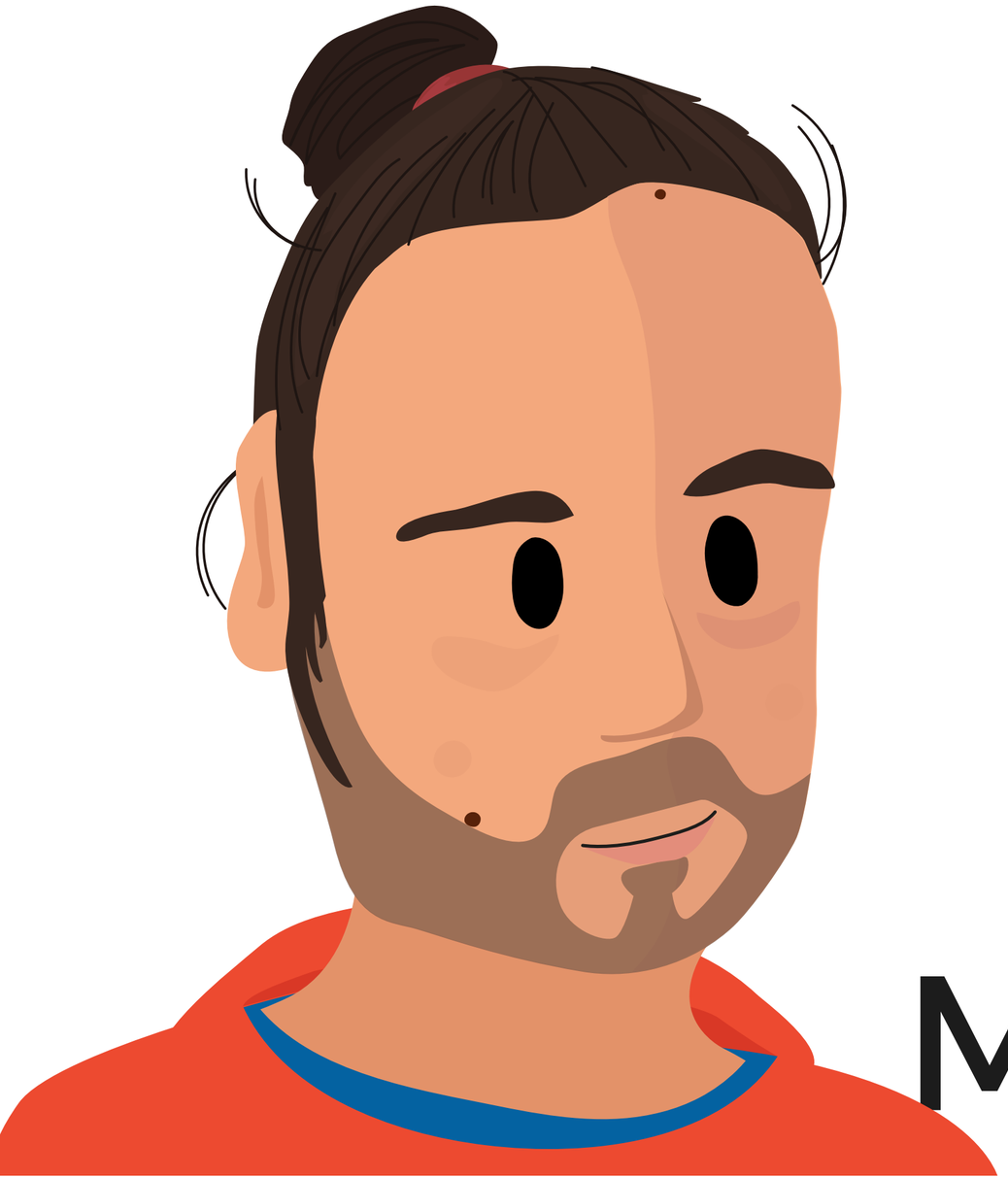
Progressive Decoupling

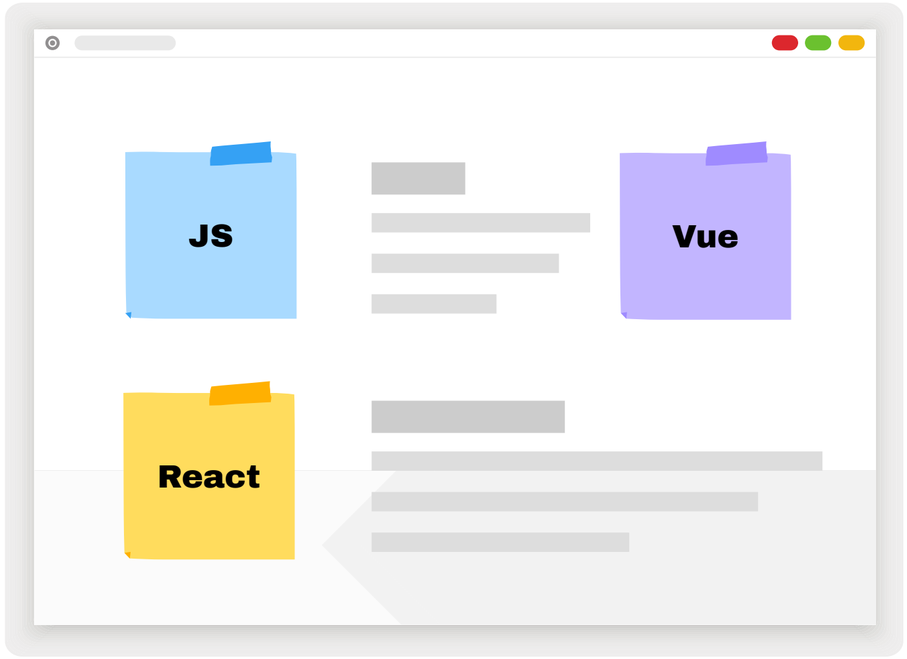


Mateu / eOipso

What we'll cover

1. What & why widgets
2. CMS integrations
3. The widget registry federation

What are “widgets”?



1. Stand alone JavaScript applications. Any framework.

2. Designed to be embedded and configured by CMS editors.

What are the benefits of JS over server generated HTML?

Better reactivity

Interaction in static pages

Bigger talent pool

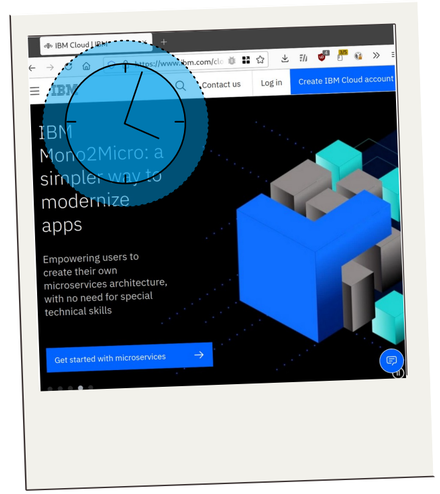
Distributed delivery

Better dev experience

Better Reactivity and Interactivity

The server can provide the unchanging parts, while the JS application adds interactivity. This reduces load on your servers while increasing website performance.

