Todolist

|  |  |  |
| --- | --- | --- |
| Todolist | Spécification technico-fonctionnelle détaillée | V 1.0 |
| S.L |
| Todolist | 04/05/2019 |

1. Description de l’usage (non technique) de l’application
2. Description de l’usage (technique) de l’application

# Description de l’utilisation (non technique) de l’application

## Ajout d’une tâche

L’application to-do list permet de crée une liste à puce des tâches que vous avez à faire. En cliquant simplement dans le champ principal, il vous suffira de décrire la tâche et d’appuyer sur la touche « Enter » et elle s’ajoutera à la liste.

Une fois cela effectué de nouveaux éléments apparaîtront :

* La tâches que vous venez de crée et son statut
* Une puce par tâche pour connaitre leur statut
* Le nombre de tâche restante
* Le filtre actif parmi les trois disponible (All/Active/Completed)

## Sauvegarde des tâches

L’application sauvegarde les tâches que vous entrez en créant un fichier JSON dans le local storage de votre navigateur.

Ce fichier peut être supprimé manuellement par l’utilisateur.

## Modification et interaction d’une tâche

En double cliquant sur une tache vous pourrez modifier son contenu sans avoir à la supprimer et la recrée.

De plus en survolant la tâche vous verrez apparaitre une croix qui vous permettra de supprimer cette dernière.

En cliquant sur la puce à gauche de votre tâche vous validerez celle-ci et sera donc considérer comme finie (complete) et se verra griser et barré et un nouveaux bouton apparaitra « Clear completed ».

## Interaction avec l’application

L’application propose plusieurs interactions avec elle en dehors des tâches.

Premièrement elle dispose de trois filtres cliquables :

* All affichera toute les tâche en cours et finie
* Active affichera seulement les tâches en cours
* Completed affichera seulement les tâches finie

Si vous avez coché toute vos tâches comme étant finie alors la petite flèche à gauche du champ « What’s to be done » sera higlighter.

Enfin en bas à gauche du cadre de l’application le nombre de tâche restante sera mis à jour suivant le nombre de tâche en cours.

## Suppression d’une ou plusieurs tâches

Une fois votre tâche effectuer et valider vous pourrez la supprimer en cliquant simplement sur la croix qui apparait au survol de cette dernière (cf Modification et interaction d’une tâche).

Si au moins une tâche possède le statut de « complete » (cf Modification et interaction d’une tâche) alors le bouton « Clear completed » apparaitra et sera cliquable. Il vous permet de supprimer d’un coup toute les tâches que vous avez indiquées comme finie.

Si toutes les tâches ont été supprimées alors l’application reviendra à son format de départ avec le seul champ « What’s to be done » de visible.

# Description de l’usage (technique) de l’application

## Présentation du projet

L’application fonctionne sur un motif Modèle-vue-contrôleur ou MVC (JavascriptMVC).

C’est un motif d'architecture logicielle destiné aux interfaces graphiques très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

Un modèle (Model) contient les données à afficher.

Une vue (View) contient la présentation de l'interface graphique.

Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

## Téléchargement et installation du projet

Vous pouvez télécharger le projet sur ce [repository](https://github.com/SIGIERLuc/todo-list) .

À la racine du projet vous trouverez le fichier package.json qui vous renseignera sur les modules dont le projet a besoin pour fonctionner.

Une fois le projet téléchargé et dézipper il faut installer les modules node :

* Rendez-vous à cet endroit dans le projet « Path\todo-list-project\todo-list-app\node\_modules »
* Ouvrez la console et rentrer la commande suivante « npm install »
* Après quelque seconde les modules nécessaires (Path\todo-list-project\todo-list-app\ package.json) seront installés.
* Le projet ainsi que les tests sont maintenant fonctionnels.

N’oubliez pas qu’à la racine un fichier .gitignore est présent et qu’il faudra réinstaller les modules si vous télécharger sur une autre machine.

## Les différent JS et leurs rôles

Le projet comporte 7 fichier JS (Path\todo-list-project\todo-list-app\js) indispensable à son fonctionnement (celui présent dans la partie test n’est pas indispensable pour le fonctionnement du projet). Chaque JS à un rôle, le code étant suffisamment commenter, nous n’allons pas ici décrire chaque fonction présente mais faire un résumer de chaque rôle qu’ils remplissent.

### app.js

Va lancer le projet en donnant un nom au fichier JSON qui servira de base de donné et initialiser tout le projet.

### store.js

Store va permettre de crée un fichier JSON en local sur le navigateur de l’utilisateur et d’interagir avec ce dernier.

Les fonctions crée ici sont utiliser par model.js afin de crée, mettre à jour, supprimer des todos sur le JSON.

### controller.js

Permet d’interagir avec les actions de l’utilisateur et de modifier les données présente dans le model et la view.

### model.js

Contient les données et la logique de fonctionnement de l’application ce JS va permettre d’interagir principalement avec store.js pour pouvoir ajouter, modifier, supprimer des données.

### view.js

Contient la présentation de l’interface graphique elle va donc traduire les fonctions de model.js en élément du DOM visible par l’utilisateur.

### helpers.js

Va créer des querySelector et des EventListener afin qu’ils soient réutilisé par view.js pour pouvoir interagir avec le DOM.

### template.js

Crée des templates qui vont être réutilisé dans le code comme l’élément HTML d’une todo ou alors sa construction dans la base de donnée

## Les tests

Ici les tests sont réalisés avec le module Jasmine installé un peu plus tôt (cf. [Téléchargement et installation du projet](#_Téléchargement_et_installation) ) et permette de tester différent cas d’utilisation afin de s’assurer que le code fonctionne correctement.

Les tests en eux même sont écris en JavaScript dans le fichier ControllerSpec.js et le fichier SpecRunner.html permet de voir si les attentes de chaque test sont juste.

Pour tout ce qui se rapporte au fonctionnement de Jasmine je vous invite à vous documenter sur [la page GitHub de Jasmine](https://jasmine.github.io/index.html)