



DOCUMENTATION TODOLIST

Rédigée par :

NGUYEN Kévin

Tables des matières :

1. Qu'est-ce qu'une ToDoList ? – p.2
2. Présentation de l'équipe ? p-3
3. Comment fonctionne le ToDoList ? p-4
4. Comment sont gérées les données ? p-5
5. Comment fonctionne l'application ? p-6

Qu'est-ce qu'une ToDoList?



Une ***todo list***, ou **liste de tâches**, est un procédé qui se veut simple et efficace pour gérer les tâches d'un projet.

Ces tâches peuvent être indépendantes ou devoir, au contraire, être accomplies dans un certain ordre. Un chef de projet informatique y notera par exemple les bugs à corriger et les dates de début et de fin de problème. Chaque tâche peut à son tour nécessiter sa propre liste de (sous-)tâches, et ainsi de suite. Des post-it où l'on note des courses à faire constituent un exemple rudimentaire de *todo list*.

Présentation de l'équipe

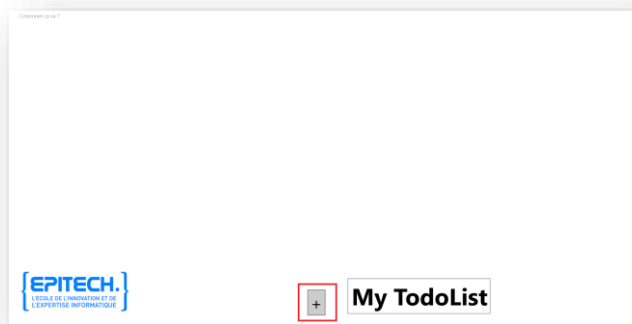
NGUYEN Kévin



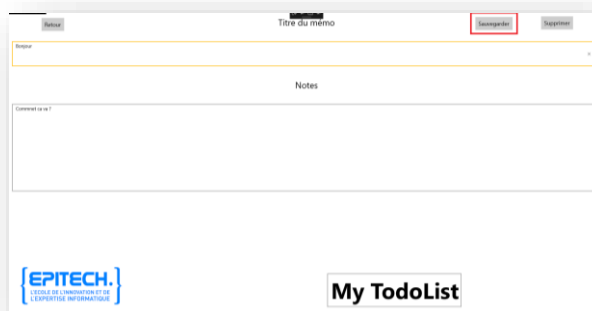
COMMENT FONCTIONNE LE TODOLIST ?

Le TodoList fonctionne de manière très simple, vous avez la possibilité de :

- Créer une tâche : Définissez un titre et le contenu de votre tâche 😊



- Modifier/supprimer une tâche :



- Marquer une tâche réalisée ou non.

COMMENT SONT GERES LES DONNEES ?

```
private void deleteButtonClick(object sender, RoutedEventArgs e)
{
    using (var db = DbConnection) {
        db.Execute("DELETE FROM Task WHERE Id = ?", task.Id);
    }
    this.Frame.Navigate(typeof(ListPage), null);
}
```

Les données sont très sécurisées, en effet l'application gère les données des tâches l'aide de Sqlite. Les requêtes vers la base de données se font logiquement par des requêtes SQL. Les données sont enregistrées vers un fichier nommé « db.sqlite ». Dans notre cas nous avons simplement créer une colonne « Tasks ».

```
public ListPage() {
    this.InitializeComponent();

    using (var db = DbConnection) {
        db.CreateTable<Task>();
        db.GetMapping(typeof(Task));
    }
    updateList();
}
```

COMMENT FONCTIONNE L'APPLICATION ?

L'application fonctionne de manière très simple, une liste de tâches est gérée durant chacune des actions proposées. Ajouter/Supprimer ajoute et supprime la tâche (lié à un ID) dans la liste. Une tâche contient aussi un titre, une description, et un statut « done » pour savoir si la tâche a été réalisée ou non.

La création de boutons et de textbox sont très simples, elles sont affichées dans l'onglet « boîte à outil » de Visual Studio.

La navigation entre les « pages » (appelés views) sont également très simples :

```
this.Frame.Navigate(typeof(ListPage), null);
```

: permet de revenir à l'accueil

```
private void updateTask(object sender, ItemClickEventArgs e) {  
    Task task = (Task)e.ClickedItem;  
    this.Frame.Navigate(typeof(UpdatePage), task);  
}  
  
private void createTask(object sender, RoutedEventArgs e) {  
    this.Frame.Navigate(typeof(CreatePage), null);  
}
```

On peut voir que les deux commandes servent respectivement à naviguer vers :

- CreatePage
- UpdatePage