

Unit Test Plan

Module #:	Application #:
Tester: Ndjock Jordan	Test Manager: Tientcheu Donald
Module Overview <u>promotions.js , exchange.js , discount.js , exchangeRateProvider.js</u> Describe the purpose of the module. promotion.js : calculatePercentageDiscount: Pourcentage de réduction, montant minimum à dépenser, prix actuel. calculateMoneyOff: Montant de la réduction fixe, montant minimum à dépenser, prix actuel. generateReferralCode: Identifiant utilisateur. applyDiscount: Code de réduction, montant total actuel. discount.js : <ul style="list-style-type: none"> Fonction pour calculer la réduction en fonction du type (par exemple, pourcentage, remise en argent). exchangeRateProvider.js : callExchangeRateProvider: Code de devise. <ul style="list-style-type: none"> Fonction pour appeler un service externe pour obtenir les taux de change. exchange.js : getExchangeRate: Code de devise. <ul style="list-style-type: none"> Fonction pour obtenir les taux de change. Discount.js : <ul style="list-style-type: none"> getDiscount: Code de réduction. Fonction pour récupérer les données de réduction en fonction d'un code de réduction. 	
Module inputs : Describe the inputs to the module. getExchangeRate: Code de devise.	

callExchangeRateProvider: Code de devise.

getDiscount: Code de réduction.

Module Outputs

- fichier promotions : float,boolean
- fichier discount.js : tableau JSON,
- fichier exchange.js : float
- fichier exchangeRateprovider : float

Describe the outputs from the module.

promotion.js :

- Appliquer une réduction : Prix ajusté après application de la réduction.
- Valider une réduction : Booléen indiquant la validité du code de réduction.

exchange.js :

- Taux de change : Le taux pour convertir une devise en une autre.

exchangeRateProvider.js :

- Taux de change : Le taux récupéré auprès d'un service externe.

discount.js :

- Montant de la réduction : La réduction calculée en fonction du code de réduction.

Logic Flow

basket.js :

promotion.js :

- Appliquer une réduction : Récupérer la réduction, valider, et appliquer au montant total.
- Valider une réduction : Vérifier le code de réduction selon des règles prédéfinies.

exchange.js :

- Obtenir le taux de change : Récupérer et retourner le taux de change pour une paire de devises donnée.

exchangeRateProvider.js :

- Appeler le fournisseur de taux de change : Récupérer le taux de change auprès d'un service externe.

discount.js :

- Calculer la réduction : Appliquer le type de réduction approprié au montant donné.

Test Data

List all test cases

Cas de Test 1 : Appliquer une réduction MONEYOFF

- **Entrée** : Code de réduction MONEYOFF valide, montant total actuel.
- **Test** : Appliquer la réduction.
- **Résultat Attendu** : Montant total réduit correct.

Cas de Test 2 : Appliquer une réduction PERCENTAGEOFF

- **Entrée** : Code de réduction PERCENTAGEOFF valide, montant total actuel.
- **Test** : Appliquer la réduction.
- **Résultat Attendu** : Montant total réduit correct.

Cas de Test 3 : Récupérer un taux de change valide

- **Entrée** : Code de devise valide.
- **Test** : Obtenir le taux de change.
- **Résultat Attendu** : Taux de change correct.

Cas de Test 4 : Valider un code de réduction

- **Entrée** : Code de réduction valide.
- **Test** : Valider la réduction.
- **Résultat Attendu** : Vrai pour un code de réduction valide.

Cas de Test 5 : Calculer le montant de la réduction

- **Entrée** : Code de réduction, montant total actuel.
- **Test** : Calculer la réduction.
- **Résultat Attendu** : Montant de la réduction correct

Number each test case. Indicate the test to be performed and expected outcome

Il y'aura 5 cas de test

Cas de Test 6 : Appliquer une réduction avec un code invalide

- **Entrée** : Code de réduction invalide, montant total actuel.
- **Test** : Appliquer la réduction.
- **Résultat Attendu** : Montant total actuel retourné inchangé.

Cas de Test 7 : Récupérer un taux de change pour une devise non supportée

- **Entrée** : Code de devise non supporté.

- **Test** : Obtenir le taux de change.
- **Résultat Attendu** : Erreur ou taux nul.

Cas de Test 8 : Valider un code de réduction invalide

- **Entrée** : Code de réduction invalide.
- **Test** : Valider la réduction.
- **Résultat Attendu** : Faux pour un code de réduction invalide.

Cas de Test 9 : Calculer la réduction sans code de réduction

- **Entrée** : Aucun code de réduction, montant total actuel.
- **Test** : Calculer la réduction.
- **Résultat Attendu** : Montant total actuel retourné inchangé.

Listin valid data selections

calculatePercentageDiscount

- Pourcentage : 20%, Dépense minimum : 100, Prix actuel : 150, Résultat attendu : 120
- Pourcentage : 20%, Dépense minimum : 100, Prix actuel : 80, Résultat attendu : 80
- Pourcentage : 0%, Dépense minimum : 100, Prix actuel : 150, Résultat attendu : 150
- Pourcentage : 100%, Dépense minimum : 100, Prix actuel : 150, Résultat attendu : 0
- Pourcentage : 20%, Dépense minimum : 100, Prix actuel : 100, Résultat attendu : 80

calculateMoneyOff

- Réduction : 20, Dépense minimum : 100, Prix actuel : 150, Résultat attendu : 130
- Réduction : 20, Dépense minimum : 100, Prix actuel : 80, Résultat attendu : 80
- Réduction : 0, Dépense minimum : 100, Prix actuel : 150, Résultat attendu : 150
- Réduction : 150, Dépense minimum : 100, Prix actuel : 150, Résultat attendu : 0
- Réduction : 20, Dépense minimum : 100, Prix actuel : 100, Résultat attendu : 80

generateReferralCode

- Identifiant utilisateur : 'user123'
- Résultat attendu : Code de parrainage qui commence par '#FRIEND-#', contient 'user123', et correspond au format regex `^#FRIEND-#\w{3}-#\w+$`

Interface Modules

Données de sortie : Les fonctions retournent des objets JSON ou des valeurs numériques ou des Promesses.

Données d'entrée : Les fonctions prennent divers paramètres, y compris des codes de réduction, des codes de devise et des montants.

Interface de programme interne : Les fonctions dans chaque module s'appellent entre elles.

Interface de programme externe : Les fonctions peuvent appeler des services externes (par exemple, fournisseur de taux de change).

Test Tools

Logiciel de Test Unitaire : Vitest.

Logiciel de Test de Régression : Utiliser les mêmes outils pour les tests de régression. Les tests doivent être automatisés et archivés dans un système de contrôle de version (par exemple, Git) ou une plateforme d'intégration continue (CI) comme Jenkins.

Archive Plan

Emplacement : Stocker les données et résultats de test dans un système de contrôle de version.

Procédures d'accès : Documenter les procédures pour accéder aux données de test archivées.

Specify location of archived data.

Define procedures required to obtain access to this data.

Updates

Mises à Jour du Plan de Test Unitaire : 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.