A short guide to... Mobile Web Design Techniques

A discussion of the various approaches to designing for mobile web



Contents

| Designing for Mobile Web | 3 |
|---------------------------------|----|
| Introduction | 3 |
| Evolution of content adaptation | 3 |
| Responsive Design | 5 |
| Progressive Enhancement | 6 |
| Server-side Adaptation | 7 |
| Hybrid Techniques | 8 |
| Summary of Techniques | 9 |
| For More Information | 9 |
| About dotMobi | 11 |
| Contact information | 12 |



Designing for Mobile Web

Introduction

This paper will help you pick from amongst the many techniques for building a mobile website. It doesn't describe how to do it, rather it instead tries to help you to pick the right approach. Before we begin it's worth clarifying exactly what the goal of the exercise is. Generally speaking, people who are looking to build a mobile site fall into two categories. They're either:

- 1. trying to make an existing website work passably well on mobile devices or,
- 2. building a mobile experience from the ground up

These two goals are quite different and tend to result in different approaches and solutions. The first goal tends to boil down to making a site *resolution independent* such that it works on a reasonably wide range of devices of differing screen sizes, but retaining existing site structure, navigation and use cases. The second aims to build a mobile web experience that caters for a mobile user's use cases (whether they're actually on the move or not) by building a different set of views and interactions with the site.

In order to distinguish between the techniques available this paper will use the terms *resolution independence* and *content adaptation* respectively. The former refers to making your existing site work more flexibly when faced with different resolutions, the latter means creating a *designed-for-mobile experience* with all that that entails.

Evolution of content adaptation

For the first decade or so of the mobile web there was a clear distinction between the mobile web and the desktop web, and there was really only one technique available to make content work well across multiple devices: server-side adapation. This means logic in the server that would detect the device in question and make changes to the delivered content to ensure that it worked. In fact, server-side adaptation was usually necessary to make the content work *at all*—failing to do so would render your content unviewable on many devices. In the last 5 or so years however, things have gotten quite a bit more complex: the capabilities gap between mobile and desktop browsers has now been filled by an array of smartphones of all shapes and sizes and tablets. Even the humble feature phone is inheriting rich browser features as WebKit-based browsers start to show up in lower-end phones. This has led to three outcomes:

 There is no longer a clear distinction between mobile and desktop devices—it is now a smooth continuum from the humble Nokia 1100 all the way to full desktop browsers



- 2. Now that many devices have rich, capable browsers with good JavaScript support, there are new techniques available for adapting content for these new devices
- 3. Some people now question the very need to use content adaptation at all, given that most smartphones can readily display almost any website

Unless you take the stance that your existing desktop-focused site is good enough as is for mobile users, you need to pick a technique for building out a mobile-friendly experience for your website. There are many competing techniques available, all with advantages and disadvantages. This whitepaper will discuss some of these techniques.

Once the overall approach to the site has been decided, the next step is to decide how best to tackle the mobilization aspect of the design i.e. how to make sure that the resulting site delivers the correct experience to users of widely differing devices, in different contexts. Before we begin, it is worth recalling the aims of any such techniques. Any mobile site should be able to deliver two key things:

- A contextually appropriate experience—it should be able to deliver an appropriate experience for someone using a mobile device.¹ The importance of a contextually appropriate experience is increasing rapidly as the number of ways that we interact with the web is increasing: a lean forward experience that seems appropriate on a laptop may feel entirely incorrect on a television browser that you interact with from across a room.
- A device-sensitive experience—it should be capable of delivering an experience that works well on the devices used by your customers. This range of addressable devices is increasing all the time, and growing more diverse, from feature phones to televisions. Some are held close to the face, others are interacted with from across a room. It is next to impossible to deliver a satisfactory experience on such a wide range of devices, each with their own input/output restrictions and conventions, without tailoring the experience to the device. The major internet brands are keenly aware of this and doing much more of it than may be apparent—even the seemingly simple Google homepage masks vastly different code served to different devices used to achieve a useful experience across the device landscape.

