

Comparative study of frameworks for the development of mobile HTML5 applications

Tim Ameye

tim.ameye@student.kuleuven.be

Sander Van Loock

sander.vanloock1@student.kuleuven.be

Abstract

Productivity

Usage

Support

Performance

3 Frameworks

3.1 jQuery Mobile

jQuery Mobile (jQM) is a mobile HTML5 user interface (UI) framework that was announced in 2010 [?]. In November 2011 version 1.0 was released [?] and one year later in October, version 1.2 was released [?]. As at the time of writing, jQM will be releasing version 1.3 very soon [?]. The framework is controlled by the jQuery Project that also manages jQuery Core. The latter is a JavaScript library where jQM is dependent on [?]. jQM is among other things sponsored by Adobe, BlackBerry and Mozilla [?].

Omkadering

Programmeertaal jQM is markup driven framework

Tools Een basis teksteditor voldoet om met jQuery Mobile aan de slag te kunnen. Natuurlijk kan het gemakkelijk zijn om van *integrated development environments* (IDE's) zoals Aptana Studio [?] of WebStorm [?] gebruik te maken, waardoor je handige kenmerken krijgt zoals *code completion*.

Je kan ook gebruiken maken van Codiqua om via *drag-and-drop* UI elementen op je scherm te slepen. Codiqua zal automatisch op de achtergrond de HTML code voorzien [?].

1 Introduction

2 Comparisoncriteria and related work

Many in-depth comparisons of HTML5 mobile frameworks already exists today. However, none of them are scientifically published or use a large proof of concept to validate their comparison. Blog posts like [Sarraf, 2012; Ayuso, 2012; Rozynski, 2011] all have their own criteria and methodology to assess different mobile frameworks. The overall applications of the criteria changes from use case to use case. [Rozynski, 2011] presents the chosen criteria and discusses each for each framework. [Ayuso, 2012] presents a whole bunch of criteria but all of them are discussed at once per framework. Thereafter, advantages and disadvantages are subtracted and proposed to the reader. [Sarraf, 2012] finally, presents their chosen criteria together with a scorecard and explanation of scores per criteria. Each framework gets evaluated based on the scores for each criteria.

All these blog posts compare more than two frameworks where some of them mobile and some are hybrid (see table 2). Other websites like [Bristowe, 2012; Burris,] only focus on 2 mobile HTML5 frameworks while on the other side of the spectrum [Falk, 2011] tries to compare as much as frameworks as possible in a large tabular form.

	Hybrid	Mobile
[Sarraf, 2012]	0	4
[Rozynski, 2011]	2	3
[Ayuso, 2012]*	1**	6

*is still in development

**a combination of Bootstrap, jQuery and Angular JS

As mentioned earlier, we will compare 4 HTML5 mobile frameworks based upon a large-scale proof of concept.

Community

Documentatie Documentatie is te vinden op www.jquerymobile.com/demos/1.2.0. Hierop is een catalogus te vinden van alle mogelijke elementen waarover jQuery Mobile beschikt. Door de broncode van een voorbeeld te bekijken, kan je zien welke code je moet schrijven om tot dat resultaat te komen.

Naast de UI elementen is er ook documentatie over de API. Deze gaat over initiële configuraties, events en methodes die kunnen worden gebruikt.

Marktadoptatie Als we kijken op de website van jQuery Mobile zien we een reeks applicaties gemaakt met hun raamwerk. Enkele voorbeelden zijn webapplicaties voor Ikea, Disney World, Stanford University en Moulin Rouge [?].

Licenties Vanaf september 2012 is het enkel nog mogelijk om jQuery Mobile onder de Massachusetts Institute of Technology (MIT) licentie te verkrijgen [?]. Dit betekent dat de code wordt vrijgegeven als *open-source* en dat deze tegelijkertijd kan worden gebruikt in propriëtaire projecten en applicaties [?].

Code en ontwikkeling

Zoals werd aangehaald, schrijft men vooral HTML5 code voorzien van `data-*` attributen. Daarna zal het raamwerk door middel van *progressive enhancement* allerhande code toevoegen om de beoogde UI elementen correct te tonen in de browser. Dit wordt verder uitgelegd in de sectie browserondersteuning (zie 3.1).

