

Switch from iUI to Emy

Emy Documentation

This guide's main goal is to help developers to switch from iUI to Emy. It will show and explain the main differences between those two libraries. If you've never heard or used [iUI](#) before, it's really not necessary for you to read this, the [Getting started guide](#) might be a lot more interesting.

Main differences with iUI

We are in 2013 (or perhaps even more at the time you read this), not in 2007 anymore. Modern browsers supports HTML5 and CSS3 pretty nicely, and the iPhone is not anymore the sheriff in town. We now have smartphones, [phablets](#) and tablets, running iOS, Android, Blackberry QNX, WindowsPhone. Some new players might hit our stores very soon ([Firefox OS](#), [Ubuntu Phone](#), [Samsung Tizen](#), ...). Some say TV could also be the next big screen. With that in mind, we have no choice but to support those new platforms & browsers using their advantages & new functionalities.

So the first main big difference with iUI is that Emy fully supports styling, AJAX & CSS transitions on iOS (4 and above), Android (2.3 and above), Blackberry (6 and above) & WindowsPhone (8 and above). It also works just fine on Firefox OS.

Second huge difference with iUI is that Emy is using HTML5 elements rather than DIVs to define each views. The main toolbar is a Header element, each views are Sections. Then you'll find good ol' UL, FORM & FIELDSET elements in them. But we do not use DIV, UL or FORM as views: it's always a Section.

iUI

```
<ul id="home" title="Home Screen" selected="true">
  <li><a href="#page1">Go to page 1</a></li>
  <li><a href="#contact">Go to contact form</a></li>
  <li>Alternate code here...</li>
</ul>

<form id="contact" title="Contact" name="myform" action="/submit">
  ...
</form>
```

Emy

```
<section id="home" data-title="Home Screen" selected="true">
  <ul>
    <li><a href="#view1">Go to view 1</a></li>
    <li><a href="#contact">Go to contact form</a></li>
    <li>Alternate code here...</li>
  </ul>
</section>

<section id="contact" data-title="Contact">
  <form name="myform" action="/submit">
    ...
  </form>
</section>
```

One other big change is that we talk about Views rather than Pages. This is a not an e-book library so why should we talk about Pages right? That might sounds like a small difference, but in fact, it means all Javascript functions you used to know in iUI have been renamed from xxxPages to xxxViews.

Yes, you got it, it means we are 100% NOT compatible with existing iUI code, themes & plugins. We did port pretty much all of them by the way... so it's no big deal.

Emy is using CSS animations to perform transitions between two views, rather than CSS transitions like in iUI. This mean Javascript doesn't

get involved in transitions anymore. It also means you can create your own additional transition styles to use between two views ([extended transitions](#) extension is a good example: vertical, opacity, 3D, swip, rotation, ...) . Yep, it rocks :)

Last but not least, events callback. If you ever try to create something more than a testing demo with iUI, you probably quickly face the problem of triggering a specific function or code when a new page shows up (aka Views in Emy). iUI event-log extension is a great help to do so, but you have to define everything in it or create some code by yourself based on attributes. For non-javascript savvy people, that was very obscure. Dry your tears my friend, data-onshow & data-onexit built-in attributes are here for that now!

```
<section id="home" data-title="Home Screen" selected="true">
  <ul>
    <li><a href="#view1">Go to view 1</a></li>
    <li><a href="#view2">Go to view 2</a></li>
  </ul>
</section>

<section id="view1" data-title="Page 1" data-onshow="myFunctionNameOnShow">
  <p>View 1</p>
</section>

<section id="view2" data-title="Page 2" data-onexit="myFunctionNameOnExit">
  <p>View 2</p>
</section>
```

Some other small changes or new things

We did remove orientation class & attribute on the body element. The exact same info can be set using [MediaQueries](#) so there is no need of Javascript to do it.

If browser supports it, we do use the [hashChange event listener](#) rather than a setInterval.

title="..." is now data-title="..." to avoid trackball/mouse showing a bubble next to the pointer.

Form element toggle updates an input field and does not use Javascript anymore, which means it's taken in account directly when you submit the form.

Since it's very convenient, we added [a basic \\$\(\) selector](#) to select an element by its ID, or a list of elements by classname or tagname. Nothing more than that (if you need more, JQuery-compatible [Zepto JS](#) are doing the job fine).

Each View is minimum 100% height of the browser's viewport. In other words, the background color or image fills the all viewport even if its content is too small for that. This is called "fitHeight".

Dialog views come with an opening animation.

And a lot more you will love to discover...

More resources

[Core javascript object documentation](#)

If you want to learn all nitty-gritty details about Emy's javascript code.

[Emy's CSS documentation](#)

For designers, here is the complete guide of default elements and how we style them.

[Emy community on Google+](#)

Official community's talks, links, events & discussions about Emy

[iPhoneWebDev Google Group / mailing-list](#)

Last but not least, you may find some very useful support from this worldwide community

[Follow Emy on Facebook](#)

Like the Emy Facebook page to get latest images, links & news on the biggest network worldwide

Why Emy instead of iUI 2?

Since late 2009, i did contribute to iUI in many ways: themes, plugins, bugfixes, design & development of its website, documentation, ... but every time, for each single step, i did face some resistance to bring some new ideas & improvements to iUI's main core files. For example, v0.5

was pushed to discuss more than a year before 0.4.3 finally got the new official latest release. Don't get me wrong, I have no anger about that, neither personal recognition issue of any kind, this is a very common problem with OSS projects: asking or waiting for too much feedback before doing anything.

As a result, the iUI project was moving too slow. I did try for more than a year to get the rest of the core team to involve a bit more, or let others do the job. I even redone some (very boring) code cleaning / refactoring job more than four times, to never see it accepted because of weird commit policy. Being stiff is ok when you have to deal with hundreds of commits per day (Webkit, Android, JQuery, ...), not when you are 3 guys posting 15 commits a year. Too slow, too rigid.

The other main reason is more personal. Since mobile web is gaining momentum, my situation has morphed from a "mobile web job seeker spending spare times to contribute to an open source project" to a "self-employed mobile web consultancy & development company who needs a living & up-to-date codebase". In other words: companies are contacting me to implement, customize, improve or mobilize their in-house platforms. So, for my business sake, i needed to show and use a moving-fast up-to-date library compatible with nowadays mobile browsers (different Webkit implementations, Firefox, IE10, ...). No significant official updates for more than 18 months is not acceptable, and i find myself using my own non-backward-compatible fork each time.

Those two reasons are why i finally decide to split my code from the iUI main branch & repo for good, and to move forward with Emy.