1. Pages can be static or served from cache (very fast)

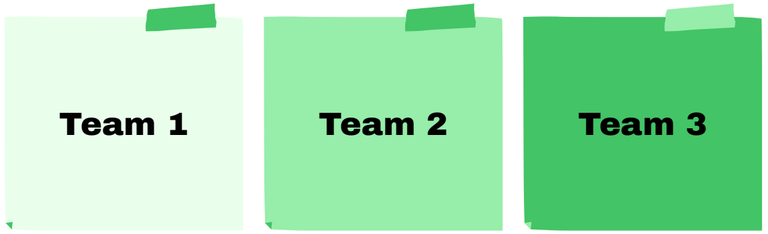


2. JS adds the reactive / interactive part

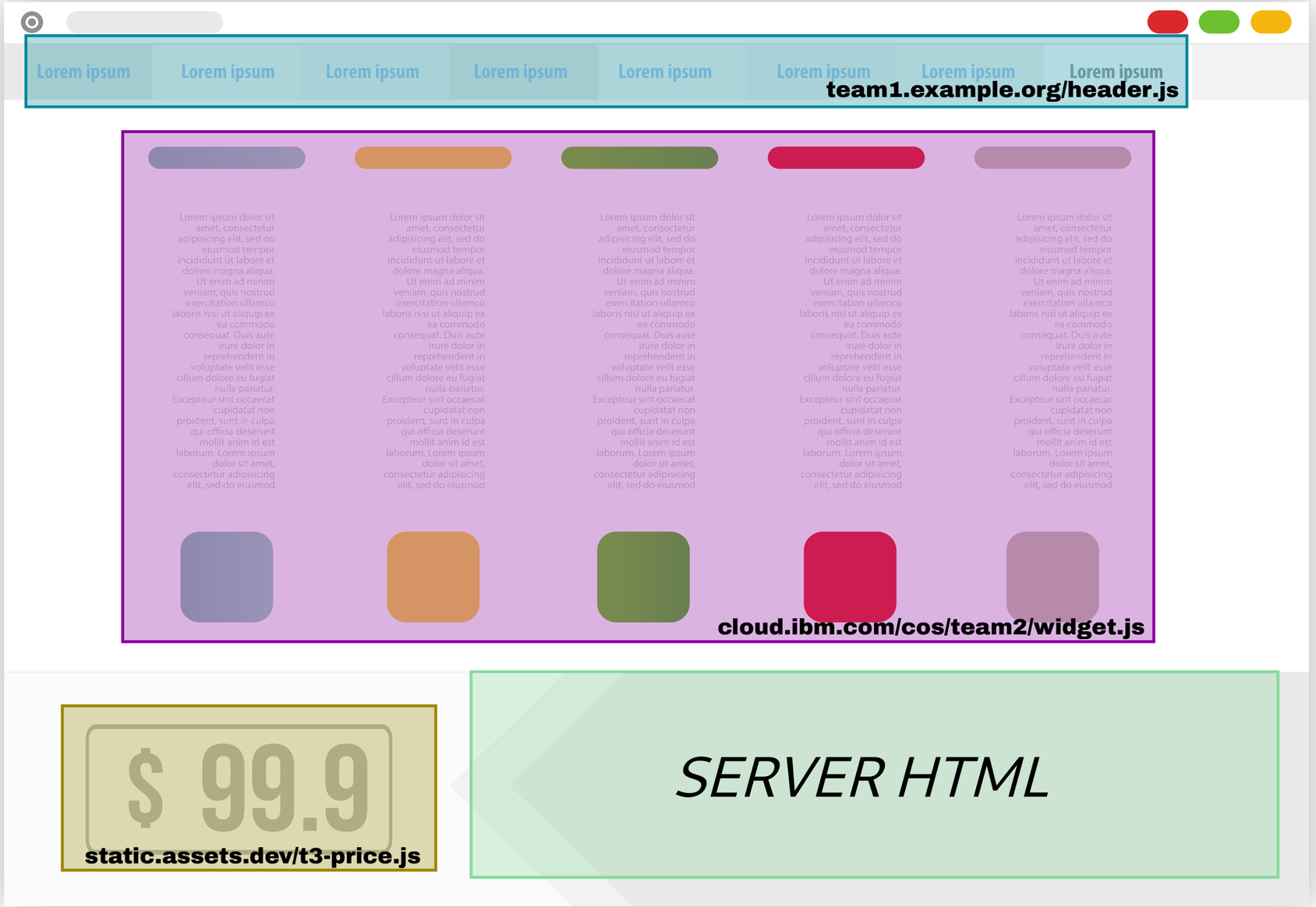
Distributed Delivery

Different development teams write software independently. They they can publish software on the same platform without coordinating complex deployment effors.

1. Teams write the JS code in isolation.



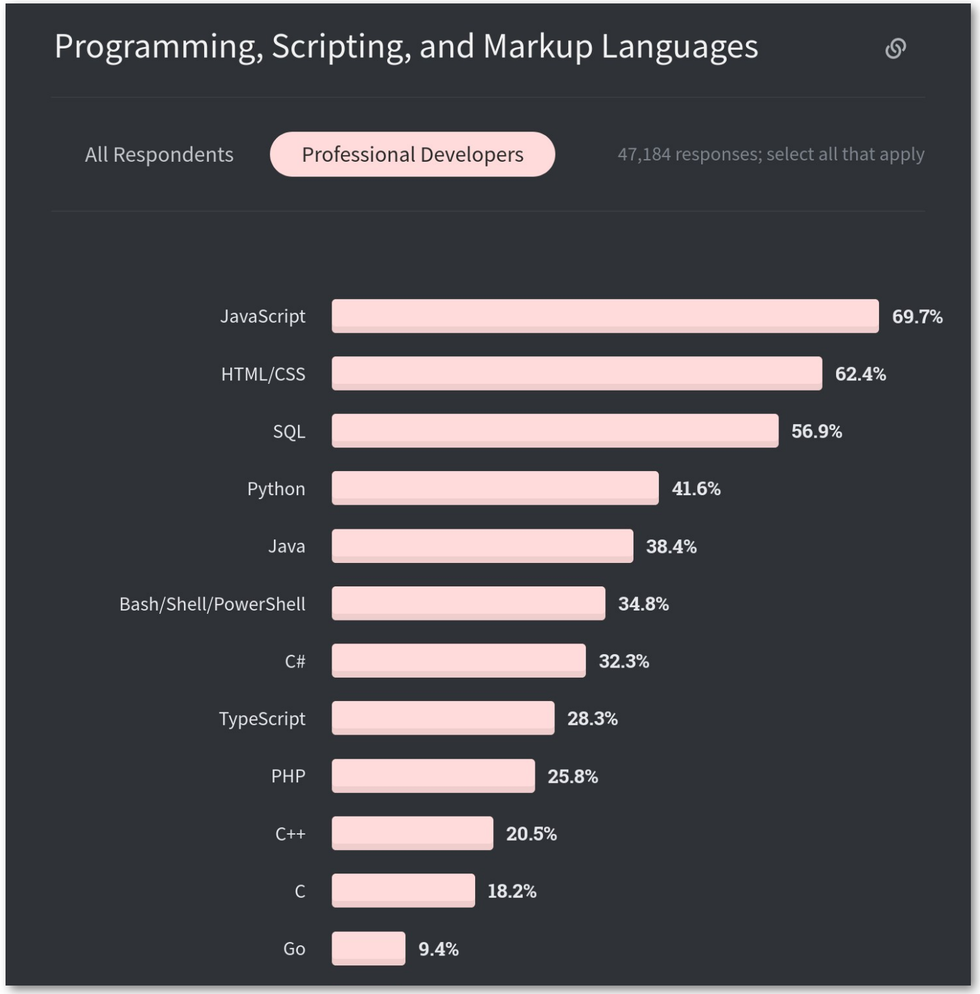
2. Execute the JS in the browser in the website.


Lorem ipsum
Lorem ipsum
Lorem ipsum
Lorem ipsum
Lorem ipsum
Lorem ipsum
Lorem ipsum
Lorem ipsum
team1.example.org/header.js
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
cloud.ibm.com/cos/team2/widget.js
$ 99.9
static.assets.dev/t3-price.js
SERVER HTML

Biggest Talent Pool

According to extensive surveys, JavaScript and TypeScript are the most commonly used languages.

Since JavaScript is so popular you can leverage many services that integrate with this technology.



Better Dev Experience

Since JS is so popular there are many **tools**, **services**, and **frameworks** that your developers can leverage.

