

# Bonus

## Auto détection de la langue d'origine

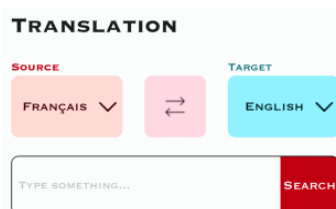
La détection de la langue d'origine du téléphone est faite grâce à la fonction suivante:

```
func getDeviceLang() -> String {  
    return Locale.current.languageCode?.capitalized ?? "Fr"  
}
```

Elle va chercher dans les réglage du téléphone la langue utilisée par le système. Si il n'y en a pas. Elle retourne « Fr » par défaut.

## Échange de la langue d'origine avec la langue cible

La langue source et la langue cible peuvent être facilement changé grace au bouton prévu à cet effet.



La fonction qui permet de faire ce « swap » est la suivante:

Dans le viewController:

```
func switchLangs() {  
    guard let target = viewModel.target else { return }  
    viewModel.swapSourceTraget(safeTarget: target)  
  
    sourceButton.updateLabel(text: viewModel.source)  
    targetButton.updateLabel(text: viewModel.target)  
  
    if #available(iOS 14.0, *) {  
  
targetButton.updateMenu(viewModel.getTargetLangs(for:  
viewModel.source))  
    }  
}  
  
@IBAction func switchButtonTapped() {  
    switchLangs()  
}
```

La fonction « switchButtonTapped() » capture l'événement d'appui sur le bouton et va exécuter la fonction « switchLangs » ensuite.

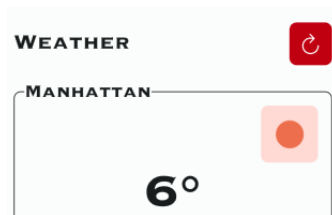
Dans le viewModel:

```
func swapSourceTraget(safeTarget target: String) {
    let tmp = source
    source = target
    self.target = tmp
}
```

Les langues source et cible sont interverties.

## Mise à jour en temps réel de la position de l'utilisateur

La position de l'utilisateur est mise à jour a chaque appuie sur le bouton de rafraichissement.



Grace à l'utilisation du Framework CoreLocation. Celui-ci permet de récupérer la position GPS de l'utilisateur grace à cette fonction système:

```
func locationManager(_ manager: CLLocationManager, didUpdateLocations
locations: [CLLocation]) {
    if let location = locations.first {
        let lat = location.coordinate.latitude
        let lon = location.coordinate.longitude
        delegate?.onUpdateLocation(lat: lat, lon: lon)
    }
}
```

Il ne reste plus qu'a appeler l'API avec ces nouvelles coordonnées:

```
func onLocationChanged(lat: Double, lon: Double) {
    loaderView.isHidden = false
    viewModel.fetchWeather(lat: lat, lon: lon) { result
in
        self.loaderView.isHidden = true
        switch result {
        case .success():
            self.updateUI()
        case .failure(let error):
            UIAlertController.showAlert(title: "Error",
message: "\(error)", on: self)
        }
    }
}
```