

Explanation of Each File

Cart Class (App\Classe\Cart)

This class encapsulates the logic for managing a shopping cart. It interacts with the session to store and retrieve the cart's state, enabling persistent cart data across user interactions.

1. **__construct:**

- Receives a RequestStack object (Symfony service for handling HTTP requests).
- Enables access to the current HTTP request's session.

2. **add(\$product):**

- Adds a product to the cart.
- If the product already exists in the cart (identified by its id), it increments its quantity (qty).
- If the product is not in the cart, it initializes the entry with a quantity of 1.
- Updates the cart in the session.

3. **decrease(\$id):**

- Decreases the quantity of a product in the cart.
- If the quantity is greater than 1, it decrements the count.
- If the quantity reaches 1 and is decremented, the product is removed from the cart.
- Updates the session cart.

4. **getCart():**

- Returns the current cart from the session.

5. **remove():**

- Completely removes the cart from the session.

6. **fullQuantity():**

- Calculates the total number of items in the cart.
- Iterates over the cart items, summing up the qty values.

7. **getTotalWt():**

- Calculates the total price of the cart (with tax).
- Iterates through the cart items, summing up the product price (getPriceWt()) multiplied by its quantity (qty).

AppExtension Class (App\Twig\AppExtension)

This is a custom Twig extension to integrate specific features into the template rendering process.

1. **__construct:**

- Injects dependencies for CategoryRepository (to fetch categories) and the Cart class (to access cart logic).

2. **getFilters():**

- Registers a Twig filter price:
 - Converts a number to a string formatted as a price with two decimal places and a suffix (e.g., 12,34 Dt).

3. **formatPrice(\$number):**

- Implements the price filter.
- Formats a number as a string with two decimals, a comma as a decimal separator, and appends "Dt".

4. **getGlobals():**

- Exposes global variables for all Twig templates:
 - allCategories: List of categories fetched from the database via CategoryRepository.
 - fullCartQuantity: Total quantity of all products in the cart using the Cart service.

Twig Template

This template renders the shopping cart page. It uses data from the Cart and AppExtension classes.

1. **HTML Structure:**

- Contains two sections:
 - **Product List:** Displays cart items.
 - **Cart Summary:** Displays the total product quantity and total price.

2. **Dynamic Data Rendering:**

- **Product List** ({% for product in cart %}):
 - Iterates through the cart (cart is passed from the controller).
 - Displays product details (name, image, price, quantity).
 - Provides links to increase or decrease the product quantity.
- **Cart Summary:**
 - Displays total product quantity ({{fullCartQuantity}}).

- Displays total price using the price filter ({{totalwt|price}}).
-

Relation Between the Files

1. Cart and AppExtension:

- The Cart class provides the logic for managing the shopping cart.
- The AppExtension class integrates the Cart's data (e.g., fullQuantity()) into Twig templates via global variables like fullCartQuantity.

2. AppExtension and Twig Template:

- The AppExtension exposes:
 - allCategories: Provides a list of categories to the template.
 - fullCartQuantity: Supplies the total quantity for the cart summary.
 - price filter: Formats prices in the template.

3. Twig Template and Cart:

- The template displays the cart's data, which is fetched via the Cart service in the controller and passed to the template.
 - Interacts with the cart via links to increase, decrease, or clear items, triggering routes handled by the Cart class logic.
-

Overall Flow

1. Data Management:

- The Cart class manages all cart-related data and interactions.
- It stores and retrieves the cart from the session and handles cart operations.

2. Template Rendering:

- The AppExtension class bridges backend logic and Twig templates, making data like allCategories and fullCartQuantity globally available.

3. User Interaction:

- The template provides an interface for users to view, modify, and clear their cart.
- These actions trigger routes in the Symfony application, which then use the Cart class to perform the requested operations.