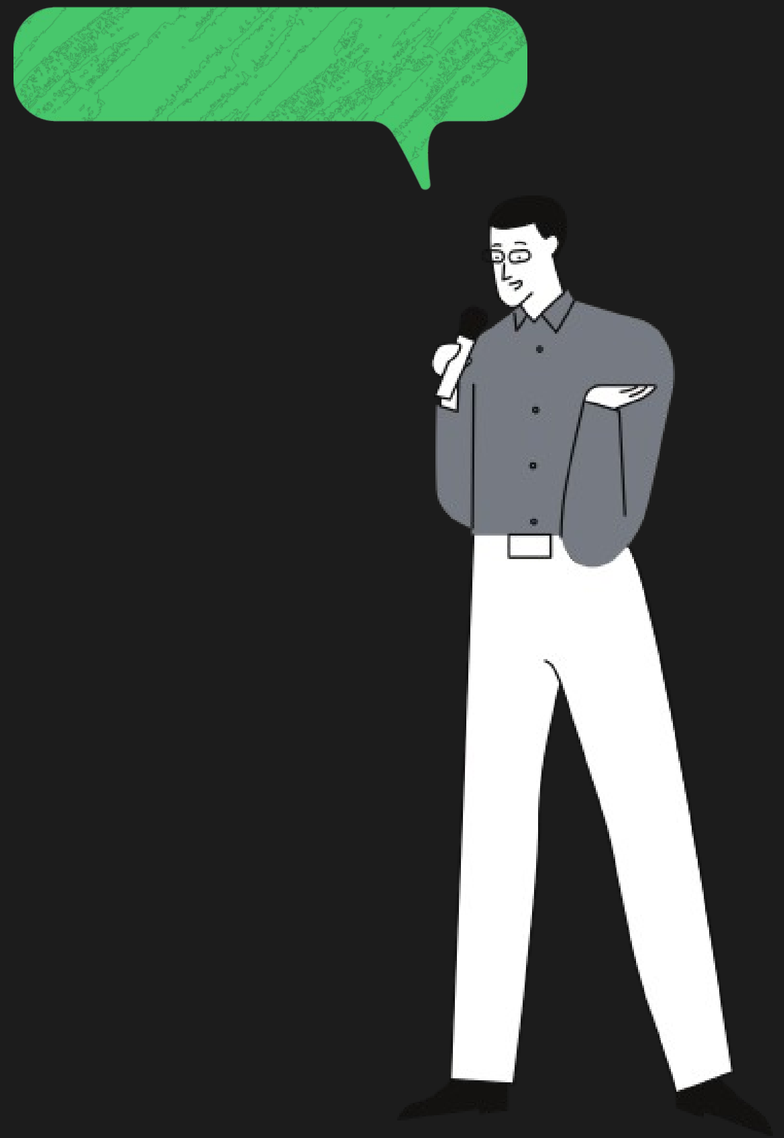
Many tools to help improve the **quality of the project**

Jest | **Storybook** | **...**

Many frameworks that will **take care of the rough edges**

Web Components | **Nuxt.js** | **...**

Should we build JavaScript apps, then?



Content, content, content

sitecore®

Server side CMS

When **content is your product**, or content is central to your business you need a CMS. **CMS provide many features** hard to build from scratch.

- Content server
- Manage pages (URLs, …)
- Access restrictions
- Manage metadata (SEO, …)
- Media library
- Security patches
- Editorially controlled layouts
- Moderation & preview
- …

How about JS in our CMS templates?

What are the benefits of widgets over JS in templates?

Improved time to market

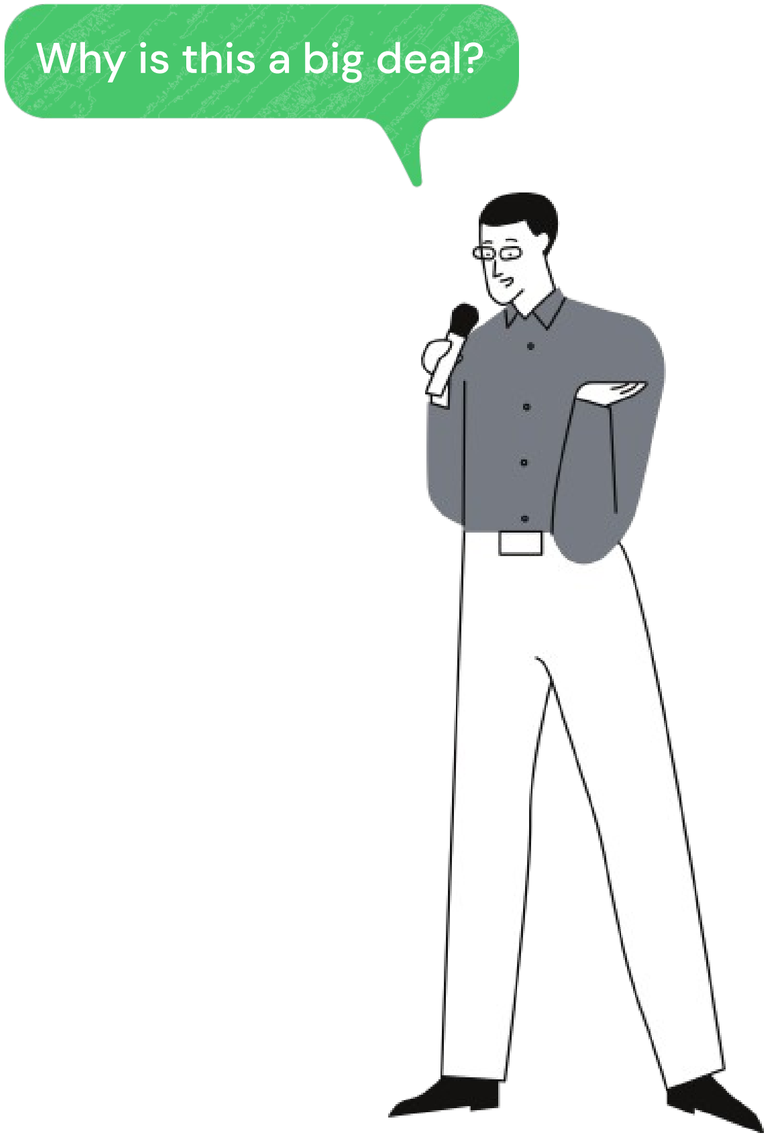
No CMS deployments

Embedded by editors

Embed anywhere!

No CMS deployments

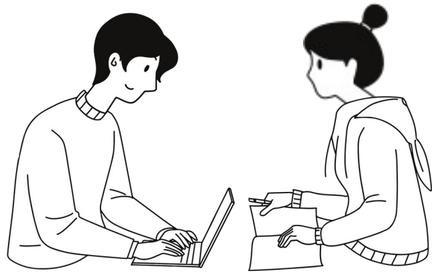
A developer creates a new widget in the registry and it appears in the CMS editorial interface for embedding. No additional effort. Bug fixes and enhancements are also instantaneous.



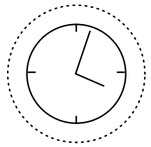
Development

JS & CMS

Editorial



End user

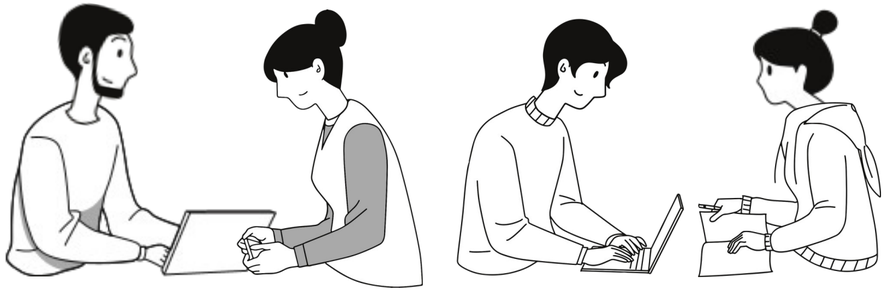


Deployment

Publication

Development & editorial

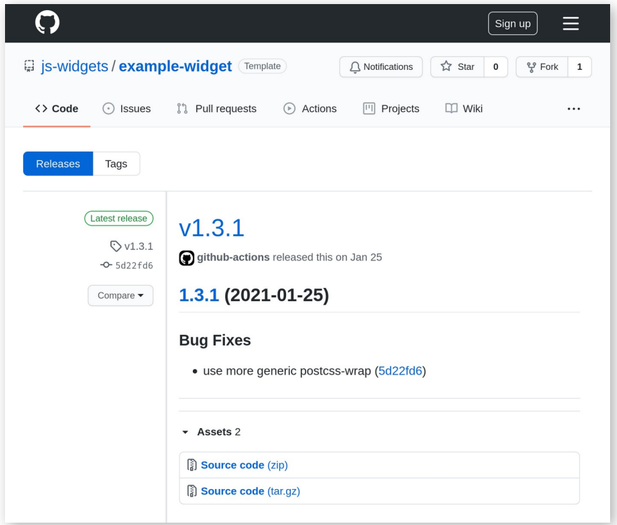
only JS!

