

TD 04 - Todo List

Navigation - Interaction avec le modèle de données - Gestion d'état

Objectifs techniques

- Dans le composant `TodoDetails`, ajoutez :
 - un bouton permettant de modifier la Todo affichée (si `"done" = false`),
 - un bouton permettant de supprimer la Todo affichée (si `done = true`),
- Si l'utilisateur clique sur le bouton de suppression, la Todo concernée doit-être supprimée et les données du composant `TodoMaster` mises à jour. De plus, le composant `TodoDetails` doit être masqué,
- Mettez en place un composant permettant de demander à l'utilisateur la confirmation de la suppression de la Todo concernée (ex: <https://flutter.dev/docs/cookbook/design/snackbars>),
- Lorsque l'utilisateur clique sur le bouton de modification, un nouvel écran doit être affiché. Aussi, créez une nouvelle page **OneTodoPage**,
- Mettez en place un système de navigation composé de 2 routes "*nommées*" (si Flutter utilisé, cf. <https://flutter.dev/docs/cookbook/navigation/named-routes>), par exemple :
 - `"/allTodosPage"` associée au composant `AllTodosPage` (liste des Todos),
 - `"/oneTodoPage"` associée au composant `OneTodoPage` (affichage d'une Todo),
- L'application doit communiquer la Todo concernée (cliquée depuis le composant `TodoDetails`) lors du passage de l'écran `AllTodosPage` à l'écran `OneTodoPage` (si Flutter utilisé, cf. <https://flutter.dev/docs/cookbook/navigation/passing-data>),
- Dans la page `OneTodoPage`, intégrez un nouveau composant nommé **TodoForm**, permettant de modifier les valeurs de la Todo sélectionnée. Utilisez des éléments UI de type "*formulaire*" afin de modifier les valeurs de la Todo passée en paramètre (cf.

<https://flutter.dev/docs/cookbook/forms>),

- Lorsque l'utilisateur clique sur le bouton de validation du formulaire, vérifiez que les données ont bien été saisies. Si ce n'est pas le cas affichez un message informatif à destination de l'utilisateur. Sinon mettez à jour les données et ré-affichez l'écran AllTodosPage,
- Enfin, affichez une confirmation de mise à jour (ex: une notification),

Objectifs pédagogiques

- programmation événementielle,
- circulation de données à travers différents composants imbriqués,
- implémentation du design pattern Master / Details,
- navigation à travers différents écrans,
- communication à travers différents écrans,
- interaction avec le modèle de données,
- gestion d'état Stateful / Stateless,