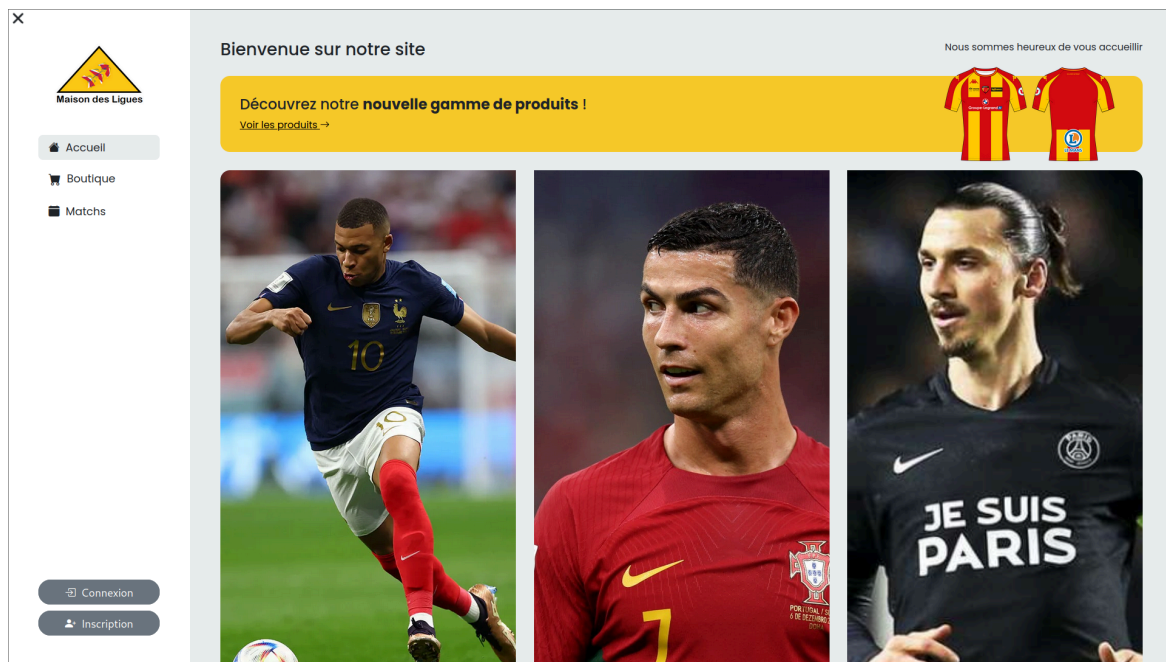


M2L: Tests unitaires

Pourquoi faire des tests unitaires ?

Les tests unitaires permettent de vérifier que chaque unité de code (généralement une fonction ou une méthode) fonctionne comme prévu. Cela aide à s'assurer que la logique interne de chaque partie du programme est correcte et performe les tâches pour lesquelles elle a été conçue.