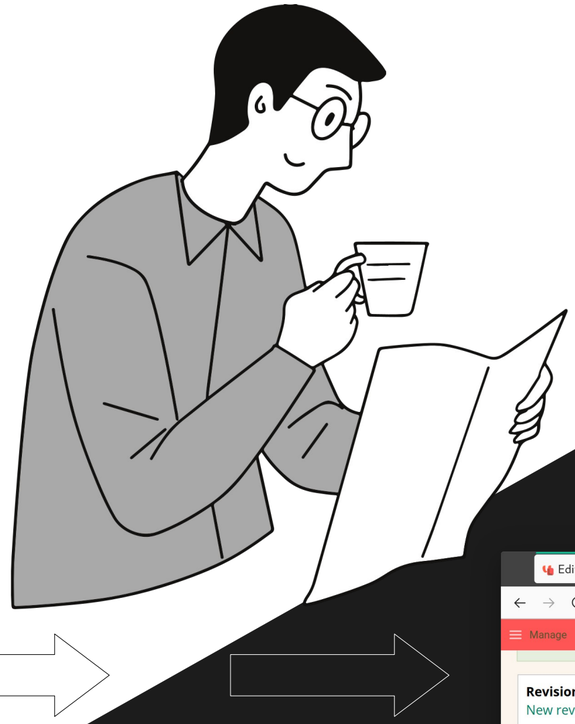


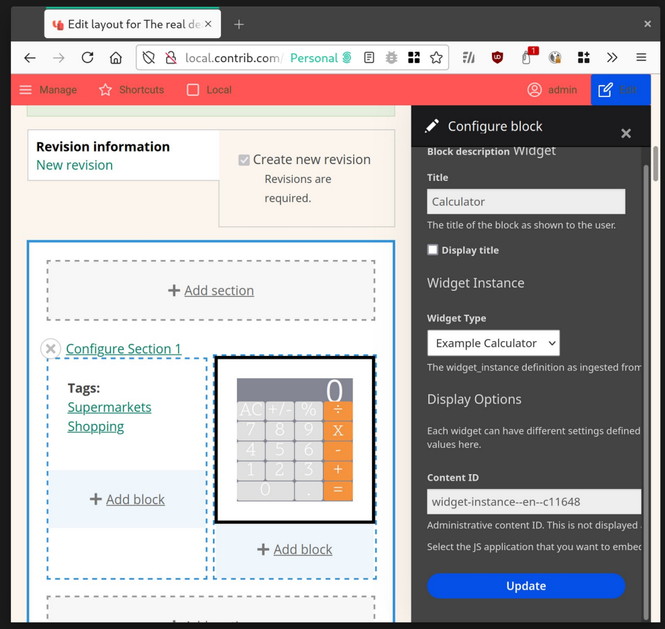
End user

JS development
immediately appears in CMS

JS

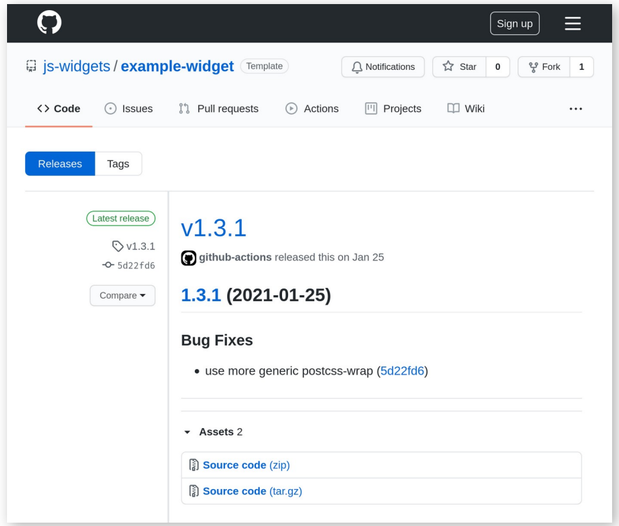

Sign up
js-widgets / example-widget
Template
Notifications
Star 0
Fork 1
Code
Issues
Pull requests
Actions
Projects
Wiki
Releases
Tags
Latest release
v1.3.1
5d22fd6
Compare
v1.3.1
github-actions released this on Jan 25
1.3.1 (2021-01-25)
Bug Fixes
use more generic postcss-wrap (5d22fd6)
Assets 2
Source code (zip)
Source code (tar.gz)

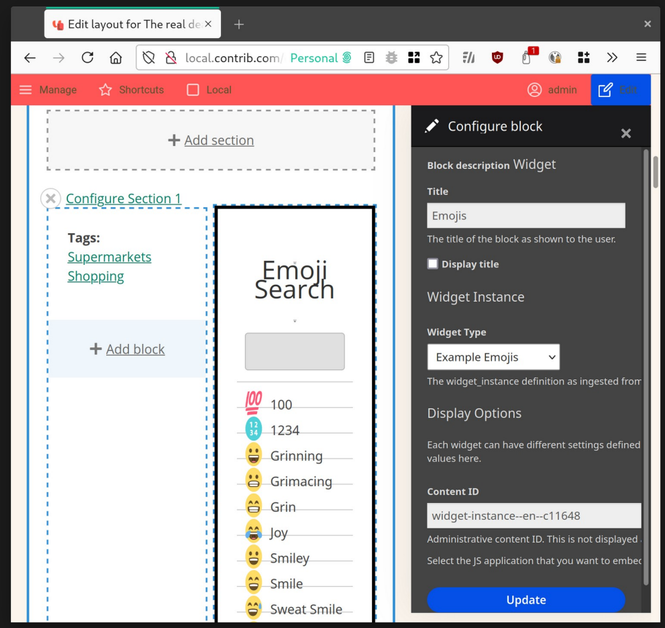



Edit layout for The real de
local.contrib.com
Personal
Manage
Shortcuts
Local
admin
Edit
Revision information
New revision
Create new revision
Revisions are required.
Add section
Configure Section 1
Tags:
Supermarkets
Shopping
Add block
Add block
Configure block
Block description Widget
Title
Calculator
The title of the block as shown to the user.
Display title
Widget Instance
Widget Type
Example Calculator
The widget_instance definition as ingested from
Display Options
Each widget can have different settings defined
values here.
Content ID
widget-instance--en--c11648
Administrative content ID. This is not displayed
Select the JS application that you want to embe
Update

CMS

JS


Sign up
js-widgets / example-widget
Template
Notifications
Star 0
Fork 1
Code
Issues
Pull requests
Actions
Projects
Wiki
Releases
Tags
Latest release
v1.3.1
5d22fd6
Compare
v1.3.1
github-actions released this on Jan 25
1.3.1 (2021-01-25)
Bug Fixes
use more generic postcss-wrap (5d22fd6)
Assets 2
Source code (zip)
Source code (tar.gz)


Edit layout for The real de
local.contrib.com
Personal
Manage
Shortcuts
Local
admin
Edit
Add section
Configure Section 1
Tags:
Supermarkets
Shopping
Add block
Emoji Search
100
1234
Grinning
Grimacing
Grin
Joy
Smiley
Smile
Sweat Smile
Configure block
Block description Widget
Title
Emojis
The title of the block as shown to the user.
Display title
Widget Instance
Widget Type
Example Emojis
The widget_instance definition as ingested from
Display Options
Each widget can have different settings defined
values here.
Content ID
widget-instance--en--c11648
Administrative content ID. This is not displayed
Select the JS application that you want to embe
Update

