

Smart contract

ToDoList

```
createTodo(: string)
```

Crée un todo

```
done(: uint)
```

Marque un todo comme fait. Emet un event.

```
pause ()
```

Met le contrat en pause. Seul le contract owner peut faire ça.

```
unpause()
```

Retire la pause du contrat. Seul le contract owner peut faire ça.

Autres fonctionnalités possibles pour améliorer :

- Faire en sorte qu'un todo enregistre l'adresse de celui qui le crée et que seule cette adresse puisse utiliser done() sur ce todo
- Rectifier (update) le descriptif d'un todo
- Ajouter un timestamp à la complétion d'un todo
- Ajouter des events pour chaque fonction
- Utiliser des hash pour les descriptions et sauver la description textuelle dans une DB externe
- ...

WebServer

Une API REST en NodeJS (express ou nestJS) qui expose les fonctions principales du contrat.

Pour la signature, deux possibilités :

- Soit l'API gère les clés des utilisateurs. Elle doit alors gérer l'authentification.
- Soit l'API reçoit les txs présignées. Le browser devra implémenter un client web3 (ethers) pour créer les txs, les signer et les envoyer telles quelles vers l'API.

Si option 1, l'API possède en outre une porte permettant de générer un compte Ethereum et le sauver en DB. Alternative : cette création de compte peut avoir lieu au moment de la création de compte utilisateur de l'API.

Front-end

Affiche les fonctions. Affiche la page d'authentification (si implémentée).

Si authentification pas nécessaire, alors :

- le mnemonic est demandé aux utilisateurs quand ils appellent une fonction entraînant une signature.
- Le browser implémente la création de compte Ethereum