# Create a New Angular Component

## 1. Setup Project

### 1.1 Install Dependencies

1. Change directory to calab:

* cd calab

1. Install dependencies by running the following command:

* npm install

1. You should see a message in your Terminal confirming the npm packages were installed successfully: 

## 2. Create Components

### 2.1 Create Product Component

1. Create a new component called Product using CLI:

* npx -p @angular/cli ng generate component components/product

### 2.2 Create ProductList Component

1. Create a new component called ProductList using CLI:

* npx -p @angular/cli ng generate component components/product-list

### 2.3 Import ProductComponent

1. Import ProductComponent into src/app/components/product-list/product-list.component.ts:

* import { ProductComponent } from '../product/product.component';

1. Inside src/app/components/product-list/product-list.component.ts update imports to include ProductComponent:

* imports: [ProductComponent],

1. Open src/app/components/product-list/product-list.component.html template and add the following element:

* <app-product></app-product>

### 2.4 Import ProductListComponent

1. Import ProductListComponent into src/app/app.component.ts:

* import { ProductListComponent } from './components/product-list/product-list.component';

1. Inside src/app/app.component.ts update imports to include ProductListComponent:

* imports: [RouterOutlet, ProductListComponent],

1. Open src/app/app.component.html template and add the following element after the <div class="divider"...

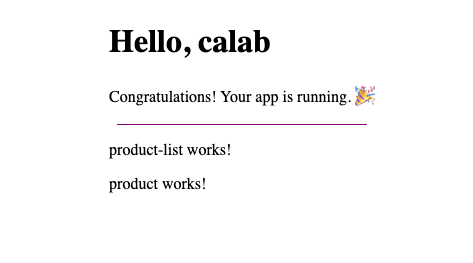
* <app-product-list></app-product-list>

### 2.5 Start The Application

1. Start Angular Development Server if not yet started:

* npx -p @angular/cli ng serve --host 0.0.0.0
* *Otherwise refresh the browser tab to see updated view.*

1. You should see the following getting rendered in your browser:

* 

## 3. Create Parent Child Communication between Components

### 3.1 Modify ProductListComponent

1. Open src/app/components/product-list/product-list.component.ts file and inside ProductListComponent class do the following:
   * Declare an emty cart list:
   * cart: Product[] = [];
   * Declare a Product:
   * product = new Product('Product A', 10.99);
   * Create a method called addToCart() that is takes product as a parameter and pushes that product to the cart:
   * addToCart(product: Product) {  
      this.cart.push(product);  
     }
2. Open src/app/components/product-list/product-lis.component.html file and do the following:
   * Display cart list length. Just before <app-product> add the following:
   * Cart:{{cart.length}}

### 3.2 Modify ProductComponent

1. Open src/app/components/product/product.component.ts file and inside ProductComponent class do the following:
   * Declare a product Input() that is passed from parent component :
   * @Input() product!: Product;
   * Declare addToCartEvent event emmiter to emit event to parent component:
   * @Output() addToCartEvent = new EventEmitter<Product>();
   * Create a method called onAddToCartClicked() will emit a product to the addToCartEvent emmiter:
   * onAddToCartClicked(){  
      this.addToCartEvent.emit(this.product);  
     }
2. Open src/app/components/product/product.component.html file and do the following:
   * Replace <p>product works!</p> with the following:
   * <div>  
      <h2>{{ product.name }}</h2>  
      <p>Price: {{ product.price }}</p>  
      <button (click)="onAddToCartClicked()">Add to Cart</button>  
     </div>

### 3.3 Pass Data to and from child component

1. Open src/app/components/product-list/product-list.component.html file and do the following:
   * Pass product that we declared earlier to the ProductComponent:
   * <app-product [product]="product"></app-product>
   * Bind output property and function that were declared earlier:
   * <app-product [product]="product"(addToCartEvent)="addToCart($event)"></app-product>

### 3.4 Instpect Changes

1. Start Angular Development Server if not yet started:

* npx -p @angular/cli ng serve --host 0.0.0.0
* *Otherwise refresh the browser tab to see updated view.*

1. You should see the following getting rendered in your browser:

* 