

# 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
<b>Vérifier que l'API est disponible</b>	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.	<a href="https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/index.js">https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/index.js</a> : lignes 16-18
<b>Passage d'un page à une autre</b>	Cliquez sur un des produits présentés sur la page index, vous devez arriver sur la page du produit.	<a href="https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/index.js">https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/index.js</a> : ligne 28

<b>Choix de la couleur du teddy</b>	Vérifiez que l'on puisse choisir la couleur du teddy	<a href="https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/produit.js">https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/produit.js</a> : lignes 113-126
<b>La mise au panier d'un article</b>	Vérifier que lors du clic sur « ajouter au panier », l'article arriver bien dans le panier	<a href="https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/produit.js">https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/produit.js</a> lignes 68-73
<b>La suppression d'un article du panier</b>	Vérifier au clic sur « supprimer », que la ligne se supprime et que le prix total du panier est bien recalculé	<a href="https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/panier.js">https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/panier.js</a> : lignes 82-97
<b>Le calcul du prix du panier</b>	Vérifier qu'à chaque ajout au panier, le prix se met bien à jour.	<a href="https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/panier.js">https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/panier.js</a> lignes : 107-119
<b>Validation des champs du formulaire</b>	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)	<a href="https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/panier.js">https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/panier.js</a> lignes : 129 - 158
<b>Validation du formulaire</b>	Vérifier que tous les champs sont remplis	<a href="https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/panier.js">https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/panier.js</a> lignes 137-141
<b>Renvoi vers la page de confirmation</b>	Affichage de la page de confirmation lors de l'envoi du formulaire avec récupération du prix total et de l'id de commande	<a href="https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/panier.js">https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/panier.js</a> lignes 189-203 ET <a href="https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/confirmation.js">https://github.com/JessicaFournier/JessicaFournier_5_01102020/blob/master/JS/confirmation.js</a> lignes 2-6

### 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, each with fields: \_id, name, price, imageUrl, and description. The status is 200 OK, time is 184 ms, and size is 1.88 KB.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
7  {
8    "id": "5be9c8541c9d44000665243",
9    "name": "Norbert",
10   "price": 2900,
11   "imageUrl": "http://localhost:3000/images/teddy_1.jpg",
12   "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."
13 },
14 {
15   "colors": [
16     "Pale brown",
17     "Dark brown",
18     "White"
19   ],
20   "id": "5beaa8b1c9d44000a57d94",
21   "name": "Arnold",
22   "price": 3900,
23   "imageUrl": "http://localhost:3000/images/teddy_2.jpg",
24   "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."
25 },
26 ]
```

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

The screenshot shows the Postman interface with the same collection 'P5'. 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, including its colors array. The status is 200 OK, time is 43 ms, and size is 745 B.

```
1  {
2    "colors": [
3      "Tan",
4      "Chocolate",
5      "Black",
6      "White"
7    ],
8    "id": "5be9c8541c9d44000665243",
9    "name": "Norbert",
10   "price": 2900,
11   "imageUrl": "http://localhost:3000/images/teddy_1.jpg",
12   "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."
13 }
```