

Atelier 6 : Les formulaires

Objectifs

Cet atelier a pour but d'utiliser un formulaire dans le framework Angular. A la fin de cet atelier vous serez capable de :

1. intégrer et paramétrer un formulaire,
2. récupérer et manipuler les données renvoyées par le formulaire dans l'application.

Enoncé

À la fin de l'atelier précédent, l'application de boutique en ligne comporte un catalogue de produits et un panier.

Dans cet atelier, nous terminerons l'application en ajoutant une fonctionnalité de paiement basée sur un formulaire. Nous allons créer un formulaire pour collecter les informations utilisateur dans le cadre du paiement.

Les formulaires dans Angular utilisent les fonctionnalités standard des formulaires HTML et ajoutent une couche d'orchestration pour aider à la création des contrôles personnalisés et pour offrir une expérience de validation de haut niveau. Un formulaire réactif dans Angular se compose de deux parties : les objets qui résident dans le component pour stocker et gérer le formulaire et la visualisation du formulaire qui réside dans le template.

Travail à faire

1. On va commencer par les importations. Dans le fichier `app.module.ts`, nous importons le `ReactiveFormsModule` depuis `@angular/forms`, puis nous l'ajoutons dans le tableau des imports.
2. Dans le `panier.component.ts`, nous importons le `FormBuilder` depuis `@angular/forms`. Nous injectons par la suite ce service dans le constructeur de la classe.
3. Nous définissons ensuite une nouvelle propriété `verifierForm` qui permettra de stocker le formulaire que nous allons créer.
4. Nous souhaitons demander à l'utilisateur de fournir son nom et son adresse. Nous affectons donc à cette propriété un modèle de formulaire contenant ces deux champs en utilisant la méthode `group()` prédéfinie dans le `FormBuilder`.

5. Une fois le nom et l'adresse sont envoyés par l'utilisateur via le formulaire, nous procéderons à vider les champs du formulaire ainsi que les achats effectués dans le panier. Dans la pratique réel, cette méthode servira à envoyer les données récupérées vers un serveur. Nous allons définir une nouvelle méthode `onSubmit(infoClient)`, qui :
 - a. affichera dans une alerte le nom et l'adresse du client depuis la variable reçue en paramètre,
 - b. vide le panier en utilisant la méthode `viderPanier()` créée précédemment dans le `PanierService`
 - c. vider les champs du formulaire.
6. Nous passons maintenant au fichier `panier.component.html`. Nous ajoutons à la fin de ce template un formulaire avec la liaison de la propriété `formGroup` avec la valeur `verifierForm`, ainsi qu'une liaison de l'événement `ngSubmit` qui lancera la méthode `onSubmit()` créée dans la question précédente avec les valeurs du `verifierForm`. Nous ajoutons aussi à ce formulaire un bouton `submit`.
7. Nous continuons avec le `panier.component.html` pour ajouter les deux inputs pour le nom et pour l'adresse du client. Nous utiliserons l'attribut `formControlName` pour faire la liaison entre le `verifierForm` et les noms des inputs que nous sommes en train de créer.