

## Projet 10 : Reciplease

---

### Présentation :

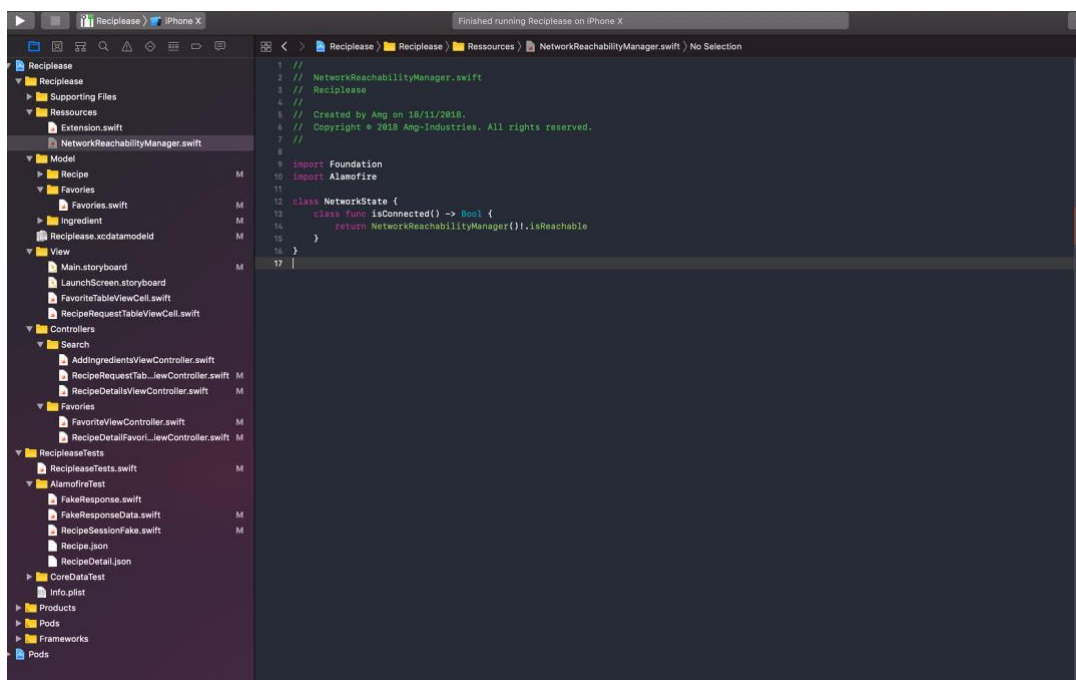
Dans le cadre de la formation Openclassrooms dans le parcours Développeur d'applications IOS. Je présente la feature créer en bonus pour le Projet 10 Reciplease.

