

Single Page Web Apps

Get your buzzword-bingo sheets out :P

Jan Monschke
Eray Basar

#!/bio

- Jan Monschke
 - Coffeescript / [brunch](#) / CouchApps / [Brunch-Colors](#)
 - [@thedeftone](#) / <http://github.com/janmonschke>
- Eray Basar
 - [@toshiyori](#)
- [@9elements](#) : @imgly, @watchlaterapp

#!/index

- The big picture
- What the #! is a Single Page Web App?
- Why would I do that?
- Why wouldn't I do that?
- What tools are there?

The big picture: Apple's ecosystem



Is the web the saviour?



A black and white engraving of Christ on the cross. He is shown from the waist up, wearing a loincloth and a shawl. He has a crown of thorns on his head and is looking down. Two hands are visible at the bottom left, reaching up towards him. Orange text and arrows are overlaid on the image:

- "CSS3" with a right-pointing arrow to the left side of Christ's chest.
- "HTML5" with a right-pointing arrow to the right side of Christ's chest.
- "JavaScript" with a left-pointing arrow to the top of Christ's head.
- "REST" with a left-pointing arrow to the middle of Christ's back.

Is the web the saviour?

- From a developer's perspective
 - Solve distribution
 - Get low tech API access (Device API)
- From a users's perspective
 - Harmonize UI concepts / overall UX
 - Web apps have to feel more **native**

Web Apps have to feel more native

- Desktop- and Web-Experience converge!



<http://www.yuiblog.com/blog/2009/10/13/video-neuberg-html5/>

Desktop- and Web-Experience converge

- Then:
 - Flash
- Now:
 - JavaScript
 - AJAX
 - JavaScript is everywhere (Frontend, Backend, DB)
 - Single Page Web Apps

#!/what

- <http://twitter.com/> / <http://facebook.com>
- Your whole app exists on a single page
 - No complete reload of the DOM
 - Only parts of your page change
- In-App URLs: mydomain.tld/#!/test
 - bookmarkable
 - client-side routing

#!/why

- ➊ Server doesn't need to render complete layouts
 - ➋ Less CPU time needed for each request
 - ➋ Server is able to handle more requests
- ➋ No huge HTML files are transported
 - ➋ Less data → less time for transport → Good for UX and Backend
 - ➋ Especially good for devices that are connected to mobile networks

#!/why/2

- Transitions between states
 - Immediate Feedback and Interaction Response
 - Elegant way to lead the user through complex processes
- Web is not silent
 - Reintroduction of music experience through-out the Web
 - Separate information flows are not interrupted (chat, music player, file transfers...)

#!/why/not

- Existing page with a good pagerank, hundreds of indexed urls
 - gawker.com → All links from Google were broken
- Intensive Data
 - You shouldn't develop a banking page like that
 - Inspect Element / See source → Your business logic exposed to the user
- Your page relies on the JavaScript performance of your clients
 - bad UX with slow browsers
 - Make sure JavaScript is enabled (<noscript/> ftw!)

#!/tools

- Backbone.js - <http://documentcloud.github.com/backbone/>
 - MVC – in JavaScript
 - Easy Backend integration
- SammyJS - <http://sammyjs.org/>
- brunch - <https://github.com/brunch/brunch>
 - Toolchain for Single Page Web Apps
 - Clean project structure → automatic builds
 - Buzzword mayhem: CoffeeScript / Backbone.js / Eco / Stylus

#!/hosting

- Easily integration into all REST backends
 - Rails / AppEngine / you name it
- The missing link for nodejs?
 - There's no Rails for nodejs (yet)
 - Single Page Web Apps + nodejs = Awesomeness

#!/thx

Any questions?

Feel free to contact us via Twitter:

@thedeftone @toshiyori