

Tips and Tricks about Plone Development for Mobile Platforms

Plone Conference 2012 – Arnhem
Fabiano Weimar dos Santos [Xiru]

Speaker

- I'm from Brazil ;)
- Contributed some lines to Archetypes and ArchGenXML (many time ago)
- Msc Computer Science (Artificial Intelligence)
- Consultant, Instructor and Sysadmin
- Plone pays my bills.

What this talk will cover

- **How Plone could be used to publish content on Mobile Platforms**
- Mobile Web Development Best Practices
- HTML 5 is the way?
- When native apps are really needed

Background Work...

- SERPRO has been concerned about mobile development
- Lots of Plone portals running on Brazilian government and no mobile friendly content
- Concerns about the right way to do it
- Fact: Plone portals usually deliver lots of information but on mobile that information are not usable



SERPRO

Desde 1964

Soluções que nos conectam

[a empresa](#)

[estrutura](#)

[transparência](#)

[ética](#)

[equidade de gênero](#)

[concurso público](#)

[acesso à informação](#)

Central de Serviços ▲

A Central de Serviços Serpro (CSS) é o canal de comunicação mais ágil entre os clientes da empresa ou usuários dos serviços Serpro com as equipes de especialistas. Confira as formas de abrir um acionamento na central e compreenda o seu funcionamento.

[+ saiba mais](#)



[Entre em contato com a CSS](#)

Notícias ▲



Processo Verde usa tecnologia livre

Ferramenta eliminará uso de papel no processo de compras corporativas

Certificação ▲

O Serpro é uma das quatro autoridades certificadoras do país. Confira mais informações sobre Certificação Digital

[+ saiba mais](#)



Tecnologia

Let's wake up

“Brazil is going to have 1 billion mobile devices with Internet access in 2020”

Paulo Bernardo, 08/08/2012
Brazilian Minister of Communications

And...

In June of 2012, Brazil had 46.6 million smartphones and tablets with 3G connection.

Agencia Brazil
08/08/2012

The population of Brazil was 190,732,694 people (2010 census)

Everybody is going mobile.

What about Plone?

On mobile...

- Information needs to be easy to access and understand
- Data reliability and security is very important, specially over bad networks
- Speed is not everything. Latency is also important.
- Mobile technology is still expensive

General Tips

- Duo limitations of the mobile devices
 - Simple web applications are better
 - Tip: keep DOM small and limit the number of CSS selectors you use
- Use fewer HTTP requests (the latency monster)
- Be careful with JavaScript code

Mobile Web Best Practices

- W3C Mobile Web Best Practices 1.0
 - 2008
 - 60 practices
- W3C Mobile Web Application Best Practices
 - 2010
 - 32 practices
- Have you read it?

The Most Important Aspects

- Portability
- Usability
- Accessibility
- Security
- Adaptation Effort
- The Learning Curve
- Observe Limitations (memory, processor, network speed and latency)

Secure Mobile Web Development

- OWASP - The Open Web Application Security Project - has interesting documents about security!
- OWASP Top 10 (security risks)
 - Plone is going great here (thanks security team)
- New: Top Ten Mobile Controls

Top 10 Mobile Controls

- Identify and protect sensitive data on the mobile device
- Handle password credentials securely on the device
- Ensure sensitive data is protected in transit
- Implement user authentication, authorization and session management correctly
- Keep the backend APIs (services) and the platform (server) secure

Top 10 Mobile Controls

- Secure data integration with third party services and applications
- Pay specific attention to the collection and storage of consent for the collection and use of the user's data
- Implement controls to prevent unauthorized access to paid-for resources
- Ensure secure distribution/provisioning of mobile applications
- Carefully check any runtime interpretation of code for errors

OK, let's suppose I'm doing everything right

- You have 3 choices (not mutually exclusive)
 - Responsive Web Design (the Plone way)
 - HTML 5 code that mimics native interfaces
 - Native Apps

Native development could be avoided

- It demands experts on several languages and platforms because the same app needs to be developed for more than one kind of mobile device
 - It is complex
 - It costs money
 - It takes time
- But, off course, native Apps still have its advantages...

