

[Skip to main content](#)

Using forms for user input

This guide builds on the [Managing Data](#) step of the Getting Started tutorial, [Get started with a basic Angular app](#).

This section walks you through adding a form-based checkout feature to collect user information as part of checkout.

Define the checkout form model

This step shows you how to set up the checkout form model in the component class. The form model determines the status of the form.

[Skip to main content](#)

2. Import the `FormBuilder` service from the `@angular/forms`

package. This service provides convenient methods for generating controls.

src/app/cart/cart.component.ts

```
import { Component } from '@angular/core';
import { FormBuilder } from '@angular/forms';

import { CartService } from '../cart.service';
```

3. Inject the `FormBuilder` service in the `CartComponent` `constructor()`. This service is part of the `ReactiveFormsModule` module, which you've already imported.

src/app/cart/cart.component.ts

```
export class CartComponent {

  constructor(
    private cartService: CartService,
    private formBuilder: FormBuilder,
  ) {}
}
```

4. To gather the user's name and address, use the `FormBuilder` `group()` method to set the `checkoutForm` property to a form model containing `name` and `address` fields.

[Skip to main content](#)

cart.component.ts

```
export class CartComponent {

  items = this.cartService.getItems();

  checkoutForm = this.formBuilder.group({
    name: '',
    address: ''
  });

  constructor(
    private cartService: CartService,
    private formBuilder: FormBuilder,
  ) {}
}
```

5. Define an `onSubmit()` method to process the form. This method allows users to submit their name and address. In addition, this method uses the `clearCart()` method of the `CartService` to reset the form and clear the cart.

The entire cart component class is as follows:

[Skip to main content](#)

cart.component.ts

```
import { Component } from '@angular/core';
import { FormBuilder } from '@angular/forms';

import { CartService } from '../cart.service';

@Component({
  selector: 'app-cart',
  templateUrl: './cart.component.html',
  styleUrls: ['./cart.component.css']
})
export class CartComponent {

  items = this.cartService.getItems();

  checkoutForm = this.formBuilder.group({
    name: '',
    address: ''
  });

  constructor(
    private cartService: CartService,
    private formBuilder: FormBuilder,
  ) {}

  onSubmit(): void {
    // Process checkout data here
    this.items = this.cartService.clearCart();
    console.warn('Your order has been submitted',
    this.checkoutForm.value);
    this.checkoutForm.reset();
  }
}
```

[Skip to main content](#)

Create the checkout form

Use the following steps to add a checkout form at the bottom of the Cart view.

[Skip to main content](#)

`cart.component.html`, add an HTML `<form>` purchase button.

2. Use a `formGroup` property binding to bind `checkoutForm` to the HTML `<form>`.

src/app/cart/cart.component.html

```
<form [formGroup]="checkoutForm">

  <button class="button"
    type="submit">Purchase</button>

</form>
```

3. On the `form` tag, use an `ngSubmit` event binding to listen for the form submission and call the `onSubmit()` method with the `checkoutForm` value.

src/app/cart/cart.component.html (cart component template detail)

```
<form [formGroup]="checkoutForm"
  (ngSubmit)="onSubmit()">
</form>
```

4. Add `<input>` fields for `name` and `address`, each with a `formControlName` attribute that binds to the `checkoutForm` form controls for `name` and `address` to their `<input>` fields. The complete component is as follows:

[Skip to main content](#)

