

How to build a Widget

For an up-to-date version and discussions see
<https://github.com/consti/widget-howto>

Constantin Hofstetter

January 9, 2015

Widgets

Design a widget infrastructure so that a client can implement a part of your website as a third-party component.

Goals

- ▶ Easy to integrate for the client
- ▶ Flexible for us to change in the future
- ▶ Customizable
- ▶ Responsive
- ▶ Cross-Browser compliant
- ▶ Good for SEO

Integration: IFRAME vs. JSONP

IFRAME

Render the widget as an IFRAME element.

- ▶ Advantages: IFRAME content is sandboxed: CSS and Javascript that will not interfere with the clients page.
- ▶ Drawbacks: Impossible to get the IFRAMES height or the `document.location` from the client site (cross-domain security).

JSONP

Render the widget as complete Javascript/AJAX request.

- ▶ Advantages: We can freely interact with the client site, since everything is rendered inline and in the current DOM. Usually also faster to render than an IFRAME.
- ▶ Drawbacks: Our CSS and Javascript interferes with the client site. Usually a lot more difficult to maintain.

Integration: Best practice

<script/> that renders an IFRAME

```
# On the client site
<div id="widget-placeholder" />
<script src="http://example.com/widget.js" />

# in widget.js
var ph = document.getElementById('widget-placeholder');
var widget = document.createElement('IFRAME');
widget.src = "http://example.com/widget?location=" + document.
    location;
ph.parentNode.replaceChild(widget, ph);
```

Easy to integrate for the client and flexible for us to change in the future: we have full access to the DOM and our IFRAME (we can pass arguments to the URL or use `postMessage` in Javascript).

Customization

Add data attributes to the placeholder

```
# On the client site
<div id="widget-placeholder" data-widget-locale="de" />
<script src="http://example.com/widget.js" />

# in widget.js
var widget = document.createElement('IFRAME');
var ph = document.getElementById('widget-placeholder');
var locale = ph.getAttribute('data-widget-locale');
widget.src = "http://example.com/widget?locale=" + locale;
ph.parentNode.replaceChild(widget, ph);
```

Why not add it to the script URL? Google and others will crawl the URL and we also want widget.js to be cachable and static.

Responsiveness

Widget should have a default, but customizable height and resize the width based on the parent element (the context it is based in).

```
# in widget.js
widget.height = "600";
if (ph.getAttribute('data-widget-height')) {
  widget.height = ph.getAttribute('data-widget-height');
}
widget.style = "border: none; max-width: 100%; min-width: 180px
; width: 520px;";
widget.src = "http://example.com/widget?height="+ widget.height
;
```

This way we have a widget, that is responsive in width (min 180px, max 520px) and height (default 600, but resizable). We pass the height to our widget view since we want to know it at render.

Cross-Browser compliance

Using an IFRAME makes it a lot easier to be browser compliant - just make sure the rendered widget works in every browser. For the widget.js, make sure to keep it small and don't depend on third-party libraries (no jQuery etc.).

Allow multiple widgets on one page (and still not use `getElementsByClassName` which does not work with some browsers):

```
# on client site
<div class='widget-placeholder' />
<script src="http://example.com/widget.js" />
# in widget.js
all_phs = document.getElementsByTagName('widget-placeholder');
last_ph = all_phs[all_phs.length - 1];
```


SEO: Backlink Win

Widgets that get integrated on many pages can really boost SEO:

```
<a class='widget-placeholder'  
  data-widget-city='bangkok'  
  href='http://example.com/cities/bangkok'  
    Expert Examples in Bangkok  
</a>  
<script src="http://example.com/widget.js" />
```

Use a backlink as placeholder that gets replaced by an IFRAME at render. This is not prohibited nor discouraged.

Try to link to the corresponding full-website view of the widget content.

SEO: Gotchas

- ▶ Don't communicate URLs to the embedded widget view (the IFRAME src website). Build them in Javascript.
- ▶ Don't add arguments to the widget.js URL. Add them as data attribute to the placeholder.
- ▶ Add `<meta name="robots" content="noindex, follow"/>` to the embedded widget view (the IFRAME src website). So that if the widget pages get crawled, they don't show up in the search index.
- ▶ If possible, serve the widget data (the Javascript and the embedded view) on a separate subdomain, so that if things go wrong, you don't screw up your main domain.

Development in Rails

Make sure you test your widget from different domains before shipping.

```
class WidgetController < ApplicationController
  protect_from_forgery except: :script
  after_action :allow_iframe, only: :embed
  layout false

  def script
  end

  def embed
  end

  private
  def allow_iframe
    response.headers.except! 'X-Frame-Options'
  end
end
```