### Explication de ma feature :

#### Feature sur l'utilisation utilisateur :

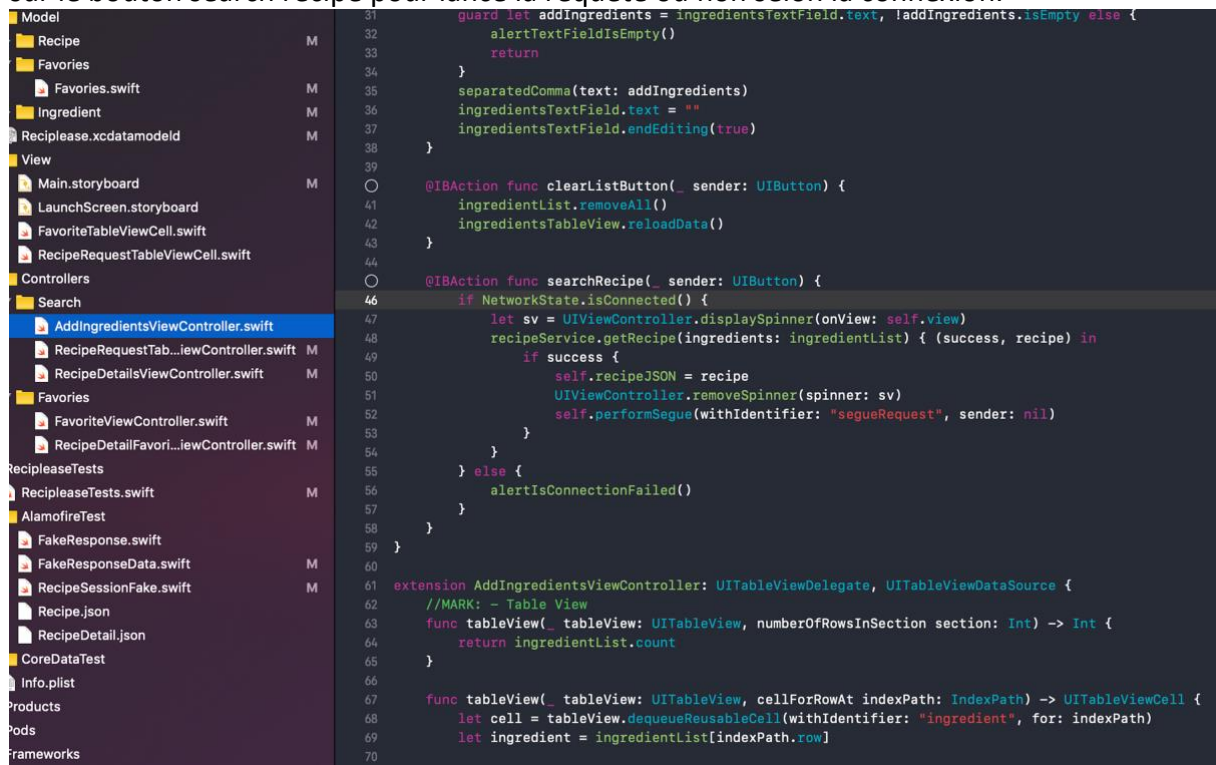
La première feature est network reachability qui va tout simplement vérifier lors du lancement de l'application le statut du réseau et le notifier s'il n'est pas connecté.

Pour cela j'ai premièrement créer un fichier dans le dossier ressources NetworkReachabilityManager. En important Alamofire j'ai créé une class networkState.



Et à l'intérieur de celle-ci j'ai appelé une fonction du Framework Alamofire qui permet de vérifier l'état du réseau.

