МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

Реферат

Тема: Описание предлагаемого способа

Студент гр. 4303 _____ Ахриев Р.А.

Преподаватель _____

Санкт-Петербург
2019

1. СТРУКТУРА ФРЕЙМВОРКА

1.1 Описание используемых технологий

Данный фреймворк будет распространяться, как суб-модуль, который будет встраиваться в проект с помощью менеджера зависимостей и содержать в себе ряд компонентов, которые будут решать наиболее типичные для приложения задачи (Рисунок 1).

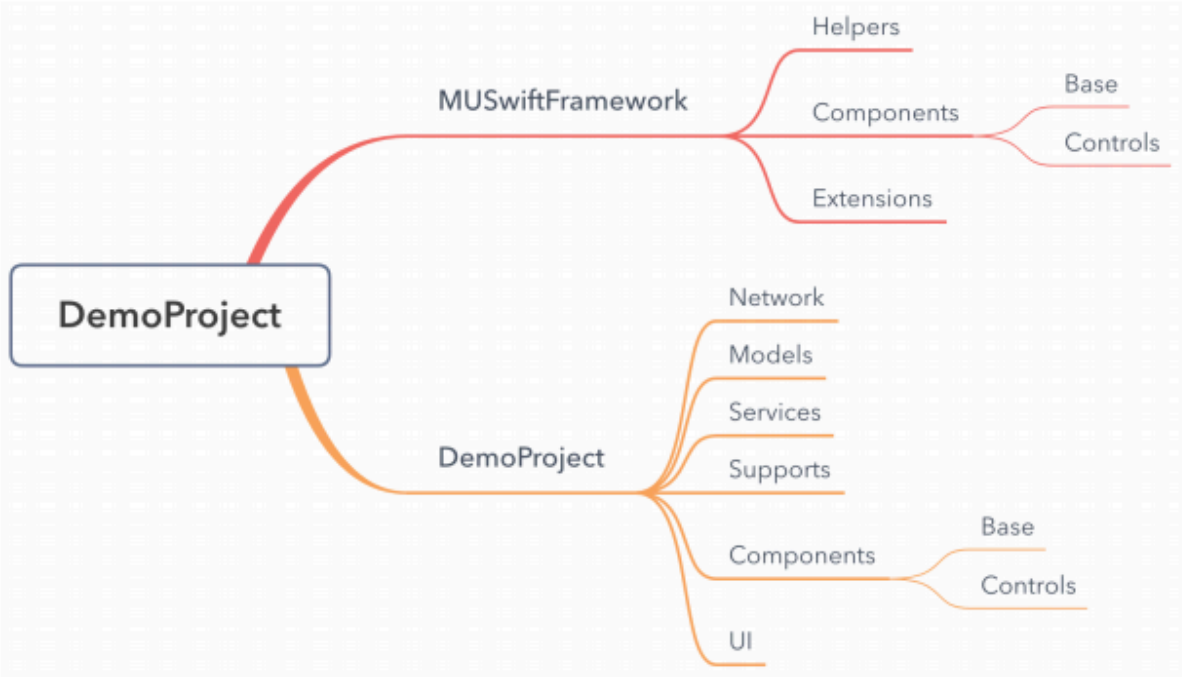

Рисунок 1 – пример структуры проекта

В качестве используемых языков предполагается следующий набор:

- Swift
- Objective-c
- Ruby
- Bash

1.2. Структура фреймворка

Фреймворк имеет доступную и интуитивно понятную структуру (Рисунок 2):

- Helpers – менеджеры для совершения различных операций, такие как запросы, запись в базу данных и т.д.
- Components – UI компоненты, с расширенным функционалом
- Extensions – расширения стандартных классов Apple API.

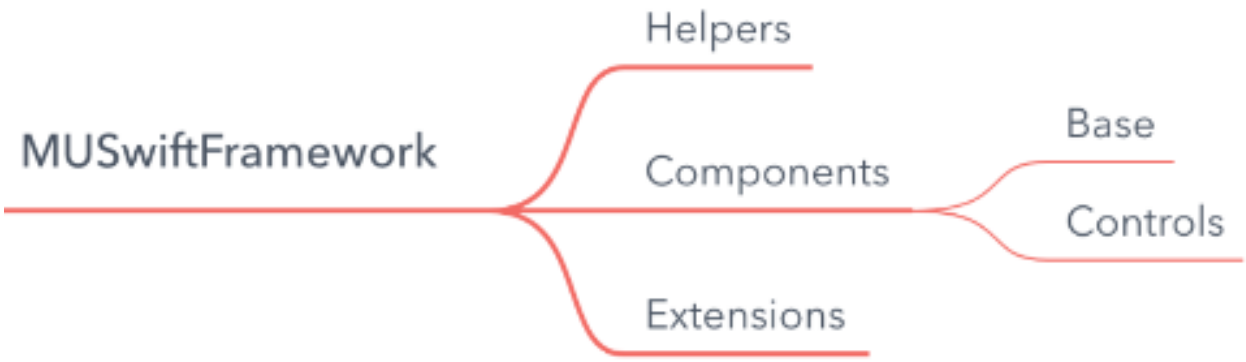

Рисунок 2 – структура фреймворка

1.3. ListController

Контроллер дополняющий функционал компонентов UITableView, UICollectionView. Имеет базовый функционал:

- Подгрузка данных с сети
- Группировка данных по свойству
- Анимациия ячеек
- Кэширование данных в файл
- Пагинация при infinity scroll

```
// MARK: - DemoListController

class DemoListController: ListController {

    class override var storyboardName: String { return "DemoList" }

    // MARK: - Override properties

    override var hasRefresh: Bool { return true }

    override var hasPagination: Bool { return true }

    override var hasEmptyState: Bool { return true }

    // MARK: - Private properties

    @IBOutlet private weak var tableProvider: UITableView! { didSet { tableView = tableProvide

    @IBOutlet private weak var emptyViewProvider: UIView! { didSet { emptyView = emptyViewProv

    // MARK: - Override methods

    override func beginRequest() {

        DemoService.getAll() { [weak self] (objects) in

            self?.update(objects: objects)
        }
    }
}
```

Рисунок 3 – листинг примера наследования от ListController

Данный компонент является лишь частью фреймворка и позволяет значительно ускорить возможность создания и отображения списка, загружаемого из сети.