

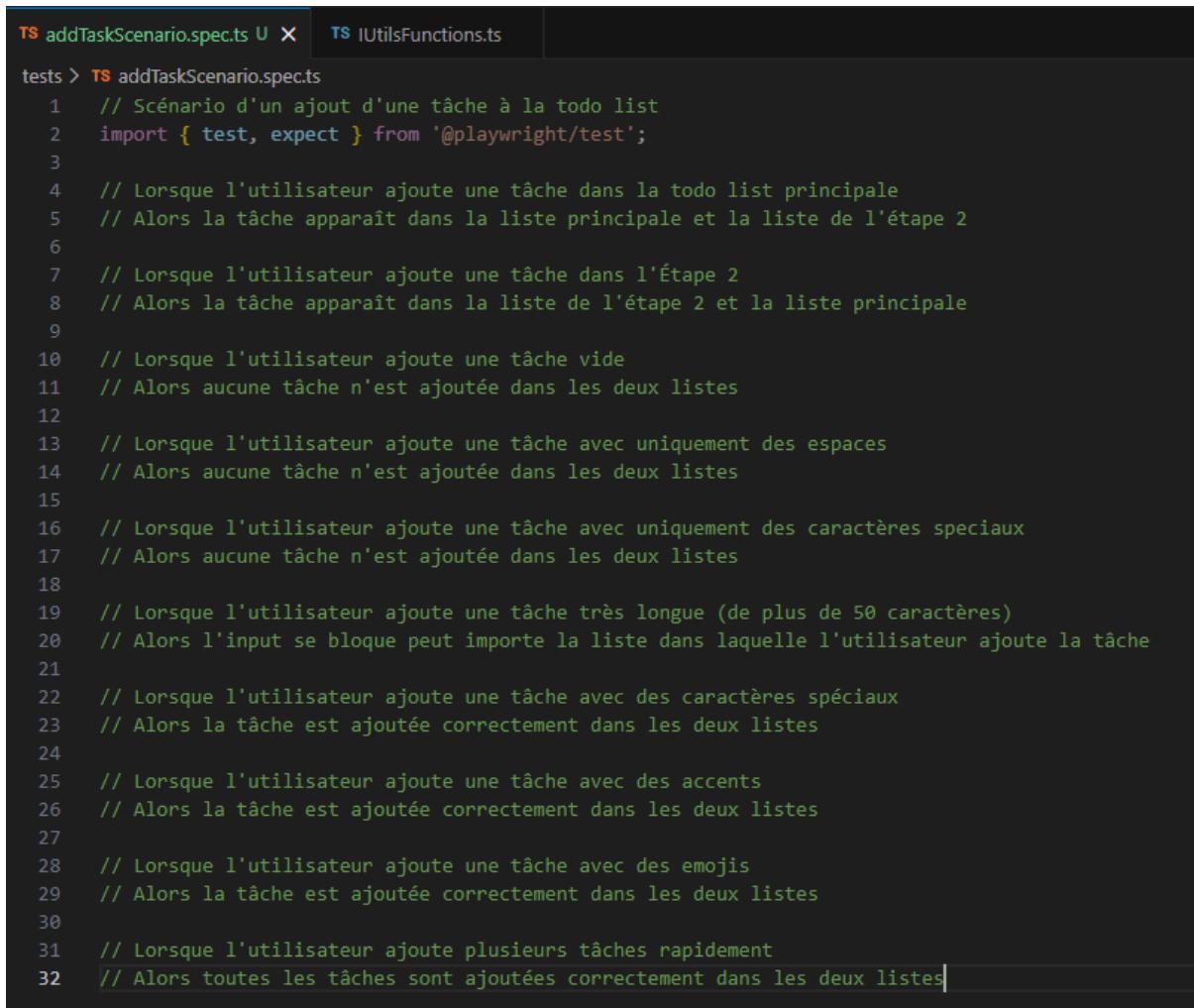
Prompts - TP IHM PlayWright

Pour les fichiers : **main.spec.ts**, **deleting.spec.ts**, **filtering.spec.ts** et **undo-redo.spec.ts** les prompts utilisés sont là :

<https://chatgpt.com/share/69271cf8-690c-8001-b084-a3e64c5878b9>.