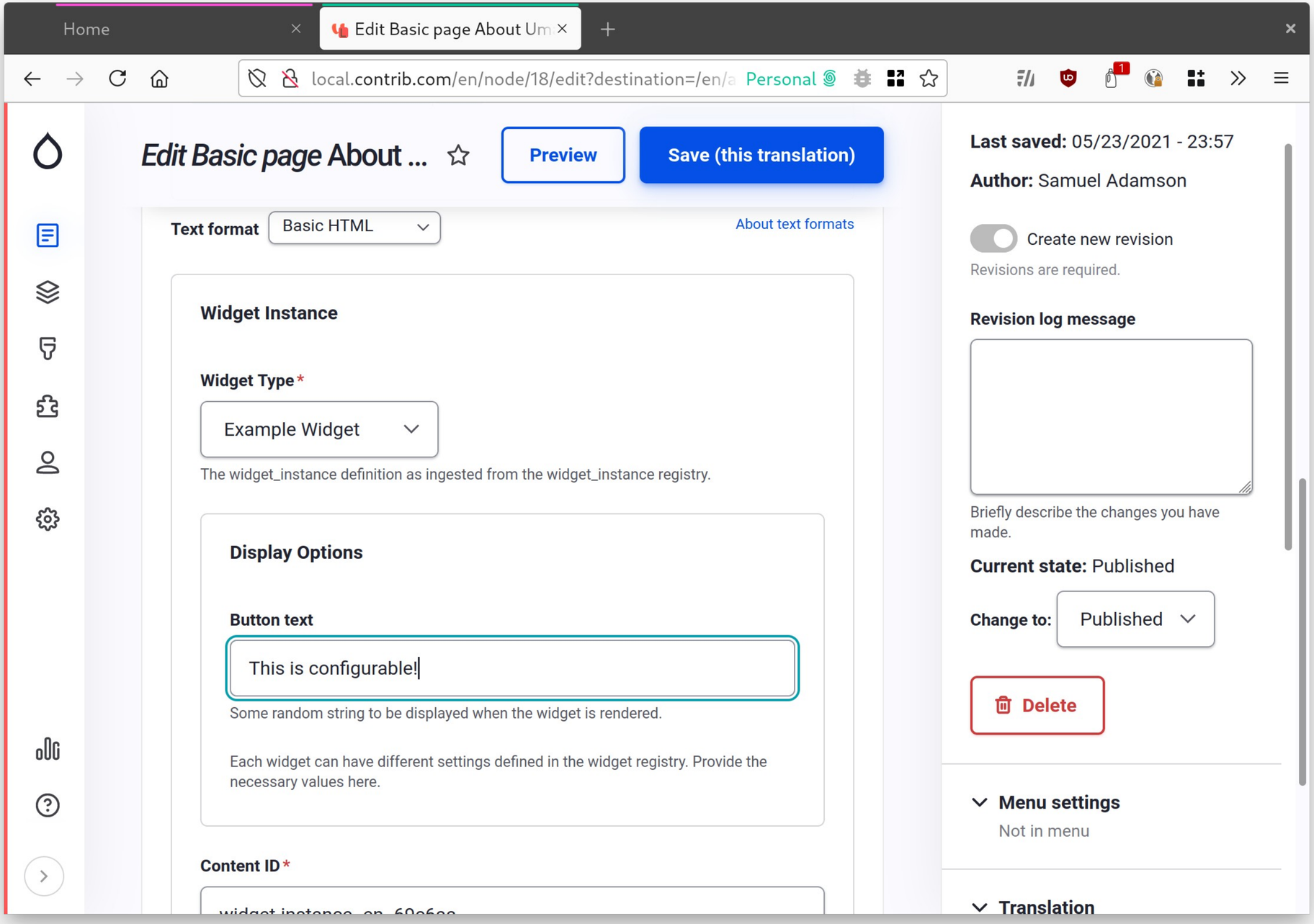
CMS

Embedded by editors

The JS developers define the input data they expect from editors. The CMS creates a form for editors to input such data.

Many instances of the same type of widget can be embedded with different configurations: different content, color palettes, external integrations, ...

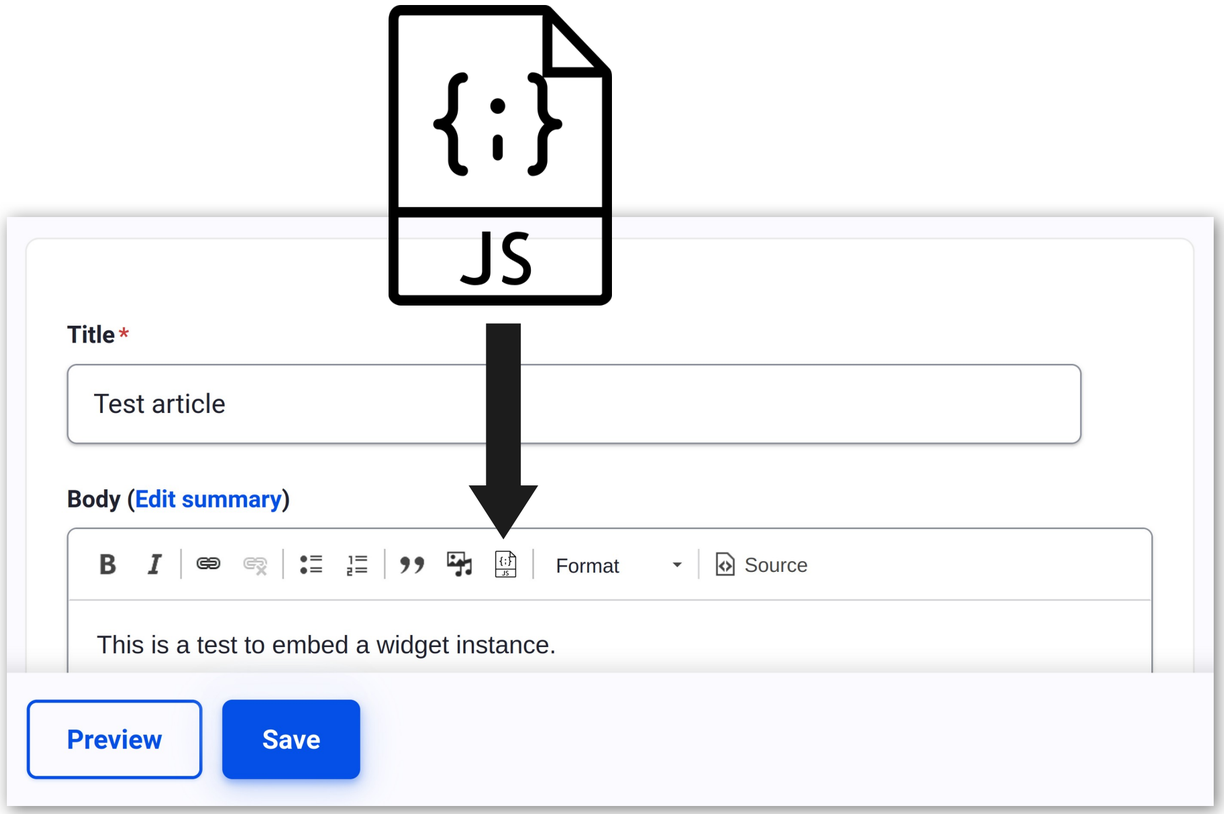
```
settingsSchema: {
  type: 'object',
  additionalProperties: false,
  properties: {
    fields: {
      type: 'object',
      properties: {
        'button-text': {
          type: 'string',
          title: 'Button text',
          description:
            'Some random string to be displayed when the widget is rendered.',
          examples: ['I am a button', 'Please, click me', 'CLICK'],
        },
      },
    },
  },
},
title: 'Example Widget',
status: 'stable',
```


Home
Edit Basic page About Um
local.contrib.com/en/node/18/edit?destination=/en/a
Personal
Edit Basic page About ...
Preview
Save (this translation)
Text format
Basic HTML
About text formats
Widget Instance
Widget Type*
Example Widget
The widget_instance definition as ingested from the widget_instance registry.
Display Options
Button text
This is configurable!
Some random string to be displayed when the widget is rendered.
Each widget can have different settings defined in the widget registry. Provide the necessary values here.
Content ID*
Last saved: 05/23/2021 - 23:57
Author: Samuel Adamson
Create new revision
Revisions are required.
Revision log message
Briefly describe the changes you have made.
Current state: Published
Change to:
Published
Delete
Menu settings
Not in menu
Translation

