3. Dans le fichier `alertes-produit.component.html`, nous ajoutons le bouton "M'avertir" qui n'apparaît que si le prix du produit dépasse les 3000 DT.
4. Le nouveau component `alertes-produit` sera un fils du component `liste-produit`. Ajoutons dans le fichier `liste-produit.component.html` son nouveau fils en ajoutons une liaison de propriété du produit pour le passer au component fils comme prévu. Le résultat à ce stade ressemblera à ceci :



5. Dans le fichier `alertes-produit.component.ts`, et à l'intérieur de la classe `AlertesProduitComponent`, nous devons définir une propriété nommée `alerte` avec le décorateur `@Output`, ce décorateur indiquera à Angular que la valeur de cette propriété sera envoyée vers le component parent. Cette propriété sera une instance de la classe Angular pré-définie `EventEmitter`. Il ne faut pas oublier d'importer les classes `input` et `EventEmitter` depuis le `@angular/core`.
6. Nous mettrons à jour le bouton "M'avertir" dans le fichier `alertes-produit.component.html` en ajoutant une liaison d'événement pour appeler une fonction `alerte.emit()`.
7. Par la suite, nous définissons le comportement que doit se produire lors d'un clic sur le bouton. Il ne faut pas oublier que le component parent qui va recevoir l'action. Dans le fichier `liste-produit.component.ts`, nous ajoutons une fonction `onNotify()` qui permet d'afficher une alerte dans le navigateur "Vous serez alertés dès que ce produit sera en promotion".
8. Finalement, nous ajoutons une liaison d'événement dans le fichier `liste-produit.component.html` pour lier le `alertes-produit` à la fonction `onNotify()`.