

Unit Test Plan

Module #:	Application #:
Tester: Esseme Audrey	Test Manager:Tientcheu Donald
Module Overview basketitem.js ,basket.js Describe the purpose of the module. basketitem.js : <ul style="list-style-type: none"> • Classe BasketItem représente un élément du panier contenant des informations sur un événement et le nombre de billets. • Méthode getPrice calcule le prix total pour les billets de l'événement. basket.js : <ul style="list-style-type: none"> • Plusieurs fonctions pour gérer les éléments du panier, calculer le total, afficher des publicités, rechercher des éléments, obtenir un élément spécifique, créer un nouvel élément et sérialiser les 	
Module Inputs basketitem.js : <p>BasketItem : Un objet événement et un nombre de billets.</p> basket.js : <p>calculateTotal : Liste des basketItems et éventuellement une remise.</p> <p>showAdverts : Objet utilisateur.</p> <p>searchBasket : Liste des basketItems et une requête de recherche.</p> <p>getBasketItem : Liste des basketItems et un objet événement.</p> <p>createBasketItem : Liste des basketItems, un objet événement et un nombre de billets requis.</p> <p>serializeBasketItemsToJson : Liste des basketItems.</p>	
Describe the inputs to the module.	

calculateTotal : Liste des basketItems et éventuellement une remise.

showAdverts : Objet utilisateur.

searchBasket : Liste des basketItems et une requête de recherche.

getBasketItem : Liste des basketItems et un objet événement.

createBasketItem : Liste des basketItems, un objet événement et un nombre de billets requis.

serializeBasketItemsToJson : Liste des basketItems

Module Outputs

-fichier basketItems : prix total des pièces

-fichier basket.js : boolean, liste basketItems, tableau JSON,

Describe the outputs from the module.

basketitem.js :

- Méthode getPrice : Retourne le prix total des billets.

basket.js :

- calculateTotal : Total des prix des éléments du panier, éventuellement ajusté par une remise.
- showAdverts : Booléen indiquant si des publicités doivent être affichées.
- searchBasket : Liste des basketItems correspondant à la requête de recherche.
- getBasketItem : L'élément du panier correspondant à l'événement, ou null.
- createBasketItem : Nouvel objet BasketItem ou null.
- serializeBasketItemsToJson : Tableau JSON des éléments du panier.

Logic Flow

basket.js :

- calculateTotal :
 - Vérifie le nombre d'éléments dans le panier.
 - Calcule le total des prix.
 - Applique une remise si présente.
- showAdverts :
 - Vérifie si l'utilisateur est premium.
- searchBasket :
 - Recherche les éléments du panier dont le nom de l'événement correspond à la requête.
- getBasketItem :
 - Parcourt les éléments du panier pour trouver une correspondance avec l'événement.
- createBasketItem :

- o Vérifie l'absence de l'élément dans le panier avant de le créer.
- `serializeBasketItemsToJson` :
 - o Convertit les éléments du panier en format JSON.

Test Data

List all test cases

Positive Test Cases

Test Case 1: Calcul du total sans remise

- Entrée : Liste d'éléments du panier.
- Test : Appeler `calculateTotal` sans remise.
- Résultat attendu : Total correct des prix.

Test Case 2: Calcul du total avec remise

- Entrée : Liste d'éléments du panier, remise.
- Test : Appeler `calculateTotal` avec remise.
- Résultat attendu : Total correct des prix avec remise appliquée.

Test Case 3: Affichage des publicités pour utilisateur non premium

- Entrée : Utilisateur non premium.
- Test : Appeler `showAdverts`.
- Résultat attendu : Retourne `true`.

Test Case 4: Recherche d'éléments dans le panier

- Entrée : Liste d'éléments du panier, requête de recherche.
- Test : Appeler `searchBasket`.
- Résultat attendu : Liste des éléments correspondant à la recherche.

Test Case 5: Obtenir un élément du panier

- Entrée : Liste d'éléments du panier, événement spécifique.
- Test : Appeler `getBasketItem`.
- Résultat attendu : L'élément du panier correspondant.

Test Case 6: Créer un nouvel élément dans le panier

- Entrée : Liste d'éléments du panier, événement, nombre de billets.
- Test : Appeler `createBasketItem`.
- Résultat attendu : Nouvel objet `BasketItem`.

Test Case 7: Sérialiser les éléments du panier en JSON

- Entrée : Liste d'éléments du panier.
- Test : Appeler `serializeBasketItemsToJson`.

- Résultat attendu : Tableau JSON des éléments du panier.

Number each test case. Indicate the test to be performed and expected outcome
Il y'aura 7 cas de test

Negative Test Cases

Test Case 8: Calcul du total avec une liste vide

- Entrée : Liste vide.
- Test : Appeler calculateTotal.
- Résultat attendu : Retourne 0.

Test Case 9: Affichage des publicités pour utilisateur premium

- Entrée : Utilisateur premium.
- Test : Appeler showAdverts.
- Résultat attendu : Retourne false.

Test Case 10: Recherche d'éléments avec requête inexistante

- Entrée : Liste d'éléments du panier, requête inexistante.
- Test : Appeler searchBasket.
- Résultat attendu : Liste vide.

Test Case 11: Obtenir un élément du panier inexistant

- Entrée : Liste d'éléments du panier, événement inexistant.
- Test : Appeler getBasketItem.
- Résultat attendu : Retourne null.

Test Case 12: Créer un élément du panier déjà existant

- Entrée : Liste d'éléments du panier, événement déjà existant.
- Test : Appeler createBasketItem.
- Résultat attendu : Retourne null.

Listin valid data selections

Interface Modules

Output data : serializeBasketItemsToJson produit des données JSON.

Data input : calculateTotal, showAdverts, searchBasket, getBasketItem, et createBasketItem prennent des données en entrée.

Internal program interface : Fonctions appelant des méthodes sur les objets BasketItem.

External program interface : Aucun mentionné explicitement.

Test Tools

Unit Testing Software : Utiliser un framework de test JavaScript comme Jest ou Mocha pour exécuter les tests unitaires.

Regression Testing Software : Les mêmes outils peuvent être utilisés pour les tests de régression. Les tests peuvent être automatisés et archivés dans un système de gestion de versions (comme Git) ou une plateforme d'intégration continue (CI) comme Jenkins.

Archive Plan

Specify location of archived data.

Define procedures required to obtain access to this data.

Updates

Unit Test Plan Updates : Le plan de test sera mis à jour à chaque modification du code source, ajout de nouvelles fonctionnalités ou correction de bugs. Les mises à jour seront documentées et versionnées dans le système de gestion de versions utilisé par le projet.

Identify how unit test plan will be updated.