Native Apps could be used

- When you need absolute performance
- When you must exploit toolkit features for better “User Experience”
- You already tried to deliver your data using a web based App and it was not enough (e.g. Facebook)

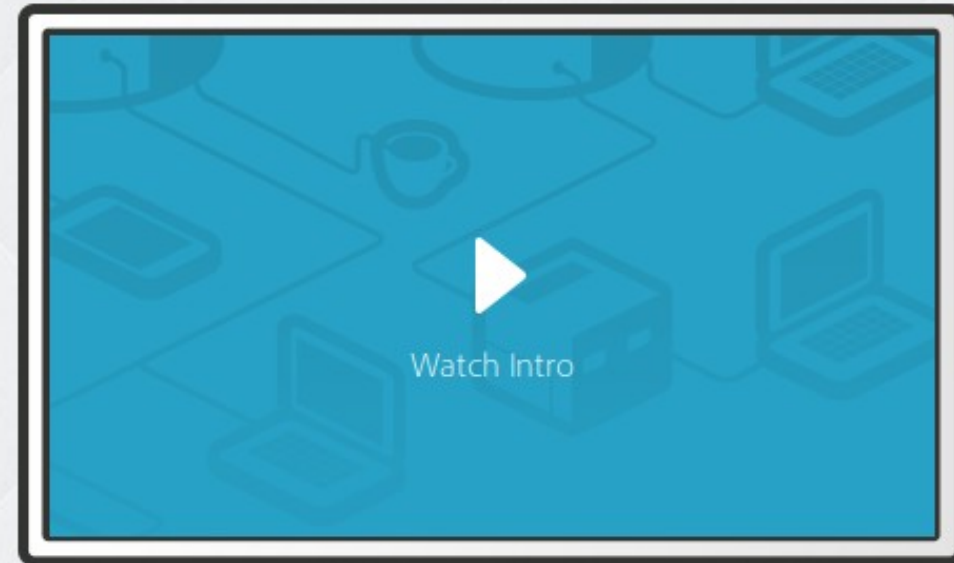
HTML 5 is the way to go?

- For most people, yes
- The final version of HTML 5 is on W3C plans for 2014 ?!
- Plone 4.2 is HTML 5 ready!
- **PhoneGap: interesting approach to create native Apps using HTML 5, CSS 3 and JavaScript**



Easily create apps with the only free **open source** framework that supports **7 mobile platforms**.

Developing with **PhoneGap** gives you the freedom to create mobile applications for **iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada** and **Symbian** using the web code you know and love: **HTML, CSS** and **Javascript**.



[Download PhoneGap](#) ▶

[Getting Started Guides](#) ▶



How Plone can deliver content to mobile?

- Responsive Web Design
 - Plone 4.2 already do that
 - Tip: do not create Diazo themes with problems that Plone already solved
- Firefox 15 shows it is practice!

Content Adaptation

- Going mobile is not making miniatures of your website, but adapt the content to a different media and audience
- Multimedia content (images, audio and video) can be trick
 - Tip: do not try to create different versions of software for different devices. Try to follow open formats and official recommendations.

Content Adaptation

- Flash must DIE!
- To detect features of HTML 5 and CSS 3 browsers use Modernizr:
 - “Modernizr is a JavaScript library that detects HTML5 and CSS3 features in the user’s browser”
 - Useful to write portable HTML 5 code
 - Plone already uses it!

“An indispensable tool.”

— Bruce Bowman, *Adobe BrowserLab Product Manager*

Modernizr is a JavaScript library that detects HTML5 and CSS3 features in the user's browser.

Why use Modernizr?

Taking advantage of cool new web technologies is great fun, until you have to support browsers that lag behind. Modernizr makes it easy for you to write conditional JavaScript *and* CSS to handle each situation, whether a browser supports a feature or not. It's perfect for doing progressive enhancement easily.

How it works

Modernizr runs quickly on page load to detect features; it then creates a JavaScript object with the results, and adds classes to the `html` element for you to key your CSS on. Modernizr supports dozens of tests, and optionally includes [YepNope.js](#) for conditional loading of external `.js` and `.css` resources.

Check out the [full list of features](#) that Modernizr detects, or learn more about [conditional resource loading with Modernizr](#).

Download Modernizr 2.6.2

Use the commented, uncompressed Development version to develop with and learn from.