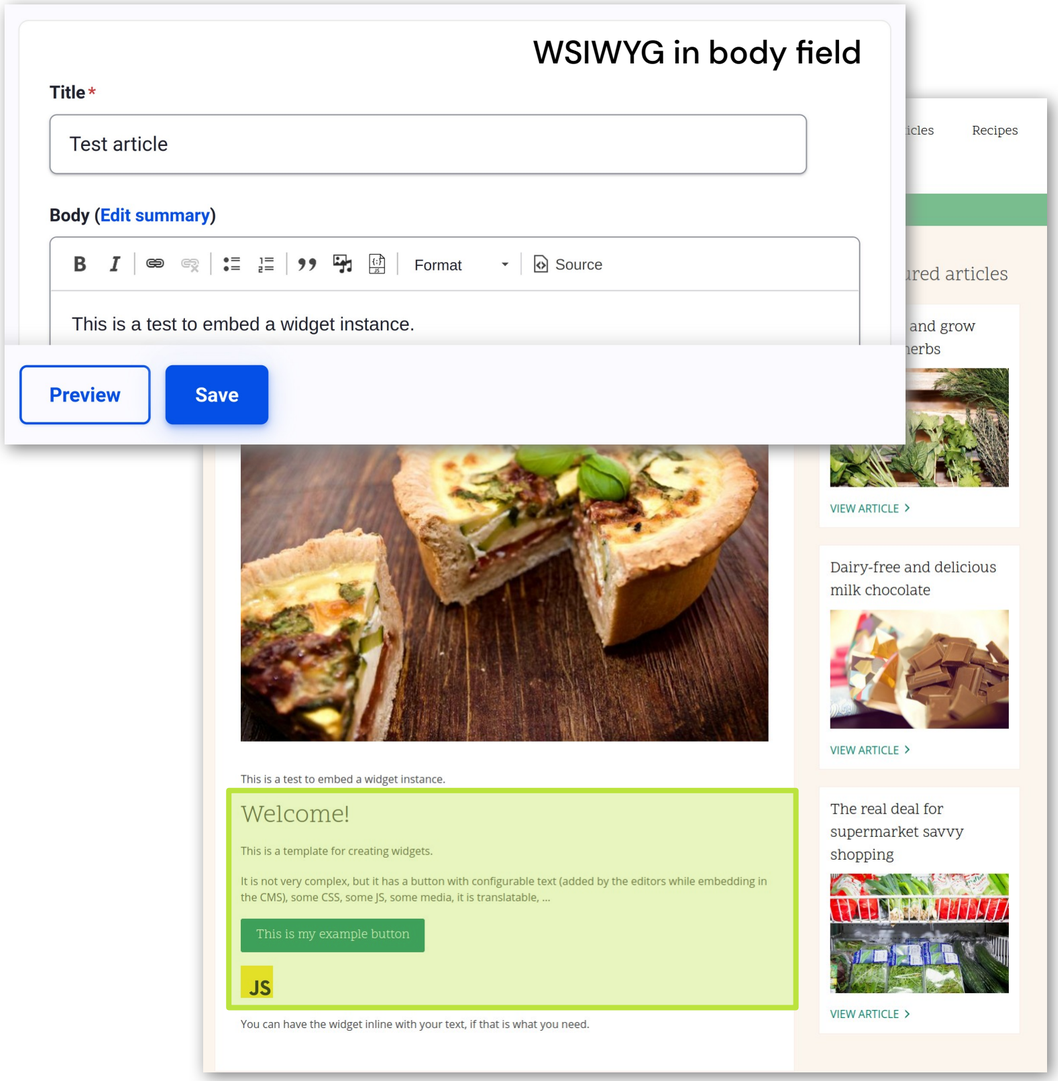
Embedded anywhere

Since widgets are not embedded at build time, but editorially, they can be placed anywhere:

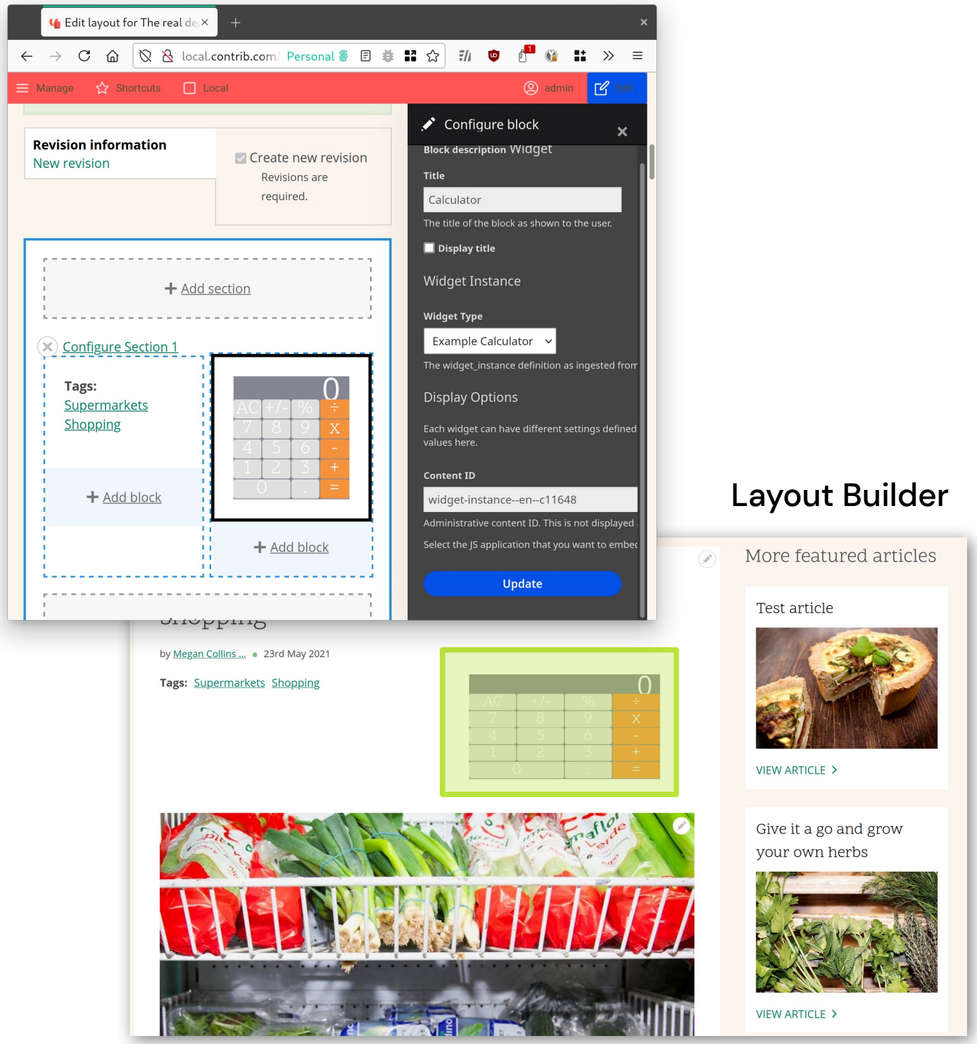
- Using layout building tools
- Using WYSIWYG integrations
- Using content modeling tools
- Using 3rd party JavaScript
- …



WSIWYG in body field



Layout Builder



When are widgets a good fit?

Structured content is still the way to go! Still, widgets are useful tools in several contexts.

- Interacting with 3rd party APIs
  Like review sites, commenting, …
- Interactive tools
  Like pricing calculators, checklists, …
- Data visualizations
  Like maps of CDN availability, …
- Adding some *pop* to the page

I want to start embedding widgets now!

How do I make it happen?

1. Create a widget

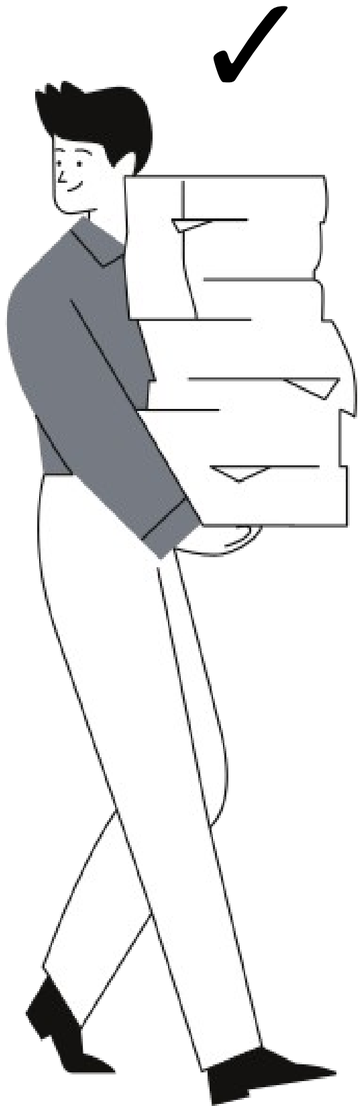
From a technical perspective a widget is a function that takes a DOM id and renders JS in it.

