

TD Todo List - Partie 02

React



07 : Variable d'état.....	2
08 : Callback.....	3
09 : Rendu conditionnel.....	6
10 : Mise à jour du statut.....	7
Pour aller plus loin.....	8

07 : Variable d'état

- Dans `./src/App.tsx`, importez le **hook** `useState` de React.
- Employez le hook `useState`, pour convertir **tasks** en variable d'état et définissez également un **mutateur** nommé `setTasks`.

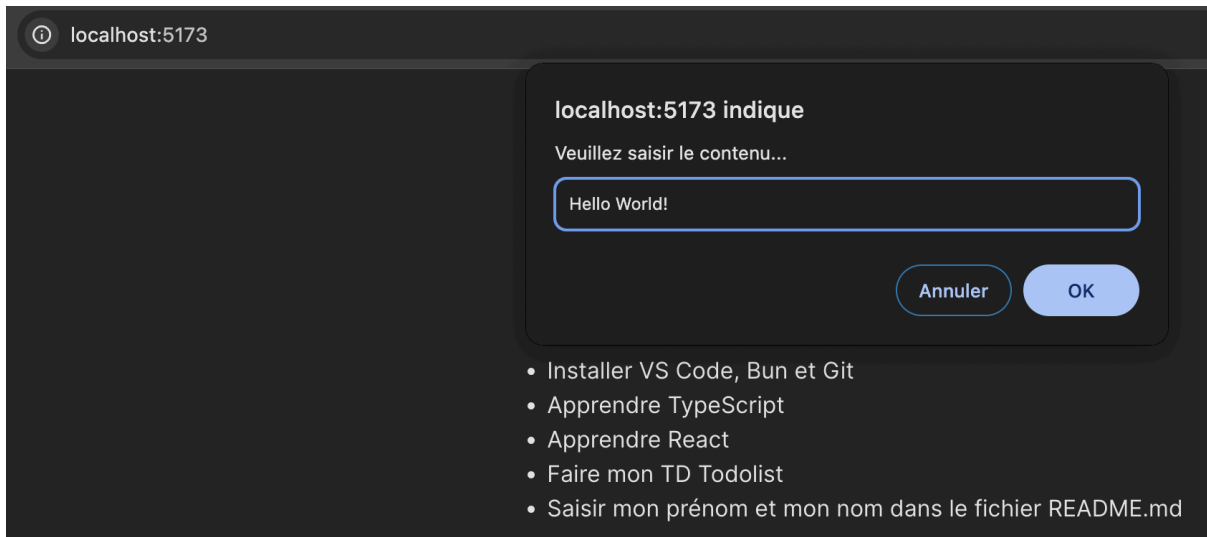
A noter : en *TypeScript*, la déclaration d'une variable d'état dans React avec `useState` doit être typée (cf. <https://www.robinwieruch.de/typescript-react-usestate/>).

