

Tool Chain Options for Hybrid Mobile App Development

Saurabh Kumar Singh

BITS Pilani

E-mail: woltort@gmail.com

Abstract—Gartner says ‘by 2017, mobile users will provide personalized data streams to more than 100 apps and services everyday’. In today’s world, from entertainment content to productivity services, from quantified-self to home; there is a mobile applications for practically anything a connected consumer is looking around. To address this increasing need for user-friendly mobile applications, enterprises are looking to leverage applications across multiple platforms. The need for content awareness in mobile apps has increased with the capabilities of mobile devices, causing developers to consider both native and hybrid architectures.

Generally app developers purchase a framework and do not explore the open source options, which is not a good approach. There are numerous cross platform frameworks available in the market today. One must look at popular frameworks available in the open source community. This is important as each mobile app has different requirement and can be optimized with a good choice of framework. This paper presents the research work that highlights the importance of framework selection for mobile app development. The research covers the framework that savors the backing of a huge developer community. As a part of our research, examination is done for mainly three frameworks: Sencha Touch 2, jQuery Mobile and Backbone + Twitter Bootstrap. The above mentioned frameworks have been compared from the point of view of a developer on multiple parameters. This comparison of cross-platform mobile app development framework is helpful for any developer as it stems the selection process of a correct tool chain to create app out of an idea. The results of the research shows that for a developer with “mobile first” approach shall use Sencha Touch 2, developer with “web first” approach must pick jQuery Mobile and a developer who wants to create a MVC based mobile app must go for Backbone+Bootstrap.

1. INTRODUCTION

The world of app development is an exciting, yet sometimes confusing place. There is no definitive answer as to which type of app you should go for. Your choice will depend on you, your budget and time-scale. Currently this world is divided between native and hybrid apps. Hybrid apps have created a following within a small span of time due to numerous reasons. Like the websites on the internet, hybrid mobile apps are built with a combination of web technologies like HTML, CSS, and JavaScript. The key difference is that hybrid apps are hosted inside a native application that utilizes a mobile platform’s web view. This enables them to do things like access hardware capabilities of the mobile device.

Hybrid mobile applications provide a way for developers to re-use their existing skills in web development. Developers don’t like the prospect of getting locked into proprietary platforms. This includes the programming languages and SDKs provided by platform vendors (more on this later).

Hybrid mobile application development looks appealing to an organization’s bottom line. This approach saves the overheads of hiring a developer for each platform when you can hire one developer and target all of them through HTML, CSS, and JavaScript instead. Below are some of the famous companies working with hybrid mobile apps :

- Netflix
- Amazon
- Evernote
- Feedly

Native mobile app development though new has already become full of options in context to available framework in the market. The biggest hurdle a mobile app developer faces is the dilemma of which framework to go with.

This paper takes a holistic approach to analyze three tool chain options and comes up with cases of when to use what. What this paper does not say is which tool chain is better than the other.

2. TOOL CHAIN ANALYSIS

2.1 Tool Chain A

This tool chain is Sencha Touch 2 based. The tool chain is widely popular among hybrid app developers. This tool chain has the following components:-

Sencha Touch 2: The framework based on MVC structure has all the components that are needed right from designing the app to packaging it. This provides a commercial free license.

Sencha Architect: It is paid IDE by Sencha and is very handy as it provides a full blown developer environment.

Google Chrome: All the hybrid apps are run and tested on a webkit based browser. Here Google Chrome is a personal

choice as Chrome Developer Tools are very helpful. This component covers the following roles in the tool chain:-

- Testing
- Debugging

2.2 Tool Chain B

This tool chain is jQuery Mobile based. The tool chain is very popular among the open source community as it is a sub set of the famous jQuery framework.

- jQuery Mobile: It is a subset of jQuery optimized for touch devices.
- jQuery: Is the legendary Javascript library for web technologies.
- JS Editor: There are a lot of free JS editors available out of which Aptana Studio 3 and Brackets are pretty nice.
- Google Chrome: Chrome developer tools provided by the Google Chrome browser are also present in this tools chain. Chrome can be considered the winner in this genre hands down so it will be common in all of our tool chains.

2.3 Tool Chain C

This tool chain is jQuery based with MVC structure. A lot of frameworks are these days built over jQuery. Bootstrap is a material based framework that depends internally on jQuery and provides a set of themes and common widgets out of the box. The app structure is created using Backbone which is well known among the web developer community.

- Twitter Bootstrap
- Backbone.js
- Other JS libraries
- JS Editor
- Google Chrome

Developers can find numerous article on the internet that present comparisons between popular frameworks but the comparison chart in figure 1 is followed by specific developer oriented section. In this section issues with each tool chain while developing is discussed and also use case for each tool chain is proposed. This provides clarity about the usability along with pros and cons.

Table 1 shows the comparison chart of the three tool chains across eight critical parameters. This comparison chart is incomplete without deep dive into the frameworks that each tool chain uses.

Table 1: Tool Chain Comparison

Parameter	Tool Chain A	Tool Chain B	Tool Chain C
Platform Support	-Compatible with WebView Browsers only. -Supports Android and iOS.	-Cross browser compatibility. -Cross platform and cross device compatibility.	-Cross browser compatibility. -Cross platform and cross device compatibility.

