Le contrôleur CartController est responsable de la gestion du panier d'achat de l'utilisateur. Il utilise le service CartService pour effectuer les opérations liées au panier (afficher, ajouter, supprimer, vider, calculer le total, regrouper les articles).

**CartController :**

* **\_\_construct(CartService $cartService) :** Le constructeur injecte une instance du service CartService, rendant ses méthodes accessibles dans le contrôleur via la propriété $this->cartService.
* **showCart(?User $user): Response (Route : /order/cart, nom : cart) :**
  + Cette action affiche le contenu du panier. Elle récupère le panier, le total et le panier regroupé par exposition en utilisant les méthodes correspondantes du CartService.
  + Elle crée également une instance du formulaire UserIdentityCartFormType, qui est utilisé pour recueillir les informations d'identité de l'utilisateur avant la commande (si elles ne sont pas déjà renseignées).
  + Enfin, elle rend le template order/cart.html.twig en passant les informations du panier, le total et le formulaire. Le paramètre optionnel ?User $user permet d'accéder à l'utilisateur connecté, bien qu'il ne soit pas directement utilisé dans la logique d'affichage du panier dans ce code.
* **addTicketToCart(TicketRepository $ticketRepo, int $exhibitionId, int $ticketId, string $origin, ExhibitionShareRepository $exhibitionRepo): Response (Route : /ticket/{exhibitionId}/addTicketToCart/{ticketId}/{origin}, nom : addTicketToCart) :**
  + Cette action ajoute un ticket spécifique au panier. Elle reçoit l'ID de l'exposition (exhibitionId) et l'ID du ticket (ticketId) via les paramètres de la route, ainsi que l'origine de la requête (origin).
  + Elle utilise le TicketRepository pour récupérer l'entité Ticket correspondant à l'$ticketId et l'ExhibitionShareRepository pour récupérer l'entité ExhibitionShare correspondant à l'$exhibitionId.
  + Elle appelle la méthode addCart() du CartService pour ajouter le ticket à la session du panier, en fournissant l'entité Ticket et l'$exhibitionId.
  + Elle effectue une redirection en fonction de la valeur du paramètre $origin. Si $origin est 'listExhibit', l'utilisateur est redirigé vers la page de l'exposition. Sinon, il est redirigé vers la page du panier.
* **removeTicketFromCart(TicketRepository $ticketRepo, int $exhibitionId, int $ticketId, string $origin, ExhibitionShareRepository $exhibitionRepo) (Route : /ticket/{exhibitionId}/removeTicketFromCart/{ticketId}/{origin}, nom : removeTicketFromCart) :**
  + Cette action soustrait un ticket du panier. Elle reçoit les mêmes paramètres que addTicketToCart.
  + Elle récupère les entités Ticket et ExhibitionShare en utilisant leurs repositories.
  + Elle appelle la méthode removeCart() du CartService pour diminuer la quantité du ticket dans le panier pour l'exposition donnée.
  + Elle effectue une redirection vers la page de l'exposition si l'origine est 'listExhibit', ou vers la page du panier si l'origine est 'cart'.
* **deleteCartConfirm(): Response (Route : /order/deleteCartConfirm, nom : deleteCartConfirm) :**
  + Cette action affiche une page de confirmation avant de vider complètement le panier. Elle rend le template order/deleteCartConfirm.html.twig.
* **deleteCart(): Response (Route : /order/cart/delete, nom : deleteCart) :**
  + Cette action vide complètement le panier en appelant la méthode clearCart() du CartService.
  + Elle redirige ensuite l'utilisateur vers la page du panier.
* **removeProductToCart(string $id): Response (Route : /order/cart/remove/{id}, nom : removeProduct) :**
  + Cette action retire un produit spécifique du panier en utilisant sa clé unique ($id, qui correspond à $exhibitionId.'\_'.$ticketId).
  + Elle appelle la méthode removeProduct() du CartService pour supprimer l'élément du panier.
  + Elle redirige l'utilisateur vers la page du panier.
* **orderSuccess(): Response (Route : /orderSuccess, nom : orderSuccess) :**
  + Cette action affiche une page indiquant que la commande a été validée avec succès. Elle rend le template order/orderSuccess.html.twig.

**CartService :**

* **\_\_construct(RequestStack $requestStack, TicketRepository $ticketRepo, ExhibitionShareRepository $exhibitionShareRepo) :** Le constructeur injecte le RequestStack pour accéder à la session, le TicketRepository pour récupérer les informations des tickets, et l'ExhibitionShareRepository pour récupérer les informations des expositions.
* **getSession() :** Une méthode privée pour obtenir l'objet Session courant.
* **getCart(): array :** Récupère le contenu du panier depuis la session (clé 'cart'). Retourne un tableau vide si le panier n'existe pas.
* **setCart(array $cart): void :** Enregistre le tableau $cart dans la session sous la clé 'cart'.
* **addCart(Ticket $ticket, int $exhibitionId, int $qty = 1) :**
  + Ajoute un ticket au panier. Récupère le panier actuel, les détails du ticket via le TicketRepository, et l'exposition via l'ExhibitionShareRepository.
  + Utilise une clé unique ($exhibitionId.'\_'.$ticketId) pour identifier l'article dans le panier.
  + Si l'article existe déjà, incrémente la quantité. Sinon, ajoute un nouvel élément au panier avec les informations du ticket, de l'exposition, la quantité et le prix.
  + Calcule le total de la ligne pour cet article (totalLine).
  + Met à jour le panier dans la session et recalcule le total du panier.
* **removeCart(Ticket $ticket, int $exhibitionId, int $qty = 1) :**
  + Soustrait une quantité spécifiée d'un ticket du panier.
  + Récupère le panier et la clé de l'article.
  + Si l'article existe, décrémente la quantité. Si la quantité devient inférieure ou égale à zéro, l'article est supprimé du panier.
  + Met à jour le panier dans la session.
* **clearCart(): void :** Supprime complètement le panier de la session en utilisant $this->getSession()->remove('cart').
* **removeProduct(string $cartKey): void :** Supprime un article spécifique du panier en utilisant sa clé $cartKey. Met à jour le panier dans la session.
* **getTotal(): float :** Calcule le total du panier en parcourant tous les articles et en multipliant leur prix par leur quantité.
* **updateCartTotal($cart) :** Recalcule le total du panier en se basant sur le $totalLine de chaque article et met à jour la valeur 'cartTotal' dans la session.
* **cartCount(): int :** Retourne le nombre total d'articles dans le panier en additionnant les quantités de chaque article.
* **groupCartByExhibition(array $cart): array :** Regroupe les articles du panier par exposition. Le tableau résultant a pour clés les exhibitionId, et chaque entrée contient les informations de l'exposition et un tableau des tickets associés à cette exposition dans le panier (avec leur quantité et prix).