Mobile Framework v 1.0

TABLE OF CONTENTS

I. A need for a mobile framework

EPRI has a need for a mobile framework to provide the Institute a way to rapidly develop mobile applications with quality and consistency.

Reducing the development cycle

The typical development cycle can last several months in most cases. We can reduce the cycle tremendously by using componentized, reusable user interfaces and providing a predictable and familiar user experience across all of EPRI's applications. By using defined components and UX patterns, developers will only need to figure out which components are necessary to compose the page, configure them, and then test them.

Consistent user experience

One thing that we have to do as an institute, is have a consistent look and feel across all of our web and mobile applications. Using the same components and patterns will provide our users with a familiar way to navigate and interact with our websites and applications, reducing confusion and increasing user adoption.

Quality development that isn't dependent on outside vendors, reducing costs across the board

By providing a mobile framework, we will reduce EPRI's dependence on outside vendors and reduce our costs by reducing the development cycle. Instead of having to go through a lengthy discovery process, discussion on which framework or technology to use, testing, refining, and hoping that the stakeholders and developers are on the same page, we will have a framework that has specific components and patterns that developers will be able to implement with a consistent, professional look and feel, as well as predefined ways to interact with EPRI APIs.

Additionally, this type of development assures that EPRI isn't dependent on any specific vendor to make additions or changes to existing technology. EPRI should be able to have an application built by one vendor, then have another vendor easily edit or update it later on. This approach will help to ensure that vendors don't take advantage of EPRI by building half-baked, proprietary applications that use technologies only they know how to use.

II. Ionic: What is it?

What is Ionic

Ionic is a fully cross-platform framework that will allow developers to build web and native mobile apps for every major app store with one codebase. Ionic features a simple CLI to create, build, test, and deploy apps onto any platform. Ionic Native unlocks native APIs and features by wrapping Cordova plugins in TypeScript.

How to use it

Ready to get started? **Download the EPRI starter application** and then check out the Introduction to Ionic section from the Ionic Framework website.

Additionally, check out our **EPRI Web and Mobile Style Guide** to see what all you can do.

III. SQA / W&M Process

Any application must go through the SQA process outlined at http://swdev.epri.com/req-webapps.asp. Contact **Manuel Morales** (MaMorale@epri.com) for more information.

Web and Mobile solutions is available for a scheduled consultation during your planning phase. A technology review and recommendation and a wireframing of user patterns may be provided upon request. Please reach out to **Bing Xie** (bxie@epri.com) to schedule your Web and Mobile Solutions consultation.

IV. Getting started with Ionic

Once you have gone through your SQA process and had your Web and Mobile consultation, It's time to get started with the Ionic Framework.

Ionic apps use Cordova to build and deploy as a native application through the command line utility. You will need to install a few utilities before beginning development.
Refer to the **Installing Ionic** section in the Ionic Framework documentation, located at https://ionicframework.com/docs/intro/installation/ for complete instructions.

Additionally, you may need to review the Cordova iOS Platform Guide and follow the instructions to install Xcode, and possibly register for a developer account to start building apps for iOS.

You will also want to review the Cordova Android Platform Guide and follow the instructions to install the SDK and Android Studio for Android development.

To Start a New App

To start a new App using Ionic, we recommend that you **download our Starter Applications** that have many of the pieces already in place for you to review and use as a beginning point for your development. They have some of the core patterns and styles already in place, so you can start building right away. The Starter Apps also contain our **EPRI Login** and **Search patterns** baked-in, so you don't have to worry about reinventing the wheel.

We suggest reviewing the Ionic Tutorial before you begin. **More:** The Basics | Project Structure | Adding Pages | Navigating to Pages | Component Docs | API Docs

Theming

EPRI has a default theme that you must use in all of your applications and is included in the **EPRI Starter Apps**.

Component Library

Ionic comes with several components that you can insert into your application. Please refer to the **Web and Mobile Style Guide** for tips on how many of these should look and function. Since we have adopted the Google Material spec, anything not covered in our Starter App or referenced in our Style Guide should be built using the Material specification. Please consult with **Brad Thomas** (bthomas@epri.com) regarding any new components that you may need for your project.

