Ionic. Общий обзор

Недавно в ходе работы над проектом, мне пришлось столкнуться с новым для меня фреймворком: Ionic. Поэтому у меня возникла необходимость ознакомиться с этой технологией и получить общее представление о ней, чтобы разобраться с особенностями проекта. Вот мои впечатления после первого знакомства с Ionic.

Ionic − HTML5 фреймворк, позиционируется как инструмент для легкого и быстрого написания мобильных приложений с использованием HTML, CSS и JavaScript. Он использует паттерн MVC и работает в нем только с частью view. Ionic представляет собой набор CSS и JavaScript компонентов и набор иконок, и по способу использования и той нише, которую занимает в проекте, напомнил мне Tweeter Bootstrap: необходимые компоненты встраиваются в HTML код, а для некоторых JS компонентов требуется активация или настройка в скрипте. На сайте Ionic предоставляется документация, с описанием использования всех компонентов, однако мне она показалась достаточно сдержанной и краткой, поэтому далеко не всегда мне ее хватало, чтобы разобраться с тем, как работает тот или иной элемент и как его правильно интегрировать в код приложения, приходилось дополнительно искать примеры использования компонентов или решения конкретных проблем.

Что касается модели и контроллера, общей архитектуры приложения, то тут Ionic для полноценной работы должен встраиваться в Angular, причем работает он как с первым, так и со вторым Angular и вся документация на сайте представлена в двух вариантах. Несмотря на популярность Angular, это кажется мне серьезным минусом, поскольку тот самый быстрый старт возможен только если вы знаете и готовы использовать Angular, в противном же случае, нужно будет либо осваивать дополнительную технологию, либо искать другие инструменты для проекта.

Для запуска приложения на мобильных устройствах Ionic нужна обертка. Он использует платформы Cordova или PhoneGap, и, соответственно, поддерживает некоторый набор плагинов на этих платформах. Перед загрузкой Ionic нужно установить Cordova:

$ sudo npm install -g cordova

Также потребуются инструменты для разработки под конкретную платформу: [iOS](http://cordova.apache.org/docs/en/latest/guide/platforms/ios/index.html) или [Android](http://cordova.apache.org/docs/en/latest/guide/platforms/android/index.html).

Сам Ionic, также ставится через npm:

$ sudo npm install -g ionic

Новое приложение можно сгенерировать используя стандартные шаблоны, которых Ionic предоставляет: blank, tabs, sidemenu и прочие. Полный список можно посмотреть с помощью команды:

$ ionic start --list

Все шаблоны кроме blank дают возможность познакомиться с самой общей схемой использования фреймворка. Для того, чтобы получить шаблон приложения достаточно использовать:

ionic start projectName tabs

где projectName -- имя папки которая будет создана для проекта, а tabs -- тип шаблона, в данном случае с навигацией в виде табов.

Однако более полезным инструментом для знакомства может послужить Ionic Creator ( [https://creator.ionic.io](https://creator.ionic.io/)), который позволяет довольно быстро создать несколько более сложные шаблоны. Мне трудно сказать, насколько хороший это способ создания приложения, однако весьма удобный вариант изучения того, как используются различные компоненты при первом знакомстве.

Системы навигации представленные фреймворком предполагают реализацию в виде одностраничного приложения. Например в сгенерированном шаблоне с табами файл index.html имеет следующую структуру:

<body ng-app="starter">

<ion-nav-bar class="bar-stable">

<ion-nav-back-button>

</ion-nav-back-button>

</ion-nav-bar>

<ion-nav-view></ion-nav-view>

</body>

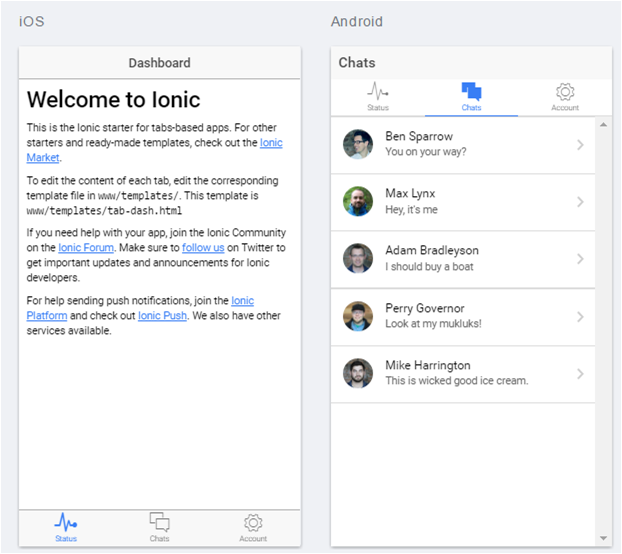
Навигационный компонент ion-nav-bar связан с контейнером ion-nav-view, в который загружается один из шаблонов исходя из выбранной вкладки или пункта меню, роутинг осуществляется средствами Angular.

В целом после некоторых экспериментов использование фреймворка кажется достаточно простым, если структура приложения предполагает какой-то из вариантов стандартной навигации. Но при усложнении интерфейса и необходимости использовать элементы вложенные друг в друга, могут возникать вопросы.

Помимо генерации приложения, предоставленный фреймворком CLI имеет еще ряд возможностей. Например можно запустить локальный сервер для тестирования приложения:

# $ ionic serve

Если использовать эту же команду с флагом --lab, то в браузере выводится страница, на которой можно посмотреть, как будет выглядеть продукт на iOS и Android платформах.



Можно генерировать иконки и сплэшскрин (экран загрузки, который пользователь видит пока приложение загружается) из одного исходника под разные платформы, с помощью команды:

$ ionic cordova resources **[**platform]

Разумеется, можно собирать приложение, эмулировать запуск на конкретной платформе, запускать непосредственно на мобильном устройстве, генерировать отдельные компоненты -- полный список команд с примерами лучше смотреть в [документации](http://ionicframework.com/docs/cli/).

В общем и целом, мне кажется что Ionic достаточно удобный инструмент, имеющий тем не менее серьезные ограничения как в сфере используемых совместно с ним технологий так и в уровне сложности осуществляемого проекта.

Агамян Ирина. Апрель 2017