cart.component.html

```
<h3>Cart</h3>

<p>
  <a routerLink="/shipping">Shipping Prices</a>
</p>

<div class="cart-item" *ngFor="let item of items">
  <span>{{ item.name }} </span>
  <span>{{ item.price | currency }}</span>
</div>

<form [formGroup]="checkoutForm"
  (ngSubmit)="onSubmit()">

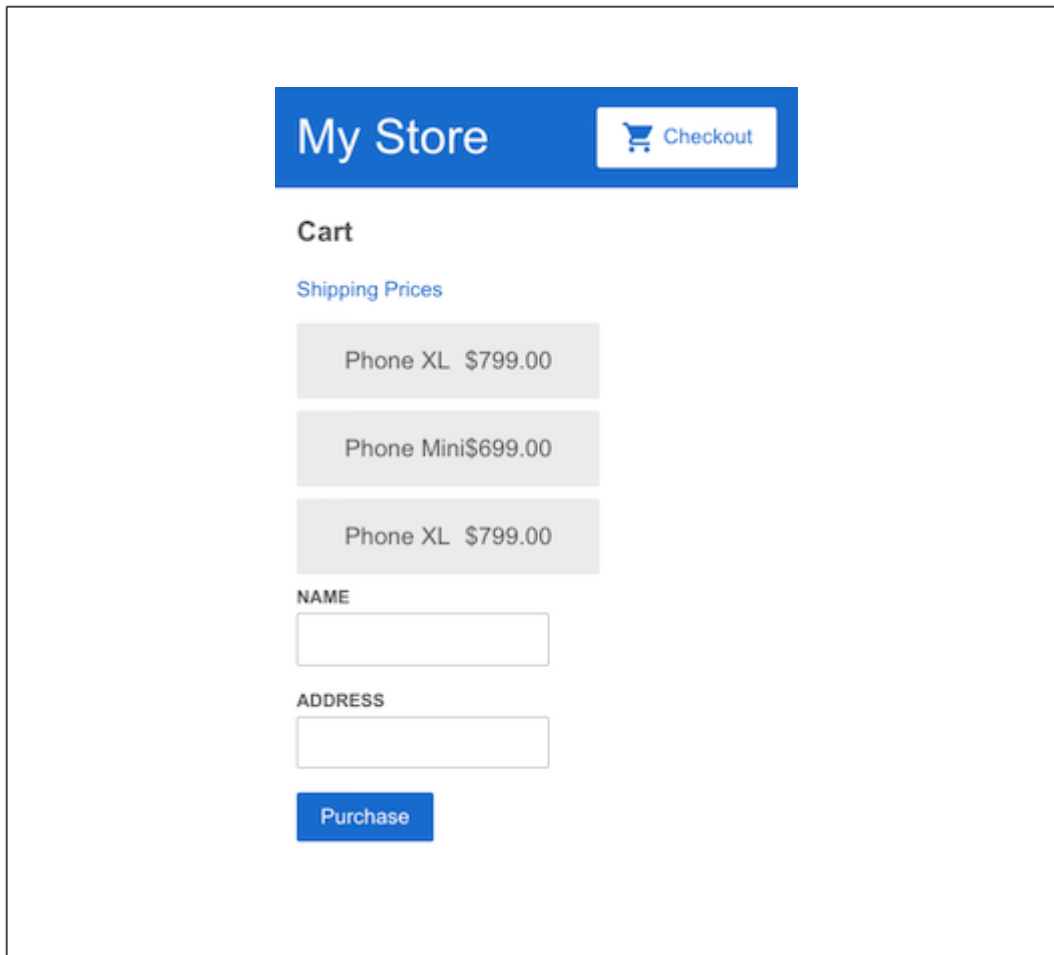
  <div>
    <label for="name">
      Name
    </label>
    <input id="name" type="text"
formControlName="name">
  </div>

  <div>
    <label for="address">
      Address
    </label>
    <input id="address" type="text"
formControlName="address">
  </div>

  <button class="button"
type="submit">Purchase</button>

</form>
```

After putting a few items in the cart, users can review their items, enter their name and address, and submit their purchase.



The screenshot shows a web application interface for 'My Store'. At the top, there is a blue header bar with the text 'My Store' on the left and a 'Checkout' button with a shopping cart icon on the right. Below the header, the section is titled 'Cart'. Underneath the title, there is a link for 'Shipping Prices'. The cart contains three items, each in a light gray box: 'Phone XL \$799.00', 'Phone Mini\$699.00', and 'Phone XL \$799.00'. Below the cart items, there are two form fields: 'NAME' and 'ADDRESS', each with a corresponding text input box. At the bottom of the form, there is a blue 'Purchase' button.

To confirm submission, open the console to see an object containing the name and address you submitted.

What's next

You have a complete online store application with a product catalog, a shopping cart, and a checkout function.

[Continue to the "Deployment" section](#) to move to local development, or deploy your app to Firebase or your own server.

Last reviewed on Wed Sep 15 2021