Broadly speaking there are two classes of technique used for delivering a mobile-friendly experience, categorized by where the adapting logic runs. Client-based solutions achieve their goals using a combination of CSS and JavaScript running on the device; server-side solutions execute their logic on the server before sending anything to the client. Note: while this paper describes "mobile web" techniques, this advice covers all non-desktop devices, including mobile phones, tablets, gaming consoles, book readers and televisions.

¹ Note that use of a mobile device does not necessarily imply mobility—mobile device users are often physically immobile but users may nonetheless prefer to interact with your site or service in a different way when using a mobile device.



Responsive Design

Responsive design has recently become a very popular technique for making a website work better on multiple device form factors. In fact, fluid design has always been a goal of enlightened web developers. Ethan Marcotte outlined a concrete set of techniques that are readily implementable by most web designers without requiring new tools, hence the appeal of his solution. Responsive design, as originally outlined is based on three core techniques:

- A flexible grid—making sure that the underlying page grid scales with screen resolution rather than using fixed pixel dimensions
- Flexible images—images that work well within a flexible grid
- CSS media queries—using CSS styling tailored to ranges of resolutions or types of device

By using these techniques it is possible to serve a single HTML document to a wide range of devices and expect a reasonable visual result: sites built using this technique can be made to work well on all desktop browsers and most smartphones.

Used as a means to deliver both a desktop and mobile site, however, responsive design falls short on delivering both desired aspects of an ideal mobile site. Firstly, it cannot deliver a contextually appropriate experience because it always delivers the same experience; it can deliver a device-sensitive experience only to a limited range of devices, since the core technique limits the range of devices that can be targeted to smartphones and other high-end devices.

The one-experience-fits-all issue and limited range of addressable devices may not be a problem for all websites—some sites don't lend themselves well to mobile-specific experiences and equally some site owners may not have a desire to serve a wide range of devices. It is worth noting that responsive design has an unknown impact on mobile SEO since it is not clear whether or not search engines will identify the content as being mobile-friendly and rank it accordingly in mobile searches. In summary, the term responsive design is an appropriate label for the technique—it is a set of design principles designed to achieve a degree of resolution independence within a class of devices.

Unfortunately, responsive design often gets confused with building a "proper" mobile website. Developers assume that because they've built a responsive design, their site is now mobilefriendly and they've "done" their mobile site. Granted, building a responsive site is far better than doing nothing at all, but falls short of a true mobile solution that harnesses the power of mobile. Ethan Marcotte himself does not advocate this approach for creating a mobile site and quite sensibly suggests that the best approach depends on the project. From his book:



"But most importantly, responsive web design isn't intended to serve as a replacement for mobile web sites. Responsive design is, I believe, one part design philosophy, one part front-end development strategy. And as a development strategy, it's meant to be evaluated to see if it meets the needs of the project you're working on."

Progressive Enhancement

Progressive enhancement (PE) is a newly popular entrant to the content adaptation space, but was first conceived almost a decade ago by Steven Champeon and Nick Finck in their paper Inclusive Web Design for the Future, presented at SXSW. The idea behind progressive enhancement is to serve a single base HTML page to every device, but with linked JavaScript functionality that builds up the experience as appropriate. If the device is very rudimentary, it will ignore or fail to run the JavaScript and the user is left with a reasonable low-end experience; if the device is a smartphone or desktop browser the JavaScript code will add functionality progressively to the page, until it is built up to the optimal level for the device in question. In theory there is no upper bound to the richness that can be layered onto the base page, and you can scale page richness smoothly from feature phone to desktop browser.

The appeal of PE is obvious: it can cater for the full spectrum of devices—low level devices are well catered for (it is a fail-safe solution); more capable devices are not limited by a lowestcommon-denominator experience. PE is the approach used by the recently-released jQuery Mobile library, of which dotMobi is a sponsor. In effect, PE moves the logic of content adaptation from the server to the client.

As a solution to building a dedicated mobile site PE offers reasonable scope to lighten or richen the experience according to the device capabilities with the caveat that the progressive build-up that is at the core of this technique necessarily takes time to execute, and this delay is very much dependent on the device in question, and possibly the network. As an example, certain device models may support the requisite JavaScript in theory, but in practice run it too slowly to be useful.