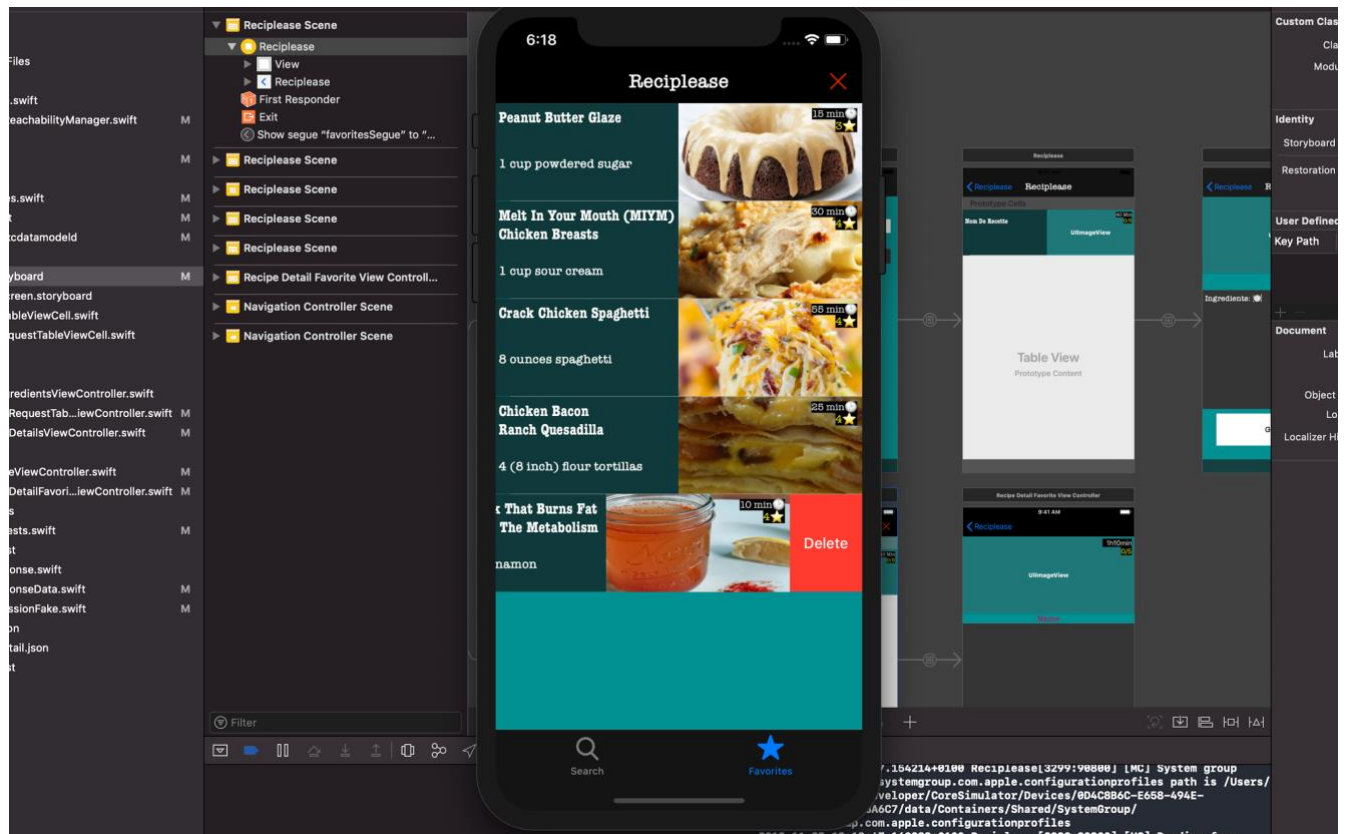
Puis je l'ai implémenter dans mon controller AddIngredientsController lorsque l'on appuie sur le bouton search recipe pour lancé la requête ou non selon la connexion.



The screenshot shows the Xcode interface with the project structure on the left and the Swift code for `AddIngredientsViewController` on the right. The project structure includes folders for `Model`, `Recipe`, `Favorites`, `Ingredient`, `Reciplease.xcdatamodeld`, `View`, `Controllers`, `Search`, `Favorites`, `RecipleaseTests`, `AlamofireTest`, `CoreDataTest`, `Info.plist`, `Products`, `Pods`, and `Frameworks`. The Swift code is as follows:

```
31 guard let addIngredients = ingredientsTextField.text, !addIngredients.isEmpty else {
32     alertTextFieldIsEmpty()
33     return
34 }
35 separatedComma(text: addIngredients)
36 ingredientsTextField.text = ""
37 ingredientsTextField.endEditing(true)
38 }
39
40 @IBAction func clearListButton(_ sender: UIButton) {
41     ingredientList.removeAll()
42     ingredientsTableView.reloadData()
43 }
44
45 @IBAction func searchRecipe(_ sender: UIButton) {
46     if NetworkState.isConnected() {
47         let sv = UIViewController.displaySpinner(onView: self.view)
48         recipeService.getRecipe(ingredients: ingredientList) { (success, recipe) in
49             if success {
50                 self.recipeJSON = recipe
51                 UIViewController.removeSpinner(spinner: sv)
52                 self.performSegue(withIdentifier: "segueRequest", sender: nil)
53             }
54         }
55     } else {
56         alertIsConnectionFailed()
57     }
58 }
59
60 extension AddIngredientsViewController: UITableViewDelegate, UITableViewDataSource {
61     //MARK: - Table View
62     func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
63         return ingredientList.count
64     }
65
66     func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
67         let cell = tableView.dequeueReusableCell(withIdentifier: "ingredient", for: indexPath)
68         let ingredient = ingredientList[indexPath.row]
69     }
70 }
```

La seconde feature est implémenter dans ma table View favorites lors de l'affichage des recettes. Elle nous permet de pouvoir faire un swipe left pour supprimer une cellule de recettes.