Er zijn drie strategieën om webapplicaties te maken in jQuery Mobile [?]. Een eerste is om de volledige applicatie in één webpagina te schrijven. Met andere woorden, de vele schermen van de webapplicatie zijn dan allemaal samengebracht op eenzelfde webpagina. Het voordeel bij deze aanpak is dat er initieel minder verzoeken zijn naar de server omdat alles in één bestand wordt opgehaald. Dit geldt ook zo voor de geïmporteerde CSS en JavaScript-bestanden.

Een tweede strategie is om voor ieder scherm een aparte webpagina aan te maken. Het voordeel hierbij is dat de eerste pagina waar de gebruiker op terecht komt, sneller wordt gedownload. Bij iedere navigatie naar een ander scherm, moet dit scherm via AJAX worden opgehaald, waardoor dit vertragend kan werken.

Een laatste strategie is om een mix tussen beide te maken. Men kan bijvoorbeeld alle schermen die de gebruiker vaak nodig heeft op één webpagina plaatsen. De schermen die de gebruiker zelden nodig heeft, plaats men dan op aparte webpagina's.

Functionele kenmerken

jQM is a UI framework and thus provides mainly UI components. jQM provides 6 categories of components: pages and

dialogs, toolbars, buttons, content formatting, form elements and lists [?].

Niet-functionele kenmerken

Performantie Zoals gezegd schrijft de ontwikkelaar HTML5 code met specifieke data attributen en zal het raamwerk daarna de code verder aanvullen. Dit gebeurt enkel op de pagina die de gebruiker op dat moment bekijkt. Dit gaat dus ook op voor een webapplicatie waarbij alle schermen op één webpagina zijn geschreven. Deze webpagina bevat allemaal `<div>`-verpakkingen voor ieder scherm. jQuery Mobile zal enkel die `<div>` verder aanvullen die op dat moment getoond wordt aan de gebruiker.

Aanpasbaarheid Als je jQuery Mobile *out-of-the-box* gebruikt, zit alles al goed qua kleur en design. Je hebt de keuze uit vijf kleurenthema's die je kan toepassen op de gehele applicatie of enkel op bepaalde elementen. Om je applicatie echt te laten onderscheiden van de andere, zal je natuurlijk graag je eigen kleurthema willen toepassen. Hier is jQuery Mobile op voorzien door hun *stylesheet* op te delen in twee delen: thema's en structuur. Je kan als ontwikkelaar ook enkel de structuur downloaden en zelf de thema CSS schrijven. Daar dit laatste heel wat inspanning vraagt, hebben de ontwikkelaars van jQuery Mobile ook een tool ter beschikking, namelijk ThemeRoller [?]. Hier kan je zeer eenvoudig kleuren slepen naar een voorbeeldapplicatie. Eenmaal tevreden kan je de overeenkomstige *stylesheet* downloaden en toevoegen aan je project.

Programmeerbaarheid Bij het programmeren in jQuery Mobile wordt geen enkel ontwerppatroon afgedwongen. De code voor de UI elementen wordt tenslotte als HTML5 code geschreven. Voor de echte functionaliteit wordt beroep gedaan op JavaScript en meer bepaald op de jQuery Core bibliotheek. Ook deze dwingt geen ontwerppatroon af.

Browserondersteuning jQuery Mobile deelt browsers op in drie verschillende klassen: A, B en C [?]. Hierbij ondersteunt een klasse A browser alles, terwijl een klasse C browser enkel de basis HTML ondersteunt (en dus bijvoorbeeld geen hippe CCS3 overgangen).

Er dient een onderscheid te worden gemaakt tussen de begrippen *progressive enhancement* en *graceful degradation* [?]. Het eerste is wat jQuery Mobile toepast, namelijk starten met de basis HTML. Deze code wordt door iedere browser, dus ook deze uit de C klasse, op een goede manier weergegeven. Daarna zal het iteratief elementen toevoegen tot het op een moment komt dat de betreffende browser een bepaald kenmerk niet meer ondersteunt.

De tegenhanger is *graceful degradation*. Hierbij wordt eerst een versie ontwikkeld die enkel in de meest recentste browser kan worden getoond. Daarna, als de ontwikkelaar nog tijd heeft, gaat hij *fallbacks* implementeren waardoor minder recente browser de applicatie ook kunnen weergeven.