As a solution to building a website that works for both desktop and mobile devices, PE suffers from the same problem as responsive design: it cannot deliver a contextually appropriate experience to the user since it relies on delivering a single HTML document to all users and devices. This core issue contrains PE's usefulness to a solution to achieving device-sensitivity within a mobile site.



Server-side Adaptation

Server-side adaptation is a technique that has been in use since the dawn of the mobile web, over twelve years ago. It relies on a device detection library or database installed on the web server (or a remote web service) to detect the device accessing the web site and return its capabilities. This set of capabilities allows the web developer to fine-tune the delivered page to match the device's capabilities and the user's context with a very high level of control.

Despite the claims of its detractors, device detection is extremely reliable and accurate, with good solutions typically reporting in excess of 99.5% accuracy in detecting devices in the wild.

The effectiveness of his technique speaks for itself: it is still by far the most common content adaptation technique and is used by almost every major internet brand that takes its mobile presence seriously, including Google, Facebook, Amazon, Youtube, EBay and Yahoo and more than 80% of the Alexa top 100 websites.

The reason service-side adaptation is so popular is because it does such good job of delivering on the two requirements outlined:

- It allows for full control over the HTML delivered in response to a request, thereby allowing you to deliver a contextually appropriate experience to any user or device, be it mobile phone, tablet, TV or desktop. The only limit is how much you wish to fine-tune the approach.
- At allows for a fully device sensitive experience to be delivered, with no execution delays on the client. Again, this approach is limited only by how far you wish to take it.

Server-side adaptation is not without its problems, however. There are two main issues:

- It requires the web developer to use a device detection solution, most of which are now commercially licensed. These databases need to be kept up to date to ensure that they kept abreast with new devices released to the market.
- Developers may still need to utilize real-time information from the user's browser to help you better serve the context of the user e.g. using GPS coordinates or device orientation to better tailor the information that you serve to the user.

Browser detection is sometimes perceived poorly due to its early use as a way to work around inconsistent browser implementations. Some of these limitations are easily mitigated by choosing the correct solution; others are addressed by hybrid solutions outlined in the following sections.



Hybrid Techniques

In addition to the three techniques outlined in the previous sections there are many variants of these techniques being used experimentally, including some where approaches are combined to yield the best of both. The following are some of the more important approaches being trialed.

Mixed Client/Server

This approach combines elements of server-side adaptation and progressive enhancement. In essence, this approach works by delivering an initial page based on server-side adaptation principles but then enhances the result by capturing device properties via client-side JavaScript, and using the captured information to fine-tune subsequent pages delivered to the device. This hybrid approach is probably the best of both worlds—you get the benefits of highspeed server-side adaptation, combined with the ability to fine-tune the results based on properties sourced from the device itself. The user gets an initial page that is well-suited to the device, with no performance overhead, and subsequent visits to pages on the site may improve on this experience. There are two downsides, however.

- It is relatively complex to implement.
- On first visit, a full round-trip is required between the server and the browser before you get to benefit from the properties sourced from the browser. This delay can be removed on subsequent requests by using cookies to recall data from previous sessions.

Responsive design can also be used in conjunction with server-side device detection to allow devices to be categorized and catered for broadly on the server, and then in finer detail with client-side logic and media queries.

Mobile-First Responsive Design

A number of people have pointed out that responsive design may make more sense if used in an inverted manner: if you design your site such that the default rendering of a page is mobile-friendly some of the issues with responsive design appear to go away. In particular, the issue with unnecessarily large images being downloaded by mobile devices can be solved with this approach. Current best practice with this variant of the technique is to initially serve mobile-friendly images to all devices and then, browser willing, swop them out one by one for desktop-sized images.

A side benefit of mobile-first design is that it can act as a "wedge" to help designers make the case for removing unnecessary clutter that invariably accumulates on desktop sites over the years, since the mobile-first design forces this approach.