Then, dive into the Production build tool and pick just the tests you need!

[View documentation](#)

 **DEVELOPMENT**
Uncompressed, 42 Kb

 **PRODUCTION**
Configure Your Build

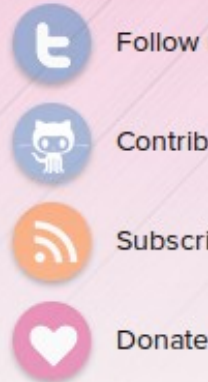
Get started with Modernizr

While Modernizr gives you finer control over the experience through JavaScript-driven feature detection, it is important to continue to use best practices throughout your development process. Use progressive enhancement wherever you can, and don't sacrifice accessibility for convenience or performance.

- [Documentation: Getting started](#)
- [Taking Advantage of HTML5 and CSS3 with Modernizr](#), Faruk Ateş
- [How to use Modernizr](#), Inayalli de León
- [Modernizr: front-end development done right](#), Ryan Seddon
- [wiki] [The Undetectables: features that cannot be detected](#)
- [wiki] [Cross-browser Polyfills](#)

Also check out our [Resources section](#).

Tip: check our [Modernizr test suite](#) to quickly test your current browser's



Latest news

September 4th, 2012
[Modernizr 2.6.2](#)
Modernizr 2.6.2 small updates.

PREVIOUS

July 19th, 2012
[Modernizr 2.6.1](#)

February 6th, 2012
[Modernizr 2.5: 2012](#)

November 7th, 2011
[How Does Modernizr Work?](#)

CSS Media Queries

- The final W3C recommendation is recent
 - W3C Recommendation 19 June 2012
 - @media screen and (max-device-width: 810px)
 - @media (min-width:500px)
 - @media (orientation: portrait)
- Tip: Plone is using it. Let's start to use it more (specially on Diazo themes).

How about Plone Portals that looks like a native App?

- **jQuery Mobile**

- Can be used to develop responsive versions of your themes for mobile devices
- Smart choice to create native Apps using HTML 5, CSS 3 and JavaScript and PhoneGap
- Hint: forms are much better on real mobile devices while using the jQuery Mobile CSS
- Hint: jQuery Mobile ThemeRoller is great!



JQUERY MOBILE 1.2.0 FINAL RELEASED!

jQuery Mobile: Touch-Optimized Web Framework for Smartphones & Tablets

A unified, HTML5-based user interface system for all popular mobile device platforms, built on the rock-solid jQuery and jQuery UI foundation. Its lightweight code is built with progressive enhancement, and has a flexible, easily themeable design.

Latest stable version - 1.2.0

Legacy versions: 1.1.1 - 1.0.1



Seriously cross-platform with HTML5

jQuery mobile framework takes the "write less, do more" mantra to the next level: Instead of writing unique apps for each mobile device or OS, the jQuery mobile framework allows you to design a single highly-branded web site or application that will work on all popular smartphone, tablet, and desktop platforms. Device support



▼ Header/Footer Bar

TEXT COLOR #3e3e3e

TEXT SHADOW 0 1px 1px #ffffff

BACKGROUND #eeeeee

BORDER #b3b3b3

▶ Content Body

▶ Button: Normal

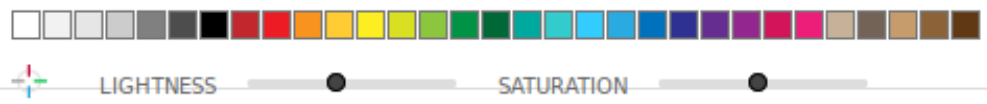
▶ Button: Hover

▶ Button: Pressed

Drag a color onto an element below or pick from the swatches >

Adobe Kuler

Recent Colors [colors...](#)



A

Sample text and [links](#).

List Header

Radio 1

Radio 2

Checkbox

On Off

Option 1

Text Input

50

Button

B

Sample text and [links](#).

List Header

Radio 1

Radio 2

Checkbox

On Off

Option 1

Text Input

50

Button

Sample text and [links](#).

List Header

Radio 1

Radio 2

Checkbox

On Off

Option 1

Text Input

50

Button

Thank you

Fabiano Weimar dos Santos

xiru@xiru.org

IRC: [Xiru]

@xiru

github.com/xiru