Etape 08 page suivante

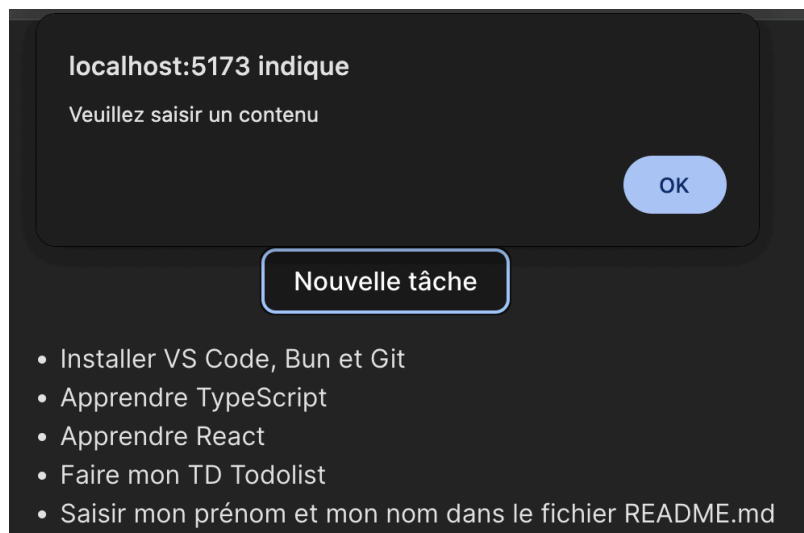
08 : Callback

- Dans le composant **App**, créez une fonction **addNewTask** avec un paramètre **content** de type chaîne de caractères. Cette fonction devra permettre d'instancier un nouvel objet de type **Task**, de l'ajouter à la liste **tasks** et de mettre à jour l'état de l'application (ce qui générera un rafraîchissement partiel de l'écran).
 - la valeur de l'attribut **id** sera obtenu dynamiquement à l'aide du module **uuid**,
 - la valeur de l'attribut **content** correspondra à la valeur passée en argument du paramètre **content** de la fonction **addNewTask**,
 - la valeur de l'attribut **status** sera par défaut **"todo"**.
 - faites en sorte que le nouvel objet de type **Task** soit ajouté en première position de la liste.
- Ajoutez une balise HTML permettant de représenter un bouton avec la mention *"Nouvelle tâche"*.
- Créez une fonction nommée **handleClick** qui fera office de fonction *callback* exécutée au clic sur le bouton.
- Associez un écouteur d'événement permettant de détecter les clics sur le bouton et exécutant la fonction *callback* **handleClick**.
- Au sein de la fonction **handleClick** affichez un *prompt* proposant à l'utilisateur de saisir le contenu d'une nouvelle tâche.
- Si la valeur saisie, retournée par le *prompt* n'est pas valide (*null*, *undefined*, chaîne vide...), affichez une alerte indiquant à l'utilisateur de saisir un contenu.
- Dans le cas contraire, appelez la fonction **addNewTask** en passant en argument la valeur saisie par l'utilisateur.

Cet ajout doit rafraîchir automatiquement l'écran via la mise à jour d'état. Vérifiez que la liste *HTML* est bien mise à jour après l'appel de la fonction **addNewTask** (uniquement lorsque la valeur saisie par l'utilisateur est valide).



Prompt affiché au clic sur le bouton "Nouvelle tâche"



Message d'alerte affiché en cas de saisie invalide

Todo List

Nombre de tâches : 6

Nouvelle tâche

- Installer VS Code, Bun et Git
- Apprendre TypeScript
- Apprendre React
- Faire mon TD Todolist
- Saisir mon prénom et mon nom dans le fichier README.md
- Hello World!

Nouvelle tâche ajoutée à la liste

Etape 09 page suivante

09 : Rendu conditionnel

- Employez, au choix, un **Emoji**, une image ou un style CSS permettant d'illustrer graphiquement la valeur de l'attribut **status** de chaque item contenu dans la liste **tasks**.

todo = 🕒
doing = 🕒
done = ✅

- Le rendu conditionnel peut être effectué directement dans le template ou délégué à une fonction retournant une valeur de type **JSX.Element** d'après la valeur de l'attribut **status** fourni en argument.

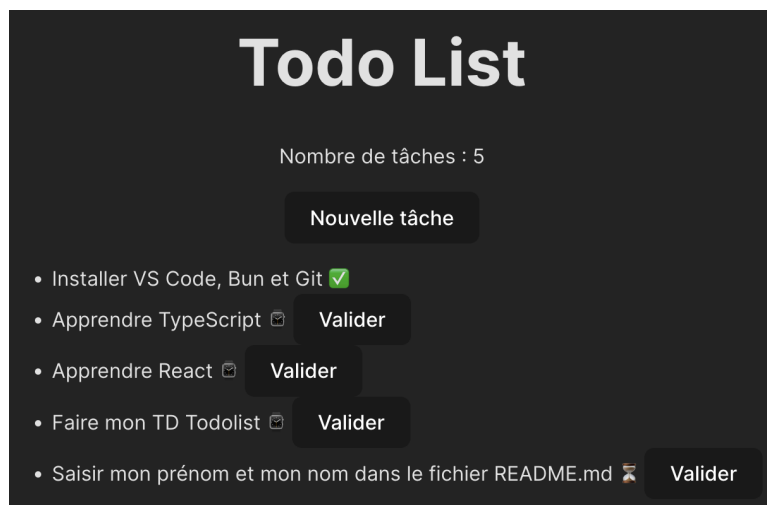


Statut illustré par un Emoji.

Etape 10 page suivante

10 : Mise à jour du statut

- Ajoutez un bouton avec la mention **"Valider"** pour chaque item de la liste, dont la valeur de l'attribut **status** est différent de **"done"**.
- Au clic, sur le bouton **"Valider"**, passez la valeur de l'attribut **status** de l'objet **Task** concerné à **"done"** et la valeur de l'attribut **completedAt** à la date courante. Cette modification doit rafraîchir automatiquement l'écran via la mise à jour d'état.



Affichage d'un bouton **Valider** permettant de passer à **"done"** la valeur de l'attribut **"status"** des tâches non complétées

Fin

Pour aller plus loin

- Mettre en place un moyen de modifier le contenu d'un item de type **Task** (ex: formulaire, prompt...),
- Mettre en place une solution de filtre par valeur de l'attribut **status**.
- Ajouter un élément permettant d'indiquer qu'une tâche est en cours de réalisation (**status = doing**).
- Connexion à une API distante pour la lecture et l'écriture de données synchronisées avec une source distante.