

DOTNETNUKE



MOBILE STRATEGIES

Party in Wallisellen, Switzerland, Okt 2012

Presented by

DotNetNuke User Group Switzerland

Daniel Mettler / Raphael Müller (2sic.com)



Mobile Strategies - Rebriefing

What to expect & what's changed

Agenda

- Background: History / Purpose
- Goals: Mobile with DotNetNuke
 - Why do we do it?
 - What should DNN do for us?
 - What should we do ourselves?
- Different Scenarios and how to solve them
- Recommendations

Background - Purpose

Provide optimal information / tools...

...in every usage scenario

→warning! this is mainly a conceptual issue, not a technical one, for eg. see Language-Detection, typical needs of mobile users, etc.

Background - Purpose - Implementation

1. Know if the user is mobile, tablet, small-screen etc.
2. Change what the user receives...
3. ...based on our idea of ideal information/tools

Ergo

1. Detect
2. Redirect...
3. or Reformat (same info, different display)
4. or Reinform (different info)

Background - History

- DNN 3, 4, 5 - DIY (Do it yourself)
- DNN 6.1
 - Open Source API / Framework from 51degrees.mobi
 - Devices-Data from WURFL
 - → Problem: License changed by ScientiaMobile
- DNN 6.2 & 7.0
 - API & Data from 51degrees.mobi
 - Light-data included in community
 - Good data for 60.-/Year/Site
 - Great data for 360.-/Year/Site or included in Pro-Edition

note: more Links in slide notes

Output Variations

Four Output Adaptions

1. Responsive Layout
→ www.2inspire.ch, boxes and fonts change on screen-resize
2. Responsive CMS-Page
 1. requires responsive Modules
 2. Razor-Templates
 3. 2SexyContent
 4. Responsive ASCX-Skins
→ anybody from the audience? simple demo <http://mobile.dnn.ch>
3. Standalone Mini-Site
→ www.dnn.ch/m , www.2sic.com/m
4. Standalone Mini-Site with Content-/Module-Sharing
→ m.kenda-dental.com/ (shared catalog-data from SharePoint)

Output Adaptions - Responsive

1. Change Output / Layout
 1. alternative CSS and/or HTML
 2. Alternative Layout-Image-Resolutions
 3. → Method: CSS, Different Skins, Smart Skins (w/Logic)
2. Change Content
 1. Content in many Variations (short/long texts, etc.)
 2. Different content-presentations
 3. Images & Resolutions
 4. Videos & Resolutions (YouTube, ...)
 5. → no standard available. use Modules supporting multiple outputs or Razor-Templates (like 2SexyContent), or use JS-tricks to automatically use other images etc.

Output Adaptions - Mini-Sites

1. Very simple
2. Usually very reduced content (easy to maintain)
3. Usually specialized content (e.g. Maps links)

→ Low effort, low maintainance, elegant

1. Optionally with User-Sharing, Module-Sharing, Content-Sharing (Pro-Edition or manual work)

Output Adaptions - Redirects

1. Demo w/DNN 7.0
2. Redirect-Settings
3. `nomo=1`, `nomo=0`

→ Low effort, low maintainance, elegant

We would recommend using at least the simple data-subscription.

Output Adaptions - about DDRs

1. DDR – Device Description Repository
2. Light-Version in DLL
3. Simple Data (60.-/year)
4. Pro-Data (~360.-/Year) w/Auto-Update
5. Can be ordered in the DNN Store

→ DotNetNuke Professional has the Pro-Subscription

User Input

Input-Adaptions

Warning: We believe that Input on mobile devices is not fun for the user

1. Recommendation: minimize input at all cost...if not a primary feature of your site
2. ...also makes development easier: leave most input-scenarios to non-mobile devices
3. If really, really needed, spend a lot of time optimizing

→ Suggestion: use QR-Codes to minimize input

Closing Recommendations

Recommendations

1. In 90% of all companies, we believe the best solution is a Mini-Mobile-Site, with Detect/Redirect (Content-Sharing?)
2. Use simple responsive Design on really simple, single-purpose sites like Micro-Sites, Product-Sites
→ and when responsive, KISS – maybe specials for Tablets/TVs, and still redirect for Mobile
3. Full Responsive-Design & Layout ONLY if you have very data-intens sites (many contributes in similar formats), and if you have a huge budget - call me 😊

Thanks!

More infos and this presentation...

mobile.dnn.ch

and www.yaml.de - Perfect CSS-Framework