Mobile-first responsive design is a compelling update to the original set of techniques, but not without its problems:



- Again, it achieves only resolution independence, does not facilitate content adaptation
- It requires that the desktop site be redesigned from scratch, although this could be argued as a good thing.
- It may force you to constrain the interactions of desktop users

In summary, if your goal is to create a mobile website, mobile-first progressive design is the only really useful variant of responsive design, since it is truly able to scale from low-end devices all the way to desktop browsers.

Summary of Techniques

Having looked at all of the techniques available, how do you choose between them? Overall, of course, the answer is "it depends". That said, any technique based on the premise of using a single HTML document to address all devices is fundamentally flawed for the same reason that most television content is not just re-purposed movies, and most websites are not pixel-perfect copies of paper publications. One can argue that where there are limited use cases for interacting with a site, such as a blog, a single set of interactions may suffice across both desktop and mobile uses. But in the more general case this seems like a serious limitation at best, and a lost opportunity at worst. If all you want to do is make sure that your site works on a few high end mobile devices, and you don't care particularly about catering for the mobile web user, try the responsive design approach, or the mobile-first responsive design. If the use cases for your site are limited this might actually work quite well. If, on the other hand, you want to deliver a full designed-for-mobile experience or you want to cater for all mobile devices out there and not just smartphones, you don't really have a choice: server-side adaptation or a hybrid approach is the only solution that will work. There is a reason why essentially all of the top internet brands use this approach.

For More Information

Contact us at sales@deviceatlas.com or visit www.deviceatlas.com

See also:

http://mobithinking.com/guide-device-detection

http://mobiforge.com/starting/story/mobile-web-content-adaptation-techniques

http://mobiforge.com/designing/story/anatomy-a-mobile-web-experience-google-com

http://mobiforge.com/designing/blog/anatomy-a-mobile-web-experience-facebook-com

http://mobiforge.com/designing/blog/server-side-device-detection-used-82-alexa-top-100-sites



Summary of Techniques 10



About dotMobi

dotMobi is a leading mobile Internet services company, helping businesses and individuals reach the world's billions of mobile phone users. dotMobi spurs mobile industry innovation by giving content providers the tools they need, to ensure the web will work on mobile phones with speed, accuracy and relevant content.

dotMobi, a wholly owned subsidiary of <u>Afilias</u>, was founded by leading mobile operators, network & device manufacturers, and Internet content providers, including Ericsson, Google, GSM Association, Hutchison 3, Microsoft, Nokia, Orascom Telecom, Samsung Electronics, Syniverse, T-Mobile, Telefónica Móviles, Telecom Italia Mobile (TIM), Visa and Vodafone.

| Device Atlas. | DeviceAtlas is one of the largest open repositories of mobile device profiles, based on W3C recommendations. It provides the supporting tools, techniques and assistance that you need to take that data and use it to rock your mobile users' world. <u>www.deviceatlas.com</u> |
|------------------------|--|
| BoMobi | goMobi™ is the world's first content mobilization platform, a hybrid of a traditional content management system and a practically automatic mobile website builder. It's the smart, simple way for businesses and designers & developers to build a mobile Web presence. www.gomobi.info |
| ⊠mobi Thinking. | mobiThinking™ is here to help you market your mobile site. It's packed with insight, analysis and opinions from the world's mobile marketing gurus. <u>www.mobithinking.com</u> |
| ∑mobi Forge. | mobiForge™ is the dotMobi developer forum a center for mobile Web developer tools, resources and support. More than 50,000 developers and designers meet here to compare notes, share tips, upload ideas and download expertise. <u>www.mobiforge.com</u> |
| mobiReady | mobiReady™ evaluates mobile-readiness using industry best practices and standards. Test your mobile website and get a free report plus in- depth analysis to determine how well your site performs on a mobile device. <u>ready.mobi</u> |



Contact information

| dotMobi | Email: | <u>sales@deviceatlas.com</u> |
|-------------------|---------|------------------------------|
| 2 La Touche House | | <u>contact@mtld.mobi</u> |
| IFSC | | |
| Dublin 1 | Phone | +353 1 854 1100 |
| Ireland | r none. | +333.1.034.1100 |
| | Fax: | +353.1.791.8569 |
| | | |