3.2 Sencha Touch

Sencha Touch is a framework developed by Sencha, a company founded in 2010 as a composition of Ext JS, jQuery Touch and Raphaël. Ext JS is a JavaScript framework for the development of web applications. jQuery Touch is a jQuery plug in for mobile development that adds touch events to jQuery and depends on the WebKit engine. Finally, Raphaël, is a JavaScript library for vector drawings. Pieces of the first two technologies can be found in the implementation of Sencha Touch framework.

Currently, Sencha Touch is at version 2.1.1 [Sencha Inc., 2013a].

Documentation All documentation for Sencha Touch 2.1.1 can be found at docs.sencha.com/touch/2-0. The most important features, are provided with code examples and an example of the code after rendered by the browser. The key concepts of Sencha Touch are explained in extensive tutorials: some texts, some videos.

Another handy tool to discover the Sencha Touch features is the 'Kitchen Sink' [Sencha Inc., 2013b]. This is a web application, written in Sencha Touch, that lines up all possibilities of the framework combined with the corresponding code.

Licenses Sencha Touch is free within a commercial context in which the developer does not share the code with its users. There is also the option to use an open source version. This comes with a GNU GPL v3 license which implies a free code redistribution as most important property. More detailed licenses can be found at [Sencha Inc.,].

Code and development Sencha Touch is written on top of Ext JS, and can also be considered as JavaScript framework. All code needs to be written in JavaScript and loaded by one HTML container. An other important aspect of Sencha Touch is that it supports the Model-View-Controller (MVC) pattern. Models group fields to data-objects, views define how the content is presented to the user and controllers connect these based on events.

Sencha Touch contains all UI-elements as JavaScript objects. Just like object-oriented programming, those objects are part of a class system. Classes can both be defined (`Ext.define`) or created (`Ext.create`). Single-inheritance and overriding is also possible.

To enhance performance, it is the programmers task to create components before they are used. In this manner, programmers can mimic asynchronous loading of pages by creating them in advance.

Browser support Just like jQuery Touch, Sencha Touch is based upon the WebKit browser engine. This forms the major requirement for browser support. Although most mobile

browsers contain this engine, some like FireFox Mobile and Opera Mobile lack behind [John E Clark, 2012]. Following [Wokke, 2013], the next release of the Opera browser will contain this engine, a trend that most browser vendors will (have to) follow.

Sencha Touch offers the programmer methods to ask for the current context where the end-user is working in. Properties like `Ext.env.Browser` and `Ext.env.OS` or methods like `Ext.Viewport.getOrientation` and `Ext.feature.has` can determine this context [John E Clark, 2012]. The latter has functionalities, just like Modernizr [Modernizr, 2012].

3.3 Table

4 Comparison

4.1 Explanation

4.2 Community

4.3 Proof of concept

5 Future work

6 Conclusion

References

- [Ayuso, 2012] Alejandro Ayuso. Mobile Frameworks Comparison, 2012.
- [Bristowe, 2012] John Bristowe. jQuery UI vs Kendo UI, 2012.
- [Burris,] Chip Burris. Sencha Touch vs. JQuery Mobile Cage-Fight REMATCH !
- [Falk, 2011] Markus Falk. Mobile Framework Comparison Chart, 2011.
- [John E Clark, 2012] Bryan P. Johnson John E Clark. *Sencha Touch Mobile JavaScript Framework*. Packt Publishing, 2012.
- [Modernizr, 2012] Modernizr. Modernizr: the feature detection library for HTML5/CSS3, 2012.
- [Rozynski, 2011] Craig Rozynski. Mobile framework SMACKDOWN!, 2011.
- [Sarraf, 2012] Ali Sarraf. Mobile JavaScript frameworks, Evaluation and Comparison, 2012.
- [Sencha Inc.,] Sencha Inc. Sencha Touch Licenses.
- [Sencha Inc., 2013a] Sencha Inc. Sencha, 2013.
- [Sencha Inc., 2013b] Sencha Inc. Sencha Touch Kitchen Sink, 2013.
- [Wokke, 2013] Arnoud Wokke. Opera wordt Chromium-browser met Webkit-engine, 2013.