Architect-ure	-MVC structure. JavaScript based development. -Multiple JS files with a single HTML loader page.	-Out of the box MVC not supported. -No established architecture. -jQuery based framework. jQuery is markup based.	-MVC structure -jQuery based framework -Backbone based structure.
Learning Curve	-Steep learning curve. -Challenge to developers new to JS.	-Very easy to learn. -Productive from day one.	-Decent learning curve. -Web developers get an upper hand.
Docume--ntation	-Good official documentation. -Bad third party support and forums. -Official support is paid.	-Decent official documentation -Excellent third party information coverage. -Good book coverage.	-Decent official documentation -Not much third party support for Bootstrap. -Decent third party support for Backbone.
Ease of Use	-Very hard to debug. -JS based development can be confusing. Paid IDE.	-Easy syntax -Any JS IDE can be used.	-Easy syntax. -Any JS IDE can be used.
UI Features	-Extremely fast and fluid on mobile devices. -Limited widgets available. -Based on mobile first approach. -Not many third party widgets available.	-Classic web based UI. -Web first approach. -UI is easily extendable. -Large number of third party widgets available.	-Bootstrap has jQuery based Material Design. -Bootstrap is Much easier to customize. -Bootstrap has a good collection of existing themes.
Extendib-ility	-In house app wrapper support. -Support for themes is available. -Paid IDE and official support.	-Cordova as app wrapper. -Support for themes. -Lot of tools options available in the market.	-Cordova as app wrapper. -Support for themes. -Lot of tools options available in the market.

3. FRAMEWORK ANALYSIS

3.1 Sencha Touch 2

“The leading cross-platform mobile web application framework based on HTML5 and JavaScript for creating universal mobile apps.”

Sencha Touch has the most prominent features of Webkit support and Native UI feel for respected platforms. As a developer one spends time in learning this framework but each hour pays off. The downside here is that Sencha Touch has a

limited set of UI widgets that come out of the box. This means that each app built will look similar to the other. Third party widgets are available but they are not optimized and slug on the mobile. As ST2 is company owned product, it falls behind when compared to developer community support for open source technologies like jQuery. Another key point about ST2 is that it does not fall under the category of cross platform, cross device, cross OS version which becomes very critical when it comes to Android.

3.2 jQuery Mobile

“jQuery Mobile is a HTML5-based user interface system designed to make responsive web sites and apps that are accessible on all smartphone, tablet and desktop devices.”

jQuery Mobile has the most prominent features of cross-platform, cross-browser, cross-device support and excellent developer community support. The learning curve is negligible and the learning starts to pay off from the first day itself. Another advantage with jQuery is that today a lot of libraries are compatible and even based on it making the options for the developer even wider. The down side here is that the app looks like a web page. Native look and feel for respective platforms can be built but with difficulties.

3.3 Twitter Bootstrap 3

“Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.”

Bootstrap is a front-end framework relying on a CSS grid system for both responsive and non-responsive look and feel. Internally it depends on jQuery Mobile. The look and feel is improved via cherry picked icons, typography, code, tables, forms, buttons, etc. Just like jQuery Mobile it also includes different UI components, like dropdowns, button(s), button groups, button dropdowns, navs, navbars, breadcrumbs, pagination and many other. And just like jQuery Mobile everything comes packed with jQuery JavaScript framework. Bootstrap is good for quick prototyping although has limited widgets and themes. Every app built using Bootstrap will give a web UI feel and look similar.

3.4 Backbone.js

“Backbone.js gives structure to web applications by providing models with key-value binding and custom events, collections with a rich API of enumerable functions, views with declarative event handling, and connects it all to your existing API over a RESTful JSON interface.”

While Backbone can be integrated with many third-party template engines, the default choice is Underscore templates. Since Underscore is a Backbone dependency and you already have it on your page, you can easily take advantage of its templating engine without adding any additional dependencies for your application. On the downside, the templating engine of Underscore is very basic and you usually have to throw

javascript into the mix. Backbone is good for mobile as it provides light weight MVC structure

4. KEY POINTS

As stated earlier, there is no best tool chain. What is important is that the developer knows what he wants to create and while designing the architecture of the app picks out the tool chain which is best suited. Now let look at the best cases for each of the three tool chains:-

- Tool Chain A

Tool chain A is best when a developer want to build an app with a native like look and feel and is contented with a limited number of widget at disposal. The developer is comfortable with JavaScript and MVC structure and has bucks to spare to purchase Sencha Architect IDE. This tool chain is capable of handling apps with large and complex data.

- Tool Chain B

Tool chain B is best when a developer is creating a light weight app. The UI is web based and the data to dealt with is also small. This tool chain fails when the code of the app becomes large.

- Tool Chain C

Tool chain C is best when a developer is looking to create a MVC based app that has web first approach. This tool chain provides a limited number of themes and UI widgets. Such an app can handle large data and complex code structure as it is MVC based.

5. CONCLUSION

A developer must give high importance of tool chain selection when it comes to hybrid mobile app development. If a developer is looking for a Mobile first approach then he must use Sencha Touch 2 tool chain, if the developer is creating a web first light weight app then jQuery Mobile tool chain is suitable and if a web based MVC mobile app is the goal then Bootstrap and Banckbone.js are the winners

6. ACKNOWLEDGEMENTS

I am thankful to Mr. Pramod Nanadikar for his guidance and support in this research work.