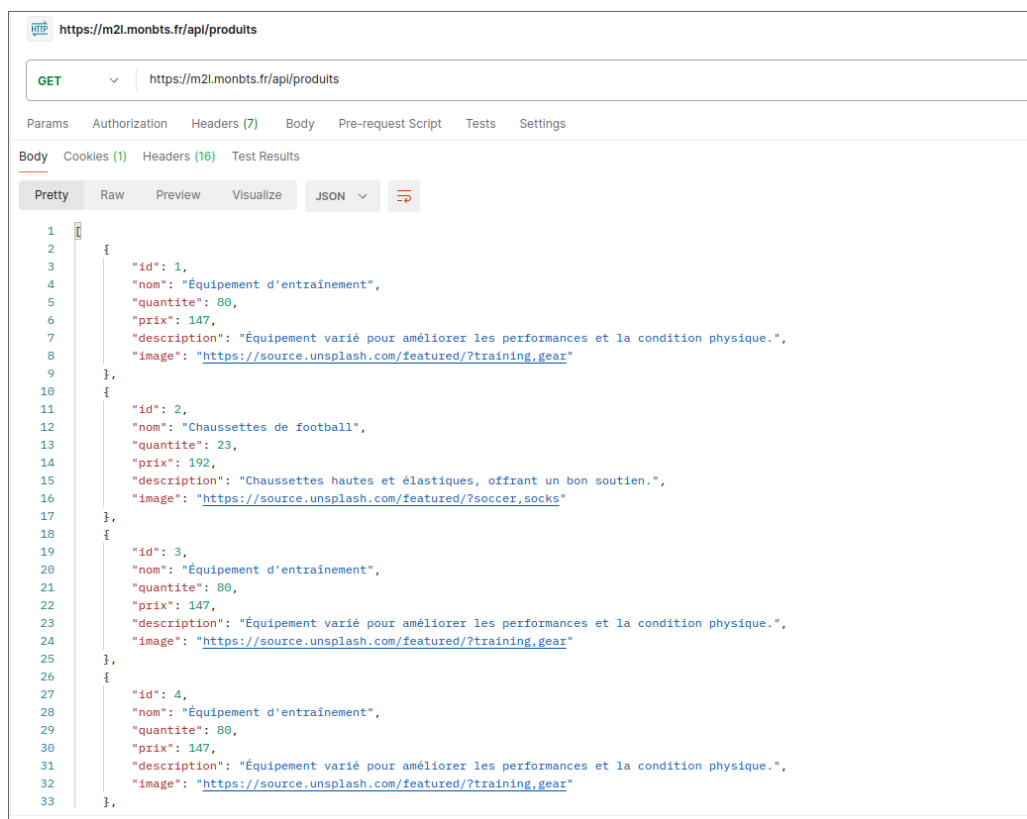
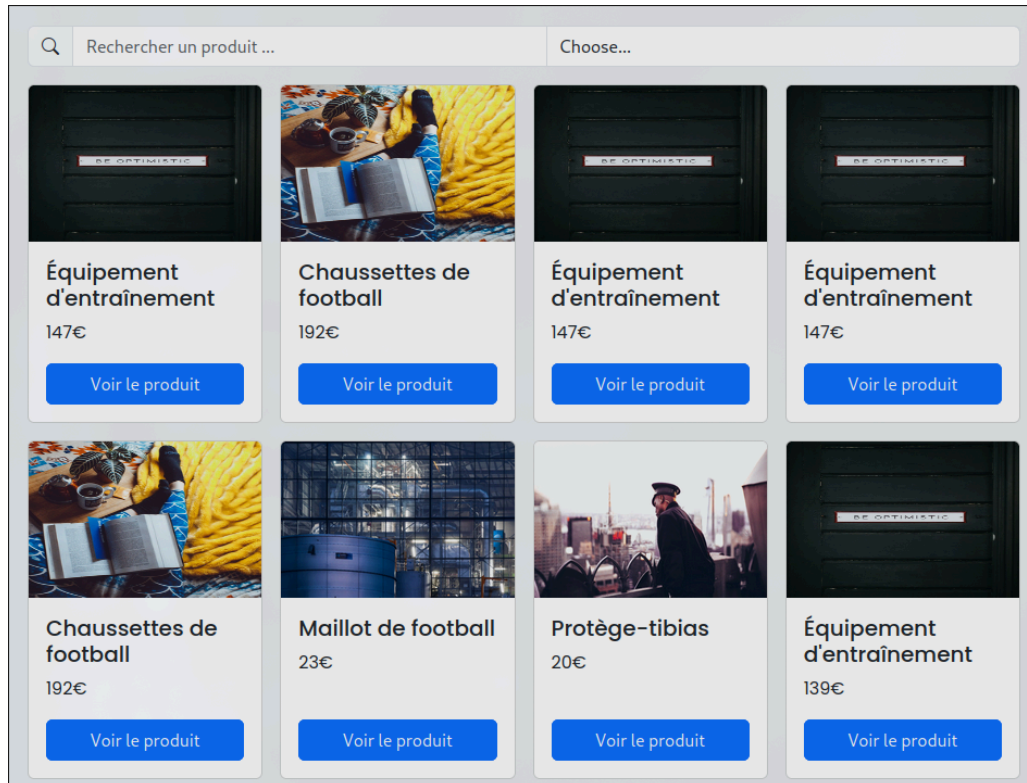
Quelles formes de tests adoptons-nous ?

Afin de vérifier que tout fonctionne, nous utiliserons 2 outils. Premièrement, Postman afin de vérifier que le serveur est prêt à recevoir des requêtes et à y répondre correctement. Deuxièmement, nous utilisons Jest, un outil qui permet de vérifier depuis le front-end que tout fonctionne.

