

EMBEDDED WIDGETS WITH EMBER.JS

by Anton Katunin
@antulik

WHAT IS WIDGET

- In my case: your app which will be embedded on some website you don't have control of
- Your ember application next to other CSS and JS libraries and frameworks
- No iframe because of third party cookies 🍪

EXAMPLE SNIPPET

```
<script type="text/javascript">
  (function(i,s,o,g,r,a,m){i['OpendataEmbedded']=r;i[r]=i[r]||function(){
  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
  })(window,document,'script','//static.opendata.dev/embedded.js','OpendataEmbedded');

  OpendataEmbedded('create', 1, 'your-element-selector');
</script>
```

PREPARE YOUR APP CONFIG

- Changing the Root Element
- Disabling URL Management
- Specifying a Root URL

<https://guides.emberjs.com/v2.5.0/configuring-ember/embedding-applications>

CHANGING ROOT ELEMENT

```
export default  
Ember.Application.extend({  
  rootElement: '#app'  
});
```

- Dynamically? (more on that later)

```
$(container).addClass('os-embedded-app');
```

DISABLING URL MANAGEMENT

- Set to none or hash

```
var ENV = {  
  locationType: 'none'  
};
```

SPECIFYING A ROOT URL

- set in the initializer:

```
export function initialize(application) {  
  application  
    .__container__  
    .lookupFactory('config:environment')  
    .baseUrl = window.location.pathname;  
}
```

```
export default {  
  name: 'base-url',  
  initialize  
};
```

MORE THINGS

- Delaying application start
- Dynamic configuration
- Issues with CSS
- Integration script
- Merge CSS with JS (race condition)
- Other

DELAYING APP START

- <https://github.com/xcambar/ember-cli-embedded>
- Why?
- How?

```
application.deferReadiness();  
application.advanceReadiness();
```

DYNAMIC CONFIGURATION

- <https://github.com/ember-cli/ember-export-application-global>

- Pass when starting

```
MyApp.start(config)
```

- Or do it yourself

```
MyApp.__container__  
  .lookupFactory('config:environment')  
  .APP.programId = 123;
```

- As a note

```
const embeddedConfig = container.lookup('config:embedded');
```

CSS ISSUES

Conflict of styles with existing styles

1. Prefix all your CSS classes
2. Using CSS libraries?
Scope and nest them to your application with Less/Sass
`.btn -> .ember-application .btn`
3. Compose your classes with framework classes
`._btn { .btn; }`
4. Use <https://github.com/salsify/ember-css-modules>
5. Use <https://github.com/ebryn/ember-component-css>
6. And fix the rest of conflicting CSS manually with overrides

INTEGRATION SCRIPT

- Provides interface for end user to configure and interact with your application
- Very small, async
- Similar to any analytics script
- Includes: load fonts, load external css libraries, render placeholder, add analytics

MERGE CSS WITH JS

- Load race condition of styles and js
- Idea is to have single JS file that you need to load
- Welcome **webpack** for the rescue
(webpack.github.io)
- Never block, keep it async

OTHER

- Google analytics with **ember-cli-google-analytics**
- Fix conflicts of two google analytics by loading into different variable

TO MENTION

- CDN and gzip

- Embedding assets

<https://github.com/dfreeman/ember-cli-node-assets>

- Conflicts of JS libraries?

THANK YOU

Anton Katunin
@antulik