

Core Documentation (JS)

Emy Documentation

[Getting Started](#)[Core documentation \(JS\)](#)[Style documentation \(CSS\)](#)[Community support](#)

*For a gentler introduction to Emy, please check out the [Getting Started](#) documentation first.
If you already know and use [iUI](#), our guide [Switch to Emy from iUI](#) might help you a lot.*

Events

[onLoad](#)[onClick](#)

Emy events

[blur / focus](#)[load / unload](#)[beforeinsert / afterinsert](#)[beforetransition / aftertransition](#)

Properties

[emy.logging](#)[emy.busy](#)[emy.transitionMode](#)[emy.httpHeaders](#)[emy.prefixedProperty](#)

Methods

Navigation methods

[emy.isNativeUrl\(\)](#)[emy.goBack\(\)](#)[emy.gotoView\(\)](#)[emy.showView\(\)](#)[emy.showViewById\(\)](#)

[emy.showViewByIdHref\(\)](#)

Views methods

[emy.ajax\(\)](#)

[emy.insertViews\(\)](#)

[emy.getSelectedView\(\)](#)

[emy.getAllViews\(\)](#)

Elements methods

[emy.\\$\(\)](#)

[emy.findParent\(\)](#)

[emy.hasClass\(\)](#)

[emy.addClass\(\)](#)

[emy.changeClass\(\)](#)

[emy.log\(\)](#)

Events

On Load

On load, Emy determines which view to display based on the top-level (child of the [body](#)) element with the [selected](#) attribute set to [true](#). This view is set as the homescreen and will then always be the root view. If the anchor part of the URL (everything after <#>, known as "hash") is empty, this view's [id](#) attribute value will be added to the URL. Ex: if `selected="true"` is set on `<section id="home" ... >` in index.html, then on load, Emy adds [#view1](#), resulting to `index.html#view1`. On load, this is done via `location.assign()`, so the browser history stack does not change.

The reason Emy does this is because it uses an anchor-based navigation mechanism like most mobile web libraries, which brings an automatic support of the browser's (and device's physical) back button. iPhone users might use Emy's toolbar back button as well as Safari's one, but Android or Blackberry or WindowsPhone users are more likely to use the device's physical back button.

on Click

Emy captures all clicks on [a](#) elements and goes through a series of checks to determine what to do.

```
<a href="#contact" title="Contact us">Contact us</a>
```

If the link has a `href="#..."`, Emy navigates to the first view in the DOM with an `id` attribute equals to what is specify after this `#`.

```
<a href="" id="backButton">Back</a>
```

If the a (link) element has an ID equals to `backButton`, Emy navigates to the previous screen (see [emy.goBack\(\)](#)). This is used only by the toolbar's back button.

```
<a href="" data-type="submit">Submit</a>  
<button href="" type="submit">Submit</button>
```

If the a (link) has a `data-type` or an input / button element has a `type` equals to `"submit"`, Emy looks for the parent `form` element, gathers up all the input values and submit the form via AJAX (see [emy.showViewByHref\(\)](#)).

```
<a href="" data-type="cancel">Cancel</a>  
<button href="" type="cancel">Cancel</button>
```

If the a (link) has a `data-type` or an input / button element has a `type` equals to `"cancel"`, Emy cancels the parent `form` element dialog.

```
<a href="more-links.html" target="_replace">Load more...</a>
```

If the a (link) has a `target` equals to `"_replace"`, Emy does an AJAX call based on the `href` value of this link, and replaces it with the content of the AJAX responseText.

```
<a href="mailto:me@domain.org">me@domain.org</a>  
<a href="tel:911">Call 911</a>  
<a href="javascript:myFunction()">myFunction()</a>
```

If the a (link) is a native URL (see [emy.isNativeURL\(\)](#)), Emy does nothing and let the browser do its job.

```
<a href="http://www.twitter.com" target="_webapp">My Twitter account</a>
```