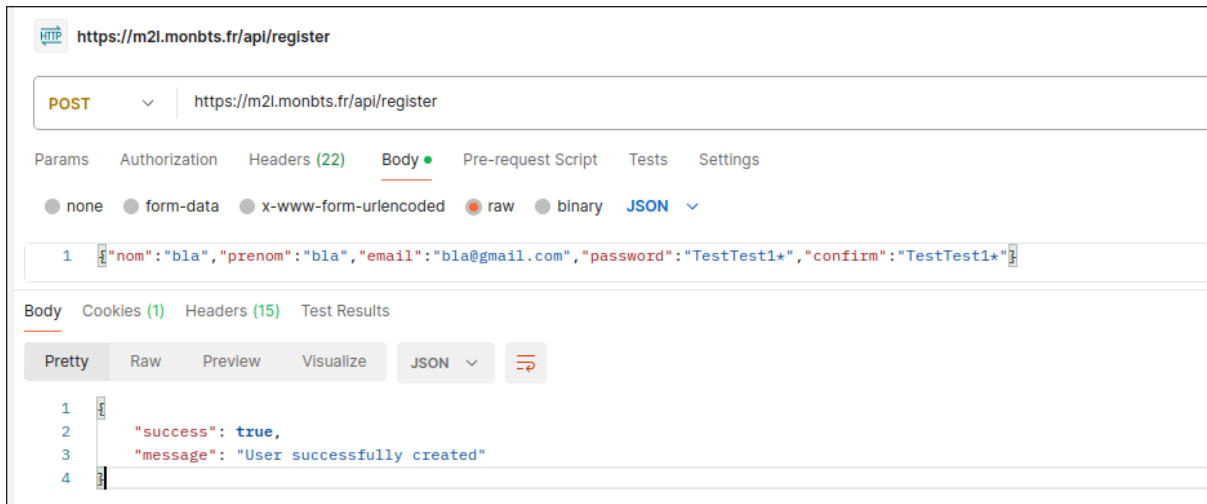


1. Tests Postman

a. Obtenir la liste des produits



b. Créer un nouveau compte utilisateur



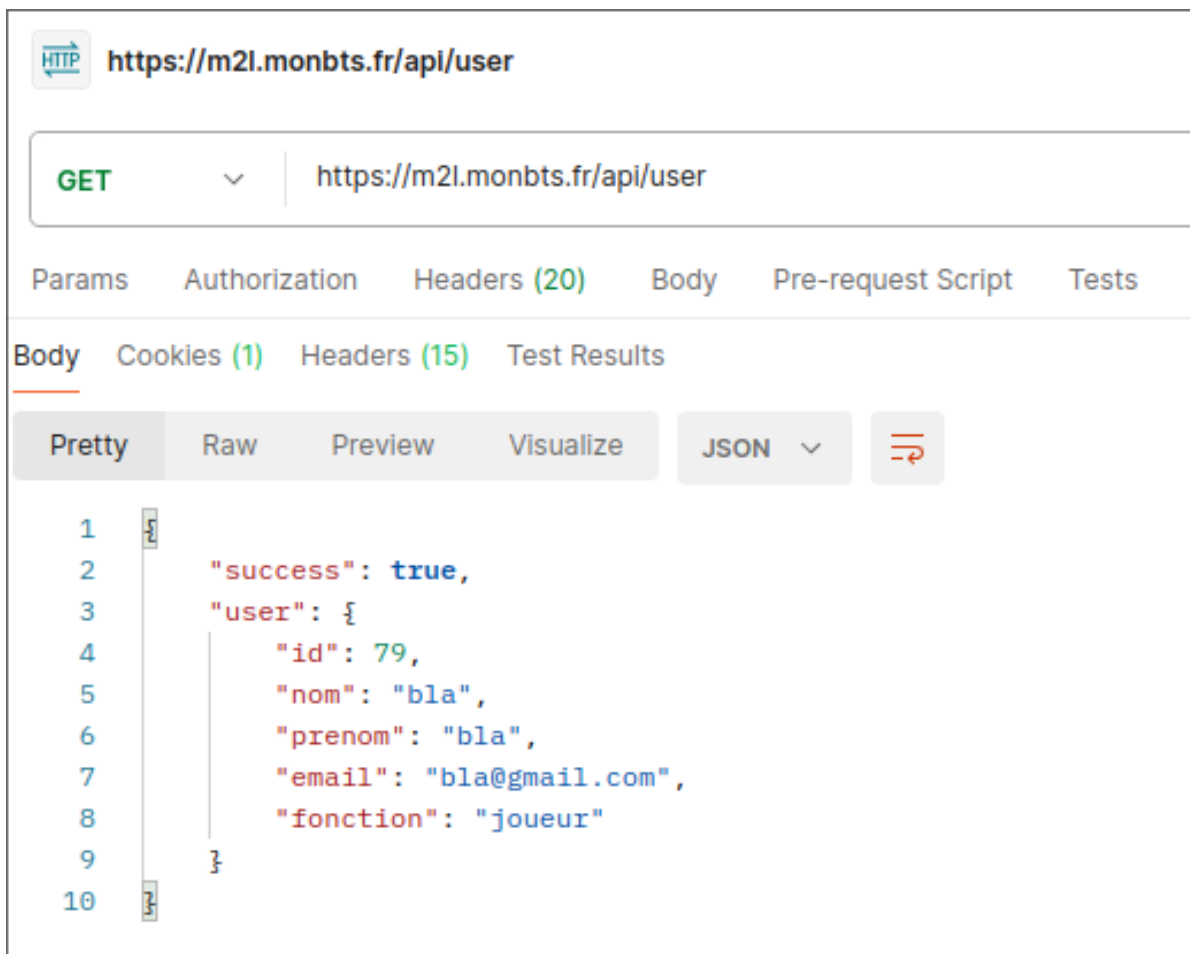
The screenshot shows a REST client interface with the URL `https://m2l.monbts.fr/api/register`. The method is **POST**. The request body is a JSON object: `{"nom": "bla", "prenom": "bla", "email": "bla@gmail.com", "password": "TestTest1*", "confirm": "TestTest1*"}`. The response body is also JSON: `{"success": true, "message": "User successfully created"}`.

```
POST https://m2l.monbts.fr/api/register

{"nom": "bla", "prenom": "bla", "email": "bla@gmail.com", "password": "TestTest1*", "confirm": "TestTest1*"}
```

```
Body
{"success": true,
 "message": "User successfully created"}
```

