

Пояснительная записка

Поставленные задачи:

- 1 – Получение с сервера JSON файла с сервера, который содержит список данных для отображения на странице нескольких типов и массив со списком порядка отображения данных на устройстве;
- 2 - Отображение списка элементов;
- 3 – Реализовать возможность вывода информации при клике на элемент или выборе одного из вариантов ответа;

Реализация проекта:

Перед открытием проекта мною было принято решение использование архитектуры MVVM, чтобы потом перестроиться на MVP. Причиной этому послужило отсутствие опыта работы с MVP и редкостью использования данной архитектуры при написании приложений на Swift.

Итак, п. 1 – JSON Parsing. Убедившись в простоте использования протокола Decodable я решил не подгружать сторонние библиотеки (SwiftyJSON/ObjectMapper). Структуры для разобранного JSON файла находятся в папке ObjectData.

п.2 – Multiple cell types. Для простоты доступа к ячейке с данными нужного типа создана структура ModelItemType, включающая в себя протокол и расширение. Так же создана структура для данных каждого типа.

п.3 – Custom cells. Созданы отдельные ячейки для каждого типа данных. Каждая ячейка с уникальным идентификатором зарегистрирована в функции cellRegister() PryanikyMainView.

п.4 – Presenter. Адрес jsonUrl известен – объявил приватную переменную. Создан массив [ModelType] для хранения данных. Так как запрос в сеть идет асинхронно с главным потоком и ответ с сервера приходит уже после конфигурации UI элементов, то я добавил callback и handlers. В блоке do-catch использован цикл for-in для удовлетворения требования отображения последовательности списка элементов.

п.5 - Picture. Чтобы не хранить загруженное изображение на устройстве я решил использовать стороннюю библиотеку KingFisher.

п.6 – Delegate. Так как методы UIPickerView реализованы внутри кастомной ячейки, то появилась необходимость уведомить main view об изменении значения. Создал протокол класса SelectorCellDelegate с единственным методом, который позволит передать значение положения пикера. Для избежания утечки памяти создал weak переменную delegate внутри класса SelectorCell.

п.7 – View – постарался максимально разгрузить View от логики с использованием функционального программирования.