V. Ionic Pro Quick Start

1. Get the appropriate certs from SQA based on the existing process.
2. Send an email to **Matt Keener** mkeener@epri.com to get access to the EPRI Ionic Enterprise account.
3. Install Ionic: npm i -g ionic
4. Add Ionic as a remote in your Git repository: git remote add ionic [your_repo]
5. Push to Ionic: git push ionic master

VI. Ionic Pro Concepts

Builds

The Git Workflow at the core of Ionic Pro takes commits of your app code, builds them in the cloud, and then makes it possible to distribute and manage those builds through a simple Dashboard interface and CLI tools.

A Build in Ionic Pro is triggered when running git push ionic master with new changes to your app. Each build goes through a CI system that triggers a full build of your app (much like running npm run build), and the completed build is stored for later use through one of the core Ionic Pro services.

Channels

A Channel points to a specific Build of your app that will be shared with specific testers and users. You can change which build a Channel points to whenever you'd like, and can rollback changes as well. Channels are used for both Ionic View and Ionic Deploy.

When you share your App with Ionic View, you're sharing a specific Channel. Whenever you update that Channel (by clicking Deploy next to a Build), users with access to that Channel in View will receive the update.

When you integrate Deploy with your App (by clicking Set Up Deploy on a Channel), the native binary set up with that Channel will poll that Channel for updates whenever it is loaded.

Setting Up a Channel

Navigate to the Code tab in your App, then click on Channels. Here you'll be able to create a new Channel.

Deploying to a Channel

Once you have created a Channel, go back to the Builds section of your app. Next to each Build, you'll see a "Deploy" button. Click this to set the Build as active for a Channel.

Once a Build is set as active for a Channel a few things will happen:

- Anyone who has access to this Channel (through the Share App Private or Public features) will receive the new Build and be able to look at it in View.
- Any native binaries that have been set up with Ionic Deploy for this Channel will receive the new update next time they check in.

Using Public Channels

In order to share your App publicly in Ionic View, you need to have at least one Channel set to public. Click on a Channel and then click the Settings tab, then check "Public in Ionic View" and click Save.

Automating Deployment from a Git Branch

Instead of clicking Deploy on a Build to set it as active for a Channel, you can also automate your Deployments using Git Branches. By default, we've set up
a Master Channel for you that automatically deploys from the Master git branch.

To set up a Channel for automatic git->Channel deployments, open that Channel and click on the Settings tab. Then type the name of the Channel you want to update from and click Save.

The next time you git push ionic that-channel it will automatically deploy to your Channel.

Sharing an Ionic app with Ionic View

In order to test anything in Ionic View, you will have to assign your Builds to a Channel. These Channels are what you will use to share a version of your app with someone.

For instance, you could set up a Staging Channel that you can share with your QA Testers, a Marketing Preview Channel that you can share with the Marketing department to give them an early preview, a Public Channel to be shared with anyone via a QR code, etc.

You have complete control over these Channels and what Build is set to active for them. If you push a new build to Staging, anyone who has been given access to that Channel will be able to see the update right in Ionic View.

Once you have a Channel set up to share, click on the Share App button in the top right of your Apps dashboard. Here you'll be able to choose which Channel to share, and manage the email addresses you've shared with.

These users will receive an email with a link that will automatically download Ionic View for them, and then grant them access to your App.

Deploying an Ionic App

Please visit the Ionic Docs for the latest deployment instructions:
https://ionicframework.com/docs/pro/deploy/setup/#installation

Cloud Build

Click on the Code tab of your App in the Ionic portal, you should be at your Builds list. Click on the Package button for the Build you'd like to get a Native binary for.

In the Package GUI you'll be able to choose the type of Build you want to perform, and the Cert you'd like to use. You'll then be taken to the History tab while the build completes. After the build completes, click Download and you'll get your native binary!

VII. EPRI Web and Mobile Review

Once you feel your app is complete, contact the Web and Mobile team to conduct a Technology and User Experience Review. This is an informal review that can often help catch any bugs and help test your application flow. Once you have complete this review, you can proceed to the final deployment and submission steps outlined by the EPRI SQA team.