c. Obtenir les informations du compte utilisateur connecté



The screenshot shows a REST client interface with the URL `https://m2l.monbts.fr/api/user`. The method is **GET**. The response body is a JSON object containing user details: `{"success": true, "user": {"id": 79, "nom": "bla", "prenom": "bla", "email": "bla@gmail.com", "fonction": "joueur"}}`.



```
GET https://m2l.monbts.fr/api/user

{"success": true,
 "user": {
   "id": 79,
   "nom": "bla",
   "prenom": "bla",
   "email": "bla@gmail.com",
   "fonction": "joueur"
 }}
```

d. Obtenir la liste des commandes de l'utilisateur

Commandes

Commande effectuée le : 2024-04-09T08:56:28.000Z

Nom	Quantité	Prix
 Chaussettes de football	3	192 €
 Équipement d'entraînement	1	147 €
		Total 723 €

https://m2l.monbts.fr/api/commandes

GEThttps://m2l.monbts.fr/api/commandes

ParamsAuthorizationHeaders (19)BodyPre-request ScriptTestsSettings

BodyCookies (1)Headers (15)Test Results

PrettyRawPreviewVisualizeJSON

```
1  {
2    "success": true,
3    "commands": [
4      {
5        "id": "92aa0223-e61d-4f4e-a96e-de94ced531e7",
6        "id_utilisateur": 79,
7        "produits": [
8          {
9            "id": 2,
10           "quantity": 3,
11           "nom": "Chaussettes de football",
12           "prix": 192,
13           "image": "https://source.unsplash.com/featured/?soccer,socks",
14           "description": "Chaussettes hautes et élastiques, offrant un bon soutien."
15         },
16         {
17           "id": 4,
18           "quantity": 1,
19           "nom": "Équipement d'entraînement",
20           "prix": 147,
21           "image": "https://source.unsplash.com/featured/?training,gear",
22           "description": "Équipement varié pour améliorer les performances et la condition physique."
23         }
24       ],
25       "date": "2024-04-09T08:56:28.000Z"
26     }
27   ]
28 }
```

2. Tests Jest

a. Test pour vérifier qu'un utilisateur peut se créer un compte

```
describe("Register user", () => {
  test("Register user", async () => {
    const response = await fetch("http://localhost:3003/api/register", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify({
        nom: "test",
        prenom: "test",
        email: "test@gmail.com",
        password: "test1234",
        confirm: "test1234",
      }),
    });

    const data = await response.json();
    expect(data.success).toBe(true);
  });
});
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PASS src/App.test.js
✓ it renders a login button (36 ms)

A worker process has failed to exit gracefully and has been force exited. This is likely caused leaks. Active timers can also cause this, ensure that .unref() was called on them.

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 4.095 s, estimated 5 s
Ran all test suites related to changed files.