

## Atelier 4 : Les services

### Objectifs

Cet atelier vise à utiliser la notion de services dans le framework Angular. A la fin de cet atelier vous serez capable de :

1. créer un service,
2. intégrer le service dans l'application,
3. pratiquer d'avantage l'injection de dépendance.

### Enoncé

À la fin de l'atelier précédent, notre application de boutique en ligne dispose d'un catalogue de produits avec deux vues :

- une liste complète de tous nos produits,
- les détails de chaque produit.

L'utilisateur peut cliquer sur le nom d'un produit dans la liste pour afficher les détails dans une nouvelle vue, avec une URL distincte (route).

Dans cette section, nous allons créer le panier en :

- Mettant à jour la page de détails du produit pour inclure un bouton "Acheter", qui ajoute le produit actuel à une liste de produits gérés par un service de panier.
- Ajoutant un component de panier, qui affiche les articles qu'on ajoute au panier.

### Travail à faire

1. Tout d'abord, nous commençons par générer un nouveau service qu'on nomme panier.
2. Dans le service qu'on vient de créer, nous définissons une propriété de type tableau qu'on nomme achats et qui aura pour rôle de stocker les achats, ainsi que trois méthodes :
  - a. **ajouterAuPanier(produit)** : qui ajoute un produit au tableau des achats.
  - b. **getAchats()** : qui retourne le contenu du panier.
  - c. **viderPanier()** : qui supprime le contenu du tableau achats et donc le contenu du panier

3. Revenons maintenant au fichier `details-produit.component.ts`, dans lequel on va importer le service que nous venons de créer et puis l'injecter dans le constructeur de la classe. Nous définissons par la suite une autre méthode `ajouterAuPanier(produit)` qui permet d'afficher une alerte "Produit ajouté au panier", puis faire appel à la méthode `ajouterAuPanier()` du service `panier` en lui passant le produit en paramètre.
4. Nous modifions le `details-prouduit.component.html` en ajoutant un bouton "Acheter", avec une liaison d'événement pour lancer la méthode `ajouterAuPanier()` qu'on vient de définir dans la question 3.

À ce stade, les utilisateurs peuvent mettre des articles dans le panier en cliquant sur "Acheter", mais ils ne peuvent pas encore voir leur panier.

Nous allons créer la page de panier en commençant par créer un component de panier et configurer le routage vers le nouveau component, à ce stade la page du panier n'aura que du texte par défaut, puis afficher les articles du panier.

5. Nous générons un nouveau component nommé `panier`.
6. Nous importons dans ce component le service que nous avons créé, et nous l'injectons dans le constructeur de la classe. Nous ajoutons une propriété qu'on nomme `achats` (qui aura pour rôle de stocker les achats ajoutés au panier), puis, dans le `ngOnInit()`, on affecte dans cette propriété le résultat de la méthode `getAchats()` du `PanierService`.
7. Dans le template `panier.component.html`, nous ajoutons une boucle qui permet de parcourir la liste des achats pour afficher, pour chaque produit sélectionné, le nom ainsi que le prix.
8. Dans le `app.module.ts`, nous ajoutons une nouvelle route vers le panier et qui pointe vers le component créé dans la question 5. Il est à noter que le bouton "Votre Panier" est déjà configuré avec un lien vers le panier.
9. Pour tester la fonctionnalité de panier que venons de mettre en place nous devons :
  - a. Dans la liste des produits cliquer sur le nom du produit puis sur le bouton "Acheter".
  - b. Cliquer sur le bouton "Votre Panier" pour voir les achats.
  - c. Pour ajouter d'autres produits dans les achats on refait l'étape a.