Kevin Chen

3/17/18

Ionic Notes

**Basics**

Ionic Overview

* Framework for building cross-platform mobile apps
* Includes UI, UX, and other services
* The Command Line Program (CLI) is used to create, emulate, test, and build projects.
* Ionic Creator is a visual creator for creating Ionic applications.
* Ionic Cloud is a set of commercial services.
* Ionic View is used for sharing/previewing an app on a real mobile device

**Installations**

Installation Requirements

* Node and npm, Devices and USB Cables, Mobile SDKs (Android/iOS native SDKs), and text editor
* npm is needed to install CLI. Installing Node is the easiest way to get npm.
* Run the following commands:
  + (sudo) npm install –g cordova
  + (sudo) npm install –g ionic

**Your First Application**

Make the Default Application

* Ionic applications are driven by the CLI.
* “ionic start” creates a new application
  + “ionic start <app\_name> --v<version\_number>” creates a new app with the name and version number. Example: “ionic start first\_app --v2”.
* This generates Cordova Resources and things specific to Ionic
  + Config.xml, hooks, platforms, plugins, and resources are used by Cordova.
  + Remainder of files are related to Ionic.
  + The app itself is in the “src” folder.
  + Ionic converts Angular II typescript code into simpler JavaScript code in www folder.
* How to see the app:
  + Method 1: in a browser. “ionic serve”
  + Method 2: in an emulator
  + Method 3: on the device
* Type “ionic serve” to run in a browser.
  + The application is automatically reloaded in the browser whenever you make a change.
* “ionic platform ls” lists all the platforms that the app can be built for.
  + “ionic platform add ios” to add IOS as a platform for the app to be built for.
  + “ionic platform add android”
  + “ionic platform remove ios” to remove iOS as a platform.
* “ionic emulate” emulates the app on all installed platforms.
  + “ionic emulate ios” for emulating just on an iOS emulator.

**Application Components**

Exploring Application Files

* “Hooks” folder lets you set up scripts while working with the project.
* “Plugins” are where installed plugins live.
* “Platforms” are the actual bundles of code for each individual platform supported by an application.
* “Node\_modules” and “package.json” are how support files are used for building the app.
* “src” are all the files seen in an Ionic application.
  + Index.html just handles bootstrapping and starting the Angular 2 app.
  + “app” folder contains files that manage how an app works as a whole…things that impact the entire application.
  + “pages folder” contains directories representing every page of the app. Each folder contains an HTML file (for what is seen), a TypeScript file (the component code, defines how the page behaves), SASS .scss file (elements that apply just for that particular page).
  + “theme” folder is for global theming
* “www” folder is where the compiles assets live.

Navigation components

* A single application metaphor is useful for simple sites. It loads the whole site in one page and uses JavaScript to modify the DOM when the user goes to another page.
* Navigation components are the UI for transitioning between views.
* There are various navigation components to support different application types.
* Tabbed application for creating a site with multiple tabs.
  + “ionic start tabs” creates a new application with tabs.
* The side menu is another navigation component.
  + “ionic start sidemenu” creates a new application with side menus.

UI Components

* Ionic provides components to create professional/good-looking mobile apps.
  + They are optimized to look great on mobile.
* Buttons component:

“<button ion-button>Regular Button</button>” creates a regular button.

“<button ion-button block>Block Button</button>” creates a long button.

* Almost all UI components have colors. There are 5 primary colors. Example:

“<button ion-button block color=“danger”>Block Button</button>” creates a long button.

* Cards component puts a rectangular border around content. Example:

<ion-card>

<ion-card-header>My Card</ion-card-header> <!-- Optional -->

<ion-card-content>

<p>Some kind of content…</p>

</ion-card-content>

</ion-card>

* Lists example:

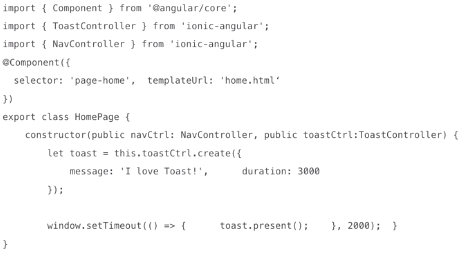
<ion-list>

<ion-item>Item 1</ion-item>

<ion-item>Item 2</ion-item>

</ion-list>

* Inset List adds a little space to the items on both sides: <ion-list inset>
* List Headers can be added into the list between list items: <ion-list-header>
* Toasts can be added. Example:



* Toolbars. Headers and footers.
* There are a lot more components!

UX Components

* These are the interactive components of an application.