

Atelier 5 : HttpClient

Objectifs

Cet atelier vise à mettre en pratique la notion de client Http du framework Angular. A la fin de cet atelier vous serez capable de :

1. définir un HttpClient dans l'application,
2. l'injecter dans un service.

Enoncé

Les données renvoyées par les serveurs prennent souvent la forme d'un flux. Les flux sont utiles car ils facilitent la transformation des données renvoyées et la modification de la manière dont les données sont demandées. Le client HTTP d'Angular (`HttpClient`) est un moyen intégré pour extraire des données d'API externes et de les transmettre à votre application sous forme de flux.

Avant de pouvoir utiliser le client HTTP d'Angular, nous devons configurer notre application pour utiliser `HttpClientModule`.

Le `HttpClientModule` d'Angular enregistre les providers nécessaires pour utiliser une seule instance du service `HttpClient` dans notre application. Le service `HttpClient` est ce que nous injectons dans nos services pour extraire des données et interagir avec des API et des ressources externes.

Dans cet atelier, nous utiliserons le fichier `assets/livraison.json` qui jouera le rôle d'API pour notre application pour avoir la liste des prix des frais de livraison.

Travail à faire

1. Nous allons entamer le travail par activer le `HttpClient` pour notre application. Dans le fichier `app.module.ts`, nous importons le `HttpClientModule` depuis le package `@angular/common/http`. Nous ajoutons ensuite ce module dans le tableau des imports. Ceci permet d'enregistrer les providers du `HttpClient` globalement au sein de notre application.
2. Nous procédons maintenant à activer le `HttpClient` pour le service qu'on a créé dans l'atelier précédent. Dans le fichier `panier.service.ts`, nous importons le

HttpClient depuis le package `@angular/common/http`, puis nous l'injectons dans le constructeur de la classe.

3. Dans le même fichier, nous définissons une nouvelle méthode nommée `getFraisLivraison()`, qui utilisera la méthode prédéfinie `get()` du `HttpClient` afin d'extraire les détails relatifs aux frais de livraison depuis le fichier `assets/livraison.json`.
4. Nous générons maintenant un nouveau composant `livraison` puis nous importons le `PanierService` dans ce composant et nous l'injectons dans le constructeur de la classe.
5. Nous définissons par la suite une nouvelle propriété dans ce même composant une nouvelle propriété `fraisLivraison`, puis nous lui attribuons, dans la méthode `ngOnInit()`, le résultat de retour de la méthode `getFraisLivraison()` qu'on a précédemment définie dans le `PanierService`.
6. Passons maintenant au `livraison.component.html` pour afficher la liste des prix pour les frais de livraison depuis la propriété qu'on a créé dans la question précédente.
7. Nous revenons ensuite au fichier `app.module.ts` pour définir une nouvelle route vers le composant `LivraisonComponent`, puis un lien (menant vers cette route) dans le fichier `panier.component.html`.