Pour l'implémentation de celle-ci. J'ai utilisé la méthode `commitEditingStyle` qui me permet de supprimer à l'index path la recette choisi en la supprimant également de notre base de données.

```

//MARK: ~ TableView
func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return favoritesAll.count
}

func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: IndexPath) -> UITableViewCell {
    guard let cell = tableView.dequeueReusableCell(withIdentifier: "favoriteCell", for: indexPath) as? FavoriteTableViewCell else {
        return UITableViewCell()
    }
    let favorite = favoritesAll[indexPath.row]
    let ingredient = ingredientAll[indexPath.row]

    cell.favorites = favorite
    cell.ingredientsLabel.text = ingredient.name
    return cell
}

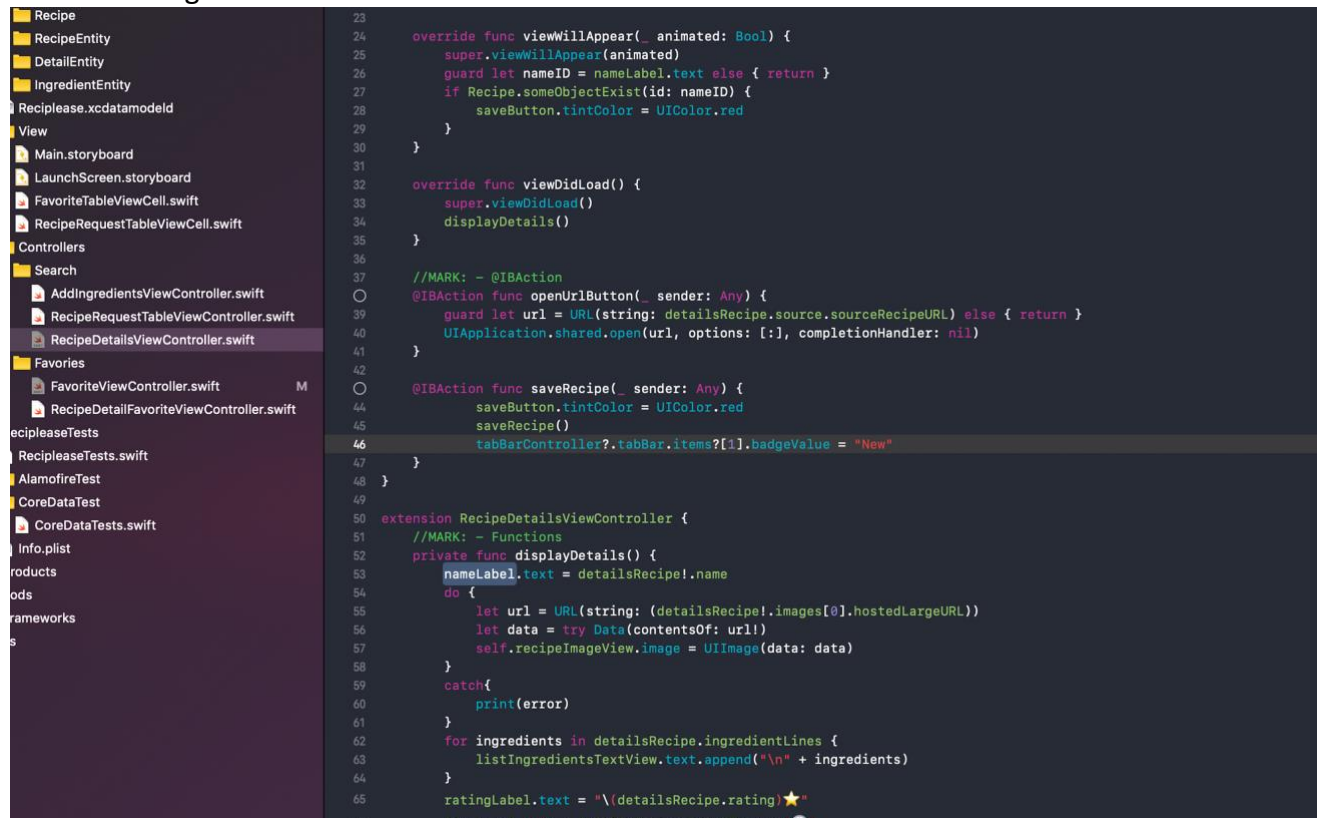
func tableView(tableView: UITableView, commitEditingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {
    switch editingStyle {
    case .delete:
        AppDelegate.viewContext.delete(favoritesAll[indexPath.row])
        favoritesAll.remove(at: indexPath.row)
        do {
            try AppDelegate.viewContext.save()
        } catch {
            print("error")
        }
        self.favoritesTableView.deleteRows(at: [indexPath], with: .automatic)
    default: break
    }
}

func tableView(tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    guard let cell = tableView.cellForRow(at: indexPath) as? FavoriteTableViewCell else { return }
    self.performSegue(withIdentifier: "FavoritesSegue", sender: cell.favorites)
}

func tableView(tableView: UITableView, viewFooterInSection section: Int) -> UIView? {
    if favoritesAll.isEmpty {
        let label = UILabel()
        label.text = "Get Recipe and Save it"
        label.font = UIFont.systemFont(ofSize: 17, weight: .semibold)
        label.textAlignment = .center
        label.textColor = .white
        return label
    }
    return UIView()
}