If the `a` (link) has a `target="_webapp"`, Emy performs a normal link, navigating completely away from the Emy app and pointing the browser to the link. No AJAX call or sliding navigation is performed.

```
<a href="contact.html">Contact us</a>
```

If there is no `target` attribute, Emy performs an AJAX call based on the `href` value of this link (see [emy.showViewByHref\(\)](#)), adds as many views the responseText contains in its first node (see [emy.insertViews\(\)](#)), and slides to the first of the newly created views.

```
<div class="toggle" onclick="">
  <input type="hidden" name="mytoggle"></span>
  <span class="thumb"></span>
</div>
```

Emy also captures `div.toggle` clicks and displays/hides the element via setting a `toggled` attribute to true/false.

Emy events

Emy fires a number of custom events on your views. Handling these events is the recommended way to do any just-in-time transformations or loading (besides the ajax pre-loading built into Emy). To avoid conflict with other code/library, all emy events are prefixed with `emy-`. To register to these events, just use [addEventListener](#) (or equivalent that libraries like JQuery may offer with `.bind()` or `.on()`).

```
emy.$('#view2').addEventListener('emy-aftertransition', callbackFunction)
```

Focus / Blur

Views receive a `emy-focus` event when they are shown and a `emy-blur` event when hidden.

Load / Unload

Views receive a `emy-load` event and (only when going backwards away from a panel) an `emy-`

[unload](#) event.

Dialog views (aka with the class `dialog`) don't receive any [emy-load](#) or [emy-unload](#) events.

Beforeinsert / Afterinsert

When new views are inserted into the DOM (see [emy.insertViews\(\)](#)), the `body` element receives a [emy-beforeinsert](#) event with `{ fragment: frag }` parameters and afterwards receives an [emy-afterinsert](#) event with `{ insertedNode: docNode }` parameters.

Beforetransition / Aftertransition

Both views involved in a slide animation receive [emy-beforetransition](#) and [emy-aftertransition](#) events. The views being navigated from receives event parameters `{ out :true }`, views being navigated to receives `{ out: false }`.

Properties

emy.logging

If set to `true`, [emy.log\(\)](#) performs a `console.log()` if the browser supports it.

Default value is obviously `false`.

emy.busy

This is set to `true` if a slide animation or an AJAX call is in progress.

Default value is `false`.

emy.transitionMode

Determines whether to do slide animations with CSS3, Javascript with `setInterval` or no animation at all. Value can then be `"css3"`, `"js"` or `"none"` and is checked before each transition. If your browser doesn't support CSS3 transitions, then Emy uses Javascript.

Default value is `"css3"`.

emy.httpHeaders

An object defining headers to be sent with Ajax requests.

Default value is `{ 'X-Requested-With': 'XMLHttpRequest' }`.

emy.prefixedProperty

An array containing vendor-prefixed methods for `transform`, `transition`, `transitionEnd`, `animationDuration` & `animationEnd`. It is populated by the `onLoad` event function, and only used by the `slideViews()` function.

This array has no default value (empty array).

Methods

emy.isNativeUrl(url)

url: a valid URL (string)

Determines whether the supplied URL string launches a native iPhone app (maps, YouTube, phone, email, etc). If so, Emy does nothing (doesn't attempt to load a page or slide to it) and allows the phone to handle it the click event natively.

```
emy.isNativeUrl('tel:123123123');    // returns 'true'  
emy.isNativeUrl('mailto:an_em@il_address');    // returns 'true'  
emy.isNativeUrl('http://maps.google.com');    // returns 'true'  
emy.isNativeUrl('maps');    // returns 'false'  
emy.isNativeUrl('view2.html');    // returns 'false'
```

emy.goBack(viewId)

viewId: the ID value of a view (string)

Navigates back to the view in the history stack equals to **viewId**'s id.

If **viewId** is set, it searches this view in the history stack and go backward in it as much as needed - note that if the id is not found in the stack, it navigates forward as a regular [gotoView](#).