A widget can also receive arguments as HTML data attributes.

```
window.renderExampleWidget = function (instanceId) {
 const element = document.getElementById(instanceId);
 const title = element.getAttribute('data-button-text');
 ReactDOM.render(
  <Widget title={title} />,
  element,
 );
};
```

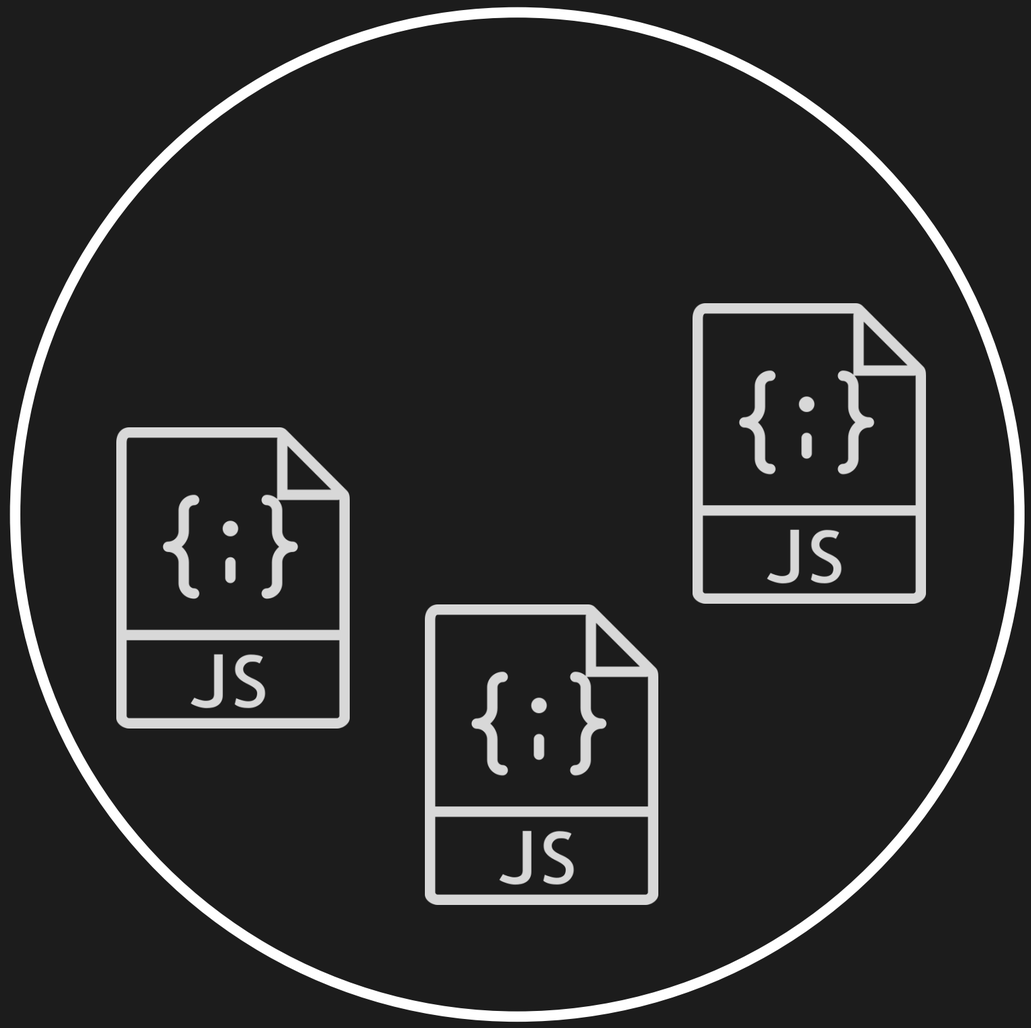
Very easy to port

(1) take any component,
(2) render it.
That's it. Yes, really.



2. Upload the app code

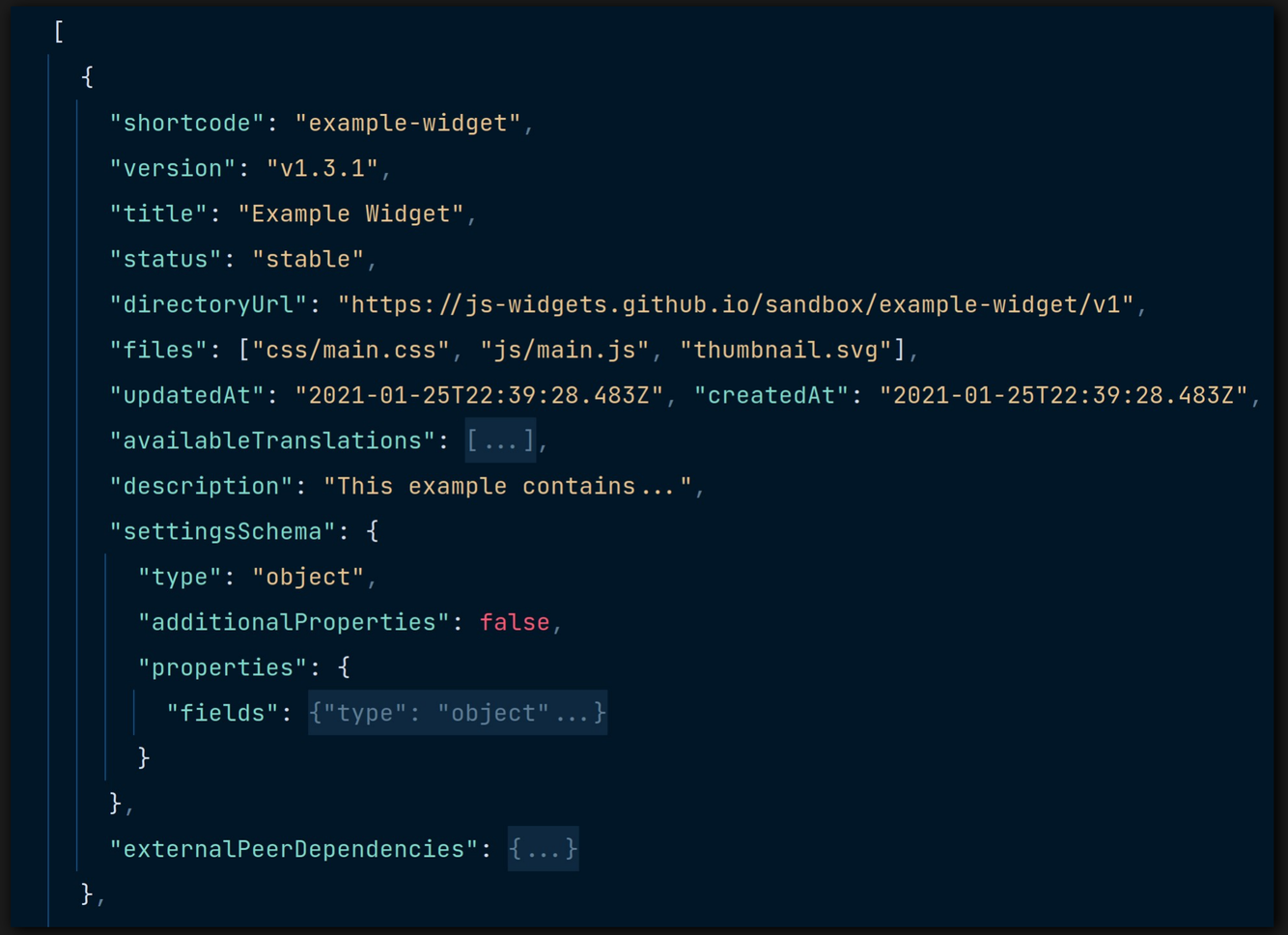
The CMS can use this to either download the files, or serve them from there.



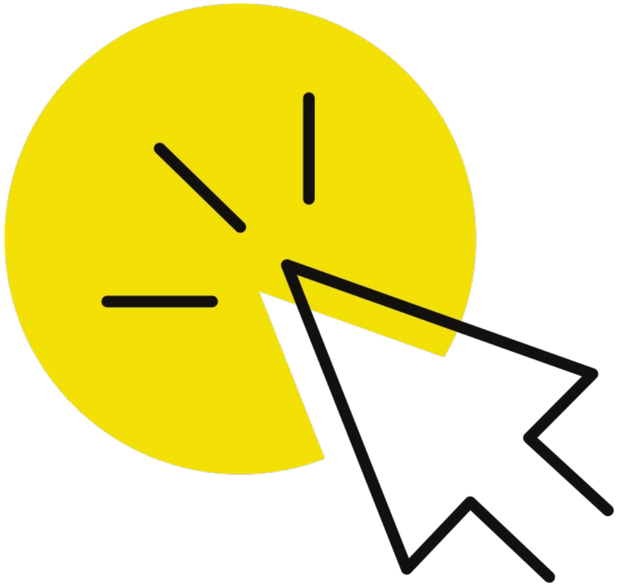
3. Publish the metadata

Otherwise this is just another JS app in some repo.

