

Parcours Développeur d'application iOS
Projet 9 – Baluchon
Bonus

Pour le projet 9 du parcours Développeur d'application iOS, il est demandé de créer une application de voyage nommé « Baluchon ». Cette application possède trois pages qui permettent d'obtenir le taux de change, des traductions ainsi que la météo.

Pour ce projet, j'ai choisi de rajouter une fonctionnalité sur la page météo. Actuellement, on obtient la météo de New-York ainsi que de Lyon grâce à l'API Yahoo Weather. Un bouton « + » a été rajouté dans la barre de navigation afin d'obtenir la météo de différentes villes.

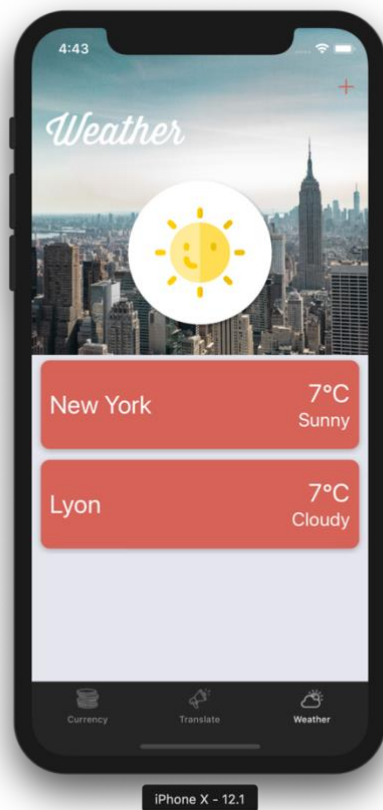


Figure 1 : La page Weather avec le bouton "+" en haut à droite

En appuyant sur ce bouton, une nouvelle page s'ouvre où l'on peut insérer le nom d'une ville afin d'obtenir les informations météo.



Figure 2 : La page pour ajouter une nouvelle ville

Cette page contient un UITextField où l'on tape le nom de la ville qu'on souhaite ajouter, un UIButton « Add » qui nous permet d'ajouter la ville écrite précédemment, et un UIButton « X » qui permet de revenir à la page Weather.

Lorsque l'on appuie sur le bouton « Add » on va tout d'abord récupérer le nom de la ville tapé dans le UITextField avec la fonction `getCityName()` dans une variable « cityName ».

```
@IBAction func addButtonDidTapped() {
    let cityName = getCityName()
    checkIfTableContainsCityName(cityName)
    cityTextField.resignFirstResponder()
}
```

Figure 3 : Les actions lorsque l'on appuie sur le bouton "Add"

La fonction `getCityName()` vérifie que le texte du TextField n'est pas vide sinon on fait apparaître une alerte. On renvoie donc la valeur du texte du TextField

```
func getCityName() -> String {
    if cityTextField.text == "" {
        showAlert(message: "You have to enter a city name !")
        return ""
    } else {
        guard let cityName = cityTextField.text else { return "" }
        return cityName
    }
}
```

Figure 4 : La fonction `getCityName()`

On fait ensuite appelle à la fonction `checkIfTableContainsCityName()`. Cette fonction va tout d'abord mettre la première lettre de la ville en capitale pour que toutes les villes du tableau soient homogénéisées. Puis on va comparer le nom de la ville au ville déjà présente dans le tableau, si elle y est déjà, on affiche une alerte, sinon nous allons utiliser la fonction `addNewCity()` du protocole `AddNewCityDelegate`.

```
func checkIfTableContainsCityName(_ cityName: String) {
    let capitalizedName = cityName.capitalized
    if !cities.contains(capitalizedName) {
        newCityDelegate.addNewCity(capitalizedName)
        dismiss(animated: true, completion: nil)
    } else {
        showAlert(message: "You already have added \(capitalizedName)")
    }
}
```

Figure 5 : La fonction `checkIfTableContainsCityName()`

En effet pour pouvoir transmettre des informations entre les deux Controllers, nous avons utilisé le Protocol/Delegate Pattern.

Dans le WeatherViewController, on crée une extension qui utilise le protocole AddNewCityDelegate. On implémente la fonction addNewCity() qui va permettre d'ajouter le nom de la ville au tableau de ville et ensuite relancer l'appel réseau pour faire apparaître les informations météo.

```
extension WeatherViewController: AddNewCityDelegate {  
    func addNewCity(_ name: String) {  
        cities.append(name)  
        displayWeatherInformations()  
    }  
}
```

Figure 6 : Extension de WeatherViewController

On peut également supprimer des villes en faisant glisser la cellule vers la droite. Grâce à une fonction déjà présente dans le protocole des tableViews

```
func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {  
    if editingStyle == .delete {  
        cities.remove(at: indexPath.row)  
        weatherInfo?.query.results.channel.remove(at: indexPath.row)  
        weatherInfo?.query.count -= 1  
        tableView.deleteRows(at: [indexPath], with: .automatic)  
    }  
}
```

Figure 7 : La fonction pour supprimer des cellules

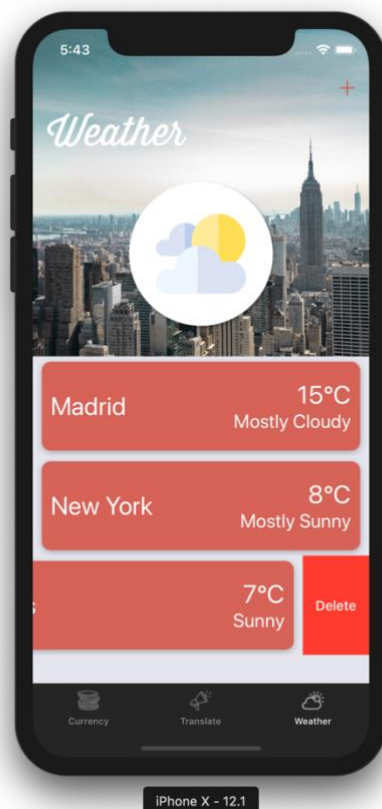


Figure 8 : Suppression de ville