If **viewId** is not set, it just goes back to the previous one.

```
emy.goBack();    // this goes one step backward  
emy.goBack('view2');    // this goes backward to the specified view in
```

emy.gotoView(view, backward)

view: a view element (DOM Element)

backward: true / false (Boolean)

view is the html element to show. If **backward** is set to **true**, it performs a right-to-left transition instead of the default left-to-right.

If **view** is the currently-displayed view, Emy does nothing. [showView\(\)](#) is used for both panel-type views and dialog-type views (dialogs float on top of others, have a cancel button and do not participate in sliding animations). Panel-type views receive [blur/focus](#) events and [load/unload](#)

events, but dialog-type views only receive [blur/focus](#) events.

```
emy.gotoView(emy.$('#view2'), false);    // this script navigates forward
emy.gotoView(emy.$('#view2'), true);     // this script navigates backward
```

emy.showView(view, backward)

view: a view element (DOM Element)

backward: true / false (Boolean)

view is the html element to show. If **backward** is set to **true**, it performs a right-to-left transition instead of the default left-to-right.

If **view** is the currently-displayed view, Emy does nothing. [showView\(\)](#) is used for both panel-type views and dialog-type views (dialogs float on top of others, have a cancel button and do not participate in sliding animations). Panel-type views receive [blur/focus](#) events and [load/unload](#) events, but dialog-type views only receive [blur/focus](#) events.

```
// this script navigates to the <section id="view2" ... >
emy.showView('view2');
```

emy.showViewById(viewId)

viewId: the ID value of a view (string)

Calling [gotoView\(\)](#) with **backward** set to **false**.

```
// this script navigates to the <section id="view2" ... >
emy.showViewById('view2');
```

emy.showViewByHref(url, args, method, replace, cb)

url: a valid URL (string)

args: key-value pairs (object)

method: HTTP method (string - "GET" by default)

replace: the element to replace (DOM element)

cb: callback function.

This method is used by Emy when a link points to a file instead of an anchor, and when submitting a form.

[href](#) should be a valid URL to a file, set as a string. If not valid, Emy triggers [emy.ajaxErrorHandler](#).

[args](#) parameter is an object of key-value pairs that are used to generate the querystring in case of a GET call.

[method](#) defines how the Ajax call should send [args](#) value, with "GET" & "POST" being the two possible values.

[replace](#) parameter is an existing element that either is the panel or is a child of the view that the incoming HTML will replace (if not supplied, Emy appends the incoming HTML to the [body](#)).

[cb](#) is a user-supplied callback function, triggered when the job is done.

```
// this link triggers showViewByHref
&lr;a href="external-view.html">Load external view&lr;/a>

// this link triggers showViewByHref and replace the link parent node
&lr;a href="external-view.html" target="_replace">Load external view&lr;

// or you can trigger the method by yourself in your code
emy.showViewByHref('external-view.html', null, 'GET', nodeElementToRepl
    // triggered 1 second after the ajax call loads the external file
});
```

emy.ajax(url, params, method, callback, errorCallback)

[url](#): a valid URL (string)

[params](#): key-value pairs (object)

[method](#): HTTP method (string - "GET" by default)

[callback](#): callback function. [errorCallback](#): callback function call fails (new in 1.1)

Handles ajax requests and also fires a [setTimeout\(\)](#) call to abort the request if it takes longer than [ajaxTimeoutVal](#) (default is 30sec).

[href](#) should be a valid URL to a file, set as a string. If not valid, [errorCallback](#) is called.

[params](#) is an object of key-value pairs that are used to generate the querystring in case of a GET call or as POST variable in case of POST call.

[method](#) defines how the Ajax call should send [params](#) value, with "GET" & "POST" being the two possible values.

[callback](#) is a user-supplied callback function, triggered when the job is done. It returns the `responseText` as plain-text, so if you do it on www.emy-library.org, you get the HTML code of the homepage.