Voici les prompts pour les fichiers : **addTaskScenario.spec.ts** et **modifyTaskScenario.spec.ts**.

Ci-dessous les commentaires pour le scénario d'ajout de tâche à la todo list.



```
TS addTaskScenario.spec.ts U X TS IUtilsFunctions.ts
tests > TS addTaskScenario.spec.ts
1 // Scénario d'un ajout d'une tâche à la todo list
2 import { test, expect } from '@playwright/test';
3
4 // Lorsque l'utilisateur ajoute une tâche dans la todo list principale
5 // Alors la tâche apparaît dans la liste principale et la liste de l'étape 2
6
7 // Lorsque l'utilisateur ajoute une tâche dans l'Étape 2
8 // Alors la tâche apparaît dans la liste de l'étape 2 et la liste principale
9
10 // Lorsque l'utilisateur ajoute une tâche vide
11 // Alors aucune tâche n'est ajoutée dans les deux listes
12
13 // Lorsque l'utilisateur ajoute une tâche avec uniquement des espaces
14 // Alors aucune tâche n'est ajoutée dans les deux listes
15
16 // Lorsque l'utilisateur ajoute une tâche avec uniquement des caractères spéciaux
17 // Alors aucune tâche n'est ajoutée dans les deux listes
18
19 // Lorsque l'utilisateur ajoute une tâche très longue (de plus de 50 caractères)
20 // Alors l'input se bloque peut importe la liste dans laquelle l'utilisateur ajoute la tâche
21
22 // Lorsque l'utilisateur ajoute une tâche avec des caractères spéciaux
23 // Alors la tâche est ajoutée correctement dans les deux listes
24
25 // Lorsque l'utilisateur ajoute une tâche avec des accents
26 // Alors la tâche est ajoutée correctement dans les deux listes
27
28 // Lorsque l'utilisateur ajoute une tâche avec des emojis
29 // Alors la tâche est ajoutée correctement dans les deux listes
30
31 // Lorsque l'utilisateur ajoute plusieurs tâches rapidement
32 // Alors toutes les tâches sont ajoutées correctement dans les deux listes|
```

Prompt 1 :

Je suis entrain de tester une page html qui sert à gérer sa todo list.

```
Voici comment est codé la page html : "<html lang="en"
data-critters-container=""><head>
<meta charset="utf-8">
<title>L3m20232024AngularTodolist</title>
...
<script src="runtime.8falb342189f1975.js"
type="module"></script><script src="polyfills.da1168390945d94d.js"
type="module"></script><script src="main.982bce64065a2736.js"
type="module"></script>

</body></html>"
```

Là, je test un scénario d'ajout d'une tâche dans la liste. J'ai détaillé tous les cas possibles dans le fichier "addTaskScenario.spec.ts". Code-moi les tests appropriés aux description en TypeScript en te servant du fichier "IUtilsFunctions.ts" qui contient plusieurs fonctions.

Fin Prompt 1.

Prompt 2 :

Je viens de rajouter un test dans le fichier "addTaskScenario.spec.ts" : "// Lorsque l'utilisateur ajoute une tâche vérifier que la phrase "nombreTache restantes" est mise à jour correctement". Comme pour les autres: implémente son test.

Fin Prompt 2.

Prompt 3 :

Je viens de rajouter deux tests dans le fichier "addTaskScenario.spec.ts" : "// Lorsque l'utilisateur plusieurs tâches vérifier que la phrase "nombreTache restantes" est mise à jour correctement" et " // Lorsque l'utilisateur ajoute une tâche Alors la tâche est ajoutée en haut de la liste". Comme pour les autres: implémente son test.

Fin Prompt 3.

Prompt 4 (pour le fichier modifyTaskScenario.spec.ts):

En reprenant le même contexte. Implémente moi les tests qui correspondent à chacun de mes commentaires en t'aident des fonctions IUtilisFunctions.ts.

Fin Prompt 4.