```

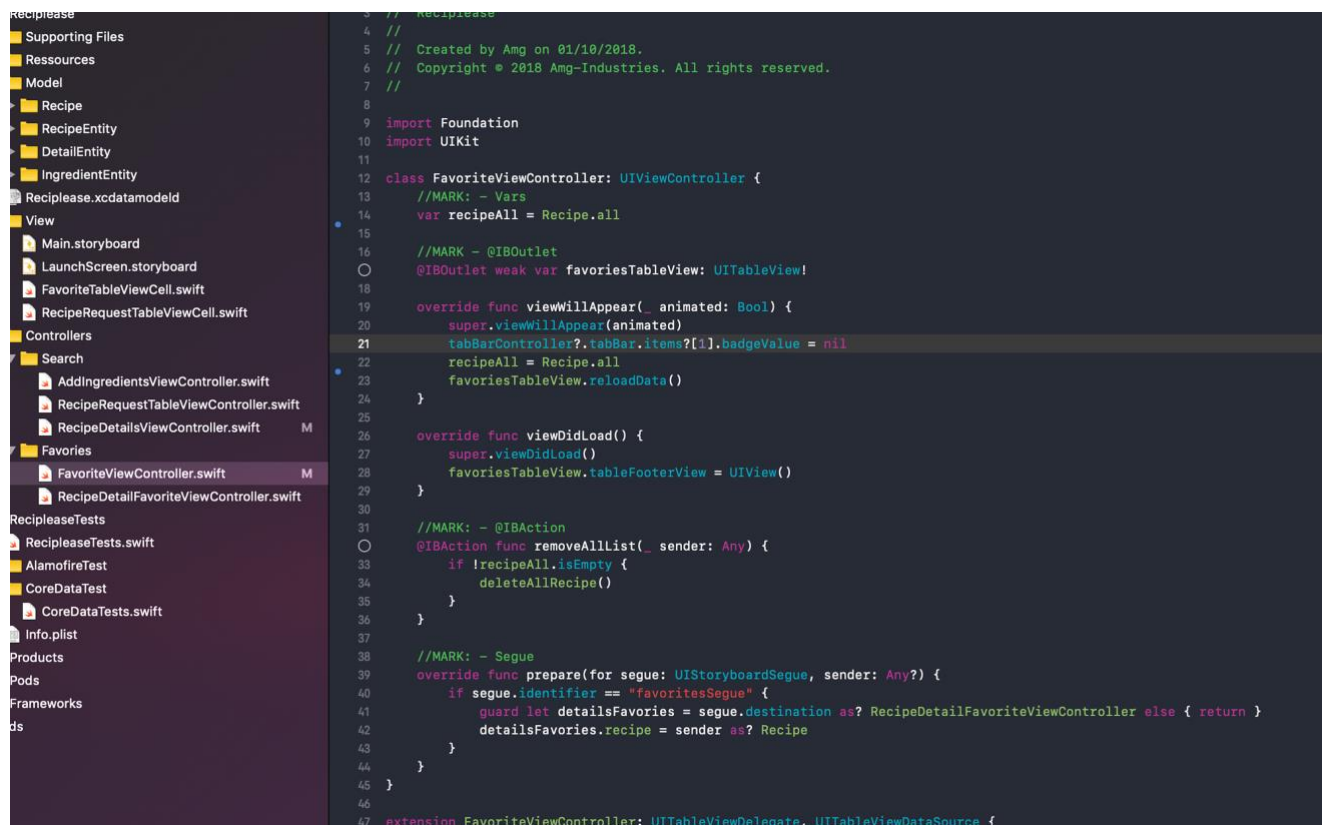
Et pour finir, ma dernière feature est un badge qui s'affiche à ma tab bar Favorites lorsque j'enregistre une recette dans ma base de données. Et qui disparaît lors de l'affichage des recettes enregistrées.



```

23
24 override func viewWillAppear(_ animated: Bool) {
25     super.viewWillAppear(animated)
26     guard let nameID = nameLabel.text else { return }
27     if Recipe.someObjectExist(id: nameID) {
28         saveButton.tintColor = UIColor.red
29     }
30 }
31
32 override func viewDidLoad() {
33     super.viewDidLoad()
34     displayDetails()
35 }
36
37 //MARK: - @IBAction
38 @IBAction func openUrlButton(_ sender: Any) {
39     guard let url = URL(string: detailsRecipe.source.recipeURL) else { return }
40     UIApplication.shared.open(url, options: [:], completionHandler: nil)
41 }
42
43 @IBAction func saveRecipe(_ sender: Any) {
44     saveButton.tintColor = UIColor.red
45     saveRecipe()
46     tabBarController?.tabBar.items?[1].badgeValue = "New"
47 }
48
49
50 extension RecipeDetailsViewController {
51     //MARK: - Functions
52     private func displayDetails() {
53         nameLabel.text = detailsRecipe!.name
54         do {
55             let url = URL(string: (detailsRecipe!.images[0].hostedLargeURL))
56             let data = try Data(contentsOf: url!)
57             self.recipeImageView.image = UIImage(data: data)
58         } catch {
59             print(error)
60         }
61         for ingredients in detailsRecipe.ingredientsLines {
62             listIngredientsTextView.text.append("\n" + ingredients)
63         }
64         ratingLabel.text = "\(detailsRecipe.rating)★"
65         titleLabel.text = detailsRecipe.totalTime + "⌚"
66     }
67 }

