

Projet 5 : Construisez un site de e-commerce

Plan de tests unitaires

Ce plan de test doit couvrir 80 % de la base du code. Dans notre cas, elle contient 4 pages HTML et 4 pages en JavaScript.

Ce qu'il faut tester :

- ✓ La disponibilité de l'API.
- ✓ Le passage d'une page à une autre,
- ✓ Le choix de la couleur du teddy,
- ✓ La mise au panier d'un article,
- ✓ La suppression d'un article du panier,
- ✓ Le calcul du prix total,
- ✓ La validation des champs du formulaire,
- ✓ La validation du formulaire,
- ✓ La page de confirmation.

Test	Description du test	Lien vers le code
Vérifier que l'API est disponible	Dans les réponses de fetch, il y a un catch(err) avec un affichage de l'erreur dans la console ainsi qu'une alerte pour informer que l'API n'est pas disponible.	https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/js/index.js : lignes 13-15
Passage d'un page à une autre	Cliquez sur un des produits présentés sur la page index, vous devez arriver sur la page du produit.	https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/js/index.js : ligne 25

Choix de la couleur du teddy	Vérifiez que l'on puisse choisir la couleur du teddy	https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/js/produit.js : lignes 109-125
La mise au panier d'un article	Vérifier que lors du clic sur « ajouter au panier », l'article arriver bien dans le panier	https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/js/produit.js : lignes 64-70
La suppression d'un article du panier	Vérifier au clic sur « supprimer », que la ligne se supprime et que le prix total du panier est bien recalculé	https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/js/panier.js : lignes 78-93
Le calcul du prix du panier	Vérifier qu'à chaque ajout au panier, le prix se met bien à jour.	https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/js/panier.js lignes : 102-114
Validation des champs du formulaire	Vérifier que chaque champ correspond bien à ce qui est attendu (nom, prénom et ville : que des lettres avec tiret ou espace, email avec ses caractéristiques, adresse autorisant les numéros)	https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/js/panier.js lignes : 130 - 153
Validation du formulaire	Vérifier que tous les champs sont remplis	https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/js/panier.js lignes 137-141
Renvoi vers la page de confirmation	Affichage de la page de confirmation lors de l'envoi du formulaire avec récupération du prix total et de l'id de commande	https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/js/panier.js lignes 175-192 ET https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/confirmation.js lignes 2-4

Vérification des routes vers l'API à l'aide de Postman :

➔ Vérification de la route : <http://localhost:3000/api/teddies>

The screenshot shows the Postman interface with a collection named 'P5' containing 6 requests. The selected request is 'GET all teddies' with the URL 'http://localhost:3000/api/teddies/'. The response is a JSON array of two teddy objects. The status is 200 OK, time is 184 ms, and size is 1.88 KB.

```
GET http://localhost:3000/api/teddies/

Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Code

Query Params
KEY VALUE DESCRIPTION *** Bulk Edit
Key Value Description

Body
Cookies
Headers (10)
Test Results
Status: 200 OK Time: 184 ms Size: 1.88 KB Save Response

Pretty Raw Preview Visualize JSON

[
  {
    "id": "5be9c8541c9d44000665243",
    "name": "Norbert",
    "price": 2900,
    "imageUrl": "http://localhost:3000/images/teddy_1.jpg",
    "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."
  },
  {
    "id": "5beaa8b1c9d44000a57d94",
    "name": "Arnold",
    "price": 3900,
    "imageUrl": "http://localhost:3000/images/teddy_2.jpg",
    "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."
  }
]
```

➔ Vérification de la route : <http://localhost:3000/api/teddies/id>

The screenshot shows the Postman interface with a collection named 'P5' containing 6 requests. The selected request is 'GET one teddy' with the URL 'http://localhost:3000/api/teddies/5be9c8541c9d44000665243'. The response is a JSON object for a specific teddy. The status is 200 OK, time is 43 ms, and size is 745 B.

```
GET http://localhost:3000/api/teddies/5be9c8541c9d44000665243

Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Code

Query Params
KEY VALUE DESCRIPTION *** Bulk Edit
Key Value Description

Body
Cookies
Headers (10)
Test Results
Status: 200 OK Time: 43 ms Size: 745 B Save Response

Pretty Raw Preview Visualize JSON

{
  "colors": [
    "Tan",
    "Chocolate",
    "Black",
    "White"
  ],
  "id": "5be9c8541c9d44000665243",
  "name": "Norbert",
  "price": 2900,
  "imageUrl": "http://localhost:3000/images/teddy_1.jpg",
  "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."
}
```