Note that Emy 1.0 used to trigger this function on each readyState change, where Emy 1.1 only triggers it once when job the call is a success (aka readyState=4 and status=200).

errorCallback is a user-supplied callback function, triggered when the call fails. This can happens for many reasons: wrong url, cross-domain call, timeout, server is down, ... It returns the full **XMLHttpRequest** object.

```
var myParams = {"myvar":"foobar", "myothervar": 12};
emy.ajax('xhr/my-ajax-script.php', myParams, 'GET', function success(re
    // 'result' is equal to the responseText
}, function error(result) {
    // if the ajax call returns an error (no connection, x-domain issue
    // you can check at result.readyState, result.status, result.respon
});
```

emy.insertViews(frag, go)

frag: HTML fragment (fragment)

go: Boolean (true by default) (new in 1.1)

Insert new view(s) in the DOM as registered Emy view(s).

It is used by [showViewByHref\(\)](#) to insert the newly ajax-loaded content as views.

Each childnode of the HTML fragment is added as a new view, and replace an existing one (aka same **id** value).

go parameter, **true** by default, means Emy trigger the [showView](#) to the first childnode/view of the newly inserted views. To avoid that, just set it to **false**.

```
emy.insertViews( nodeElements); // insert views and navigate to the fi
emy.insertViews( nodeElements, false); // insert views but doesn't nav
```

emy.getSelectedView()

Returns the currently selected view element, which means its **selected** attribute equals **true**.

```
emy.getSelectedView(); // returns a DOM element
```

emy.getAllViews()

Returns all views in the DOM.

```
emy.getAllViews(); // returns an array of DOM elements
```

emy.\$(a)

a: string

Convenience function to select elements.

if the **a** parameter's first letter is a "#", it selects the element with the **id** equals to this string without the "#".

if the **a** parameter's first letter is a ".", it selects all elements with the **class** containing this string without the ".".

if the **a** parameter contains a "." but not at the first letter, it splits the string, uses letters before the "." as a tag element name, then selects all those tag elements with the **class** containing what's after the ".".

if the **a** parameter contains nothing but letters, it selects all elements being equal to it.

```
emy.$('#view2'); // return the first DOM element with 'id' at
emy.$('.panel'); // return an array of DOM elements with cla
emy.$('section'); // return an array of 'section' tagname DOM
emy.$('#view2 .row'); // return an array of DOM element with class at
```

We use '\$' not to mimic JQuery but because one single character is quick to type. Since it's part of Emy object, it won't conflict with JQuery in case you use it.

emy.findParent(node, tagName)

node: a node

tagName: HTML tag element name

Returns the first tag element inside the parent of the specified node. If no tag found, it tries in the parent's parent until there is no more parent node (html tag).

```
emy.findParent( nodeElement, 'div');
```

emy.hasClass(self, name)

self: a DOM element

name: a string

Convenience function to determine if the given element (**self**) has the class **name**.

```
emy.hasClass( emy.$('#view1'), 'panel');
```

emy.addClass(self, name)

self: a DOM element

name: a string

Convenience function to add the given class **name** to element **self**.

```
emy.addClass( emy.$('#view1'), 'panel');
```

emy.changeClass(self, name, newname)

self: a DOM element

name: a string **newname**: a string

Convenience function to change a given class **name** to element **self**. You can also use this to remove a class, by simply leave name to an empty string.

```
emy.changeClass( emy.$('#view1'), 'panel', 'grid');  
emy.changeClass( emy.$('#view1'), 'panel', '');      // quick way to re
```

emy.log(a)

a: any type of variable

Alias function of console.log(a). Main difference is that it tests if your browser supports it and if [emy.logging](#) is set to **true** so you can switch back logging even in production.

```
emy.log('view2');      // returns "view2"  
emy.log( emy.$('view2') ); // returns false or the DOM element  
emy.log(myArray);      // returns an array
```

emy.v()

(new in 1.1)

Returns the version number of Emy. Useful if you develop a plugin or an extension.

```
emy.v(); // return '1.1'
```

More ressources

[Getting started guide](#)

For a gentler introduction to Emy.

[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

Portions of this content are ©2012–2013 by individual Emy contributors. Content available under the [Creative Commons 3 BY-SA license](#).