```



```

3 // Recipe
4 //
5 // Created by Amg on 01/10/2018.
6 // Copyright © 2018 Amg-Industries. All rights reserved.
7 //
8
9 import Foundation
10 import UIKit
11
12 class FavoriteViewController: UIViewController {
13     //MARK: - Vars
14     var recipeAll = Recipe.all
15
16     //MARK: - IBOutlet
17     @IBOutlet weak var favoritesTableView: UITableView!
18
19     override func viewWillAppear(_ animated: Bool) {
20         super.viewWillAppear(animated)
21         tabBarController?.tabBar.items?[1].badgeValue = nil
22         recipeAll = Recipe.all
23         favoritesTableView.reloadData()
24     }
25
26     override func viewDidLoad() {
27         super.viewDidLoad()
28         favoritesTableView.tableFooterView = UIView()
29     }
30
31     //MARK: - @IBAction
32     @IBAction func removeAllList(_ sender: Any) {
33         if !recipeAll.isEmpty {
34             deleteAllRecipe()
35         }
36     }
37
38     //MARK: - Segue
39     override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
40         if segue.identifier == "favoritesSegue" {
41             guard let detailsFavorites = segue.destination as? RecipeDetailFavoriteViewController else { return }
42             detailsFavorites.recipe = sender as? Recipe
43         }
44     }
45 }
46
47 extension FavoriteViewController: UITableViewDelegate, UITableViewDataSource {

```