This is just a JSON document containing all the metadata about all the available apps, available via the Internet.



CMS pulls that JSON...



and that's it!

Widget Type *

Example Widget

- Select a value -
Example Emojis
Example Calculator
Example Widget

gested from the widget_instance registry.

Button text

Some random string to be displayed when the widget is rendered.

Each widget can have different settings defined in the widget registry. Provide the necessary values here.

Save

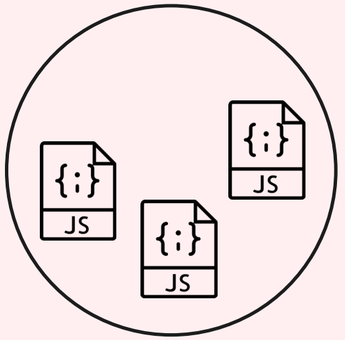
Summary

Adding widgets to your CMS is a matter of 3 simple steps.

1. Create the widget

```
window.renderExampleWidget = function (instanceId) {
  const element =
document.getElementById(instanceId);
  const title = element.getAttribute('data-button-text');
  ReactDOM.render(
    <Widget title={title} />,
    element,
  );
};
```

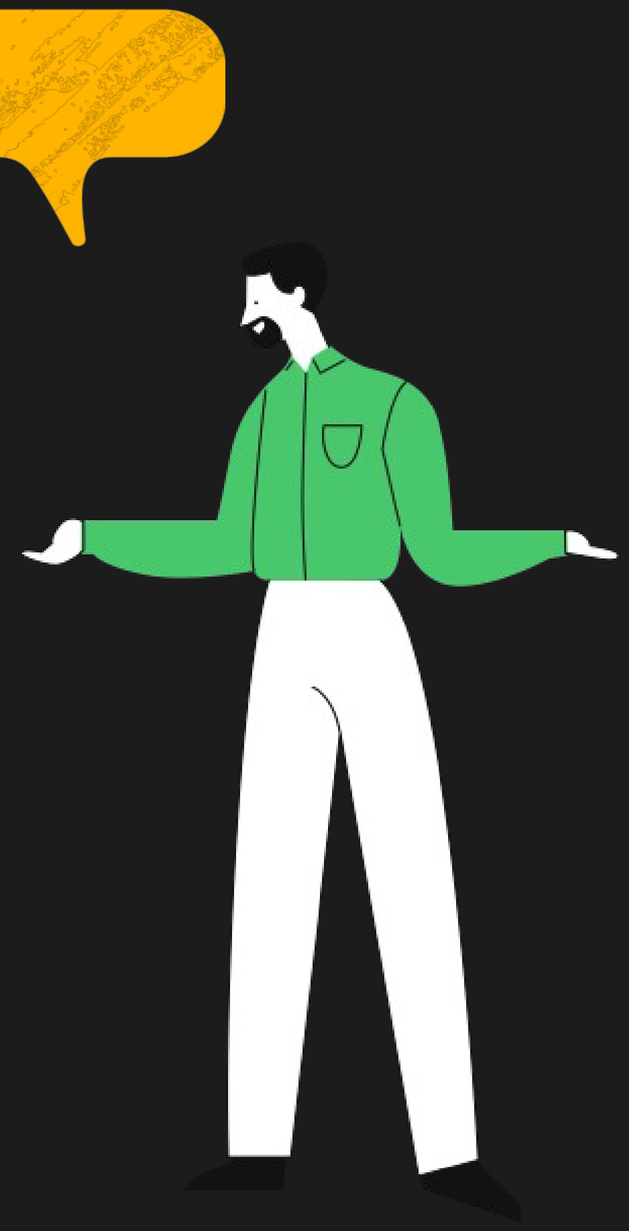
2. Upload the widget

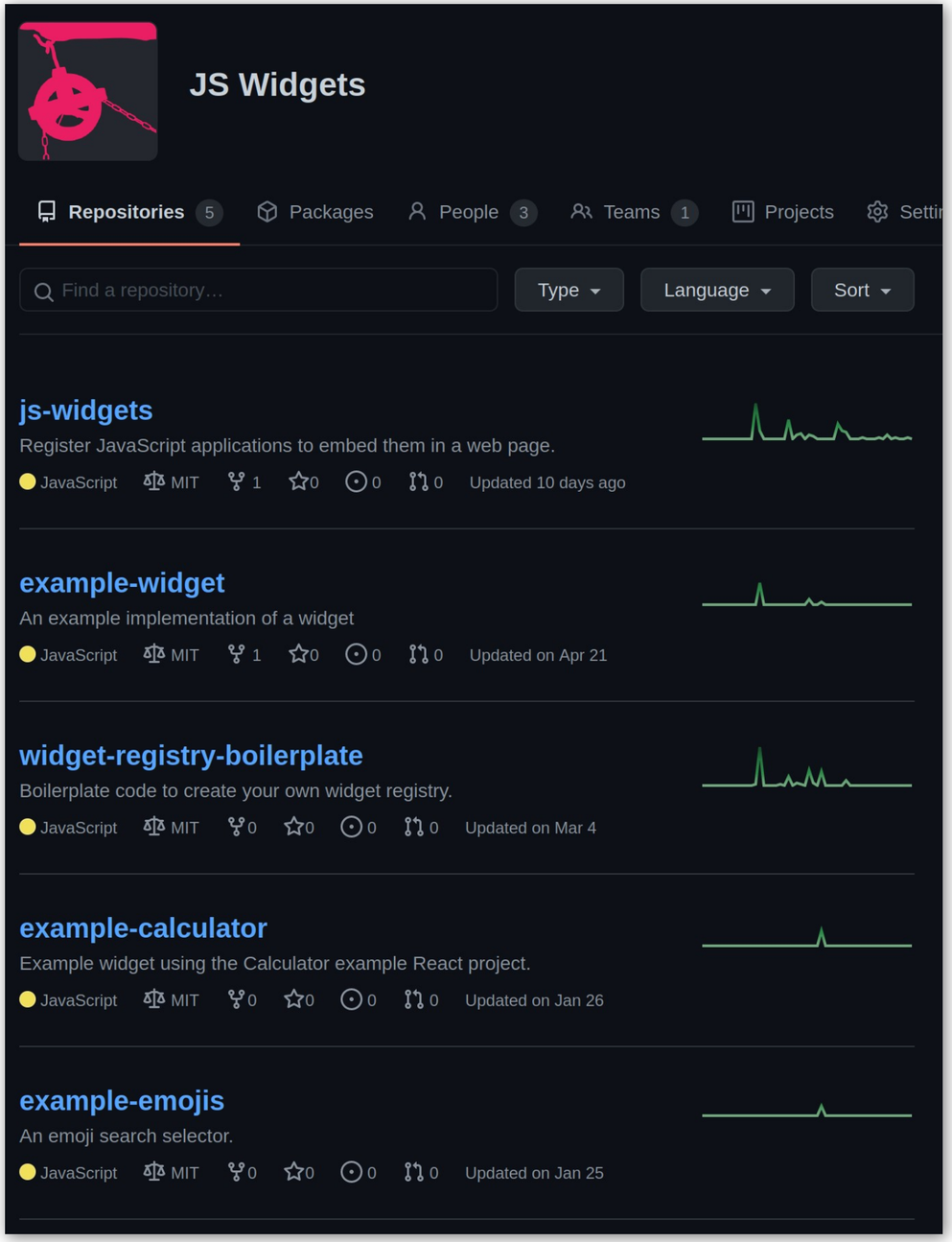


3. Publish metadata


```
[
  {
    "shortcode": "example-widget",
    "version": "v1.3.1",
    "title": "Example Widget",
    "status": "stable",
    "directoryUrl": "https://js-widgets.github.io/sandbox/example-widget/v1",
    "files": ["css/main.css", "js/main.js", "thumbnail.svg"],
    "updatedAt": "2021-01-25T22:39:28.483Z", "createdAt": "2021-01-25T22:39:28.483Z",
    "availableTranslations": [...],
    "description": "This example contains...",
    "settingsSchema": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "fields": {"type": "object"...}
      }
    },
    "externalPeerDependencies": {...}
  },
```

I still don’t know where to start...





Lots of existing tooling

- Widget examples
- Widget registry boilerplate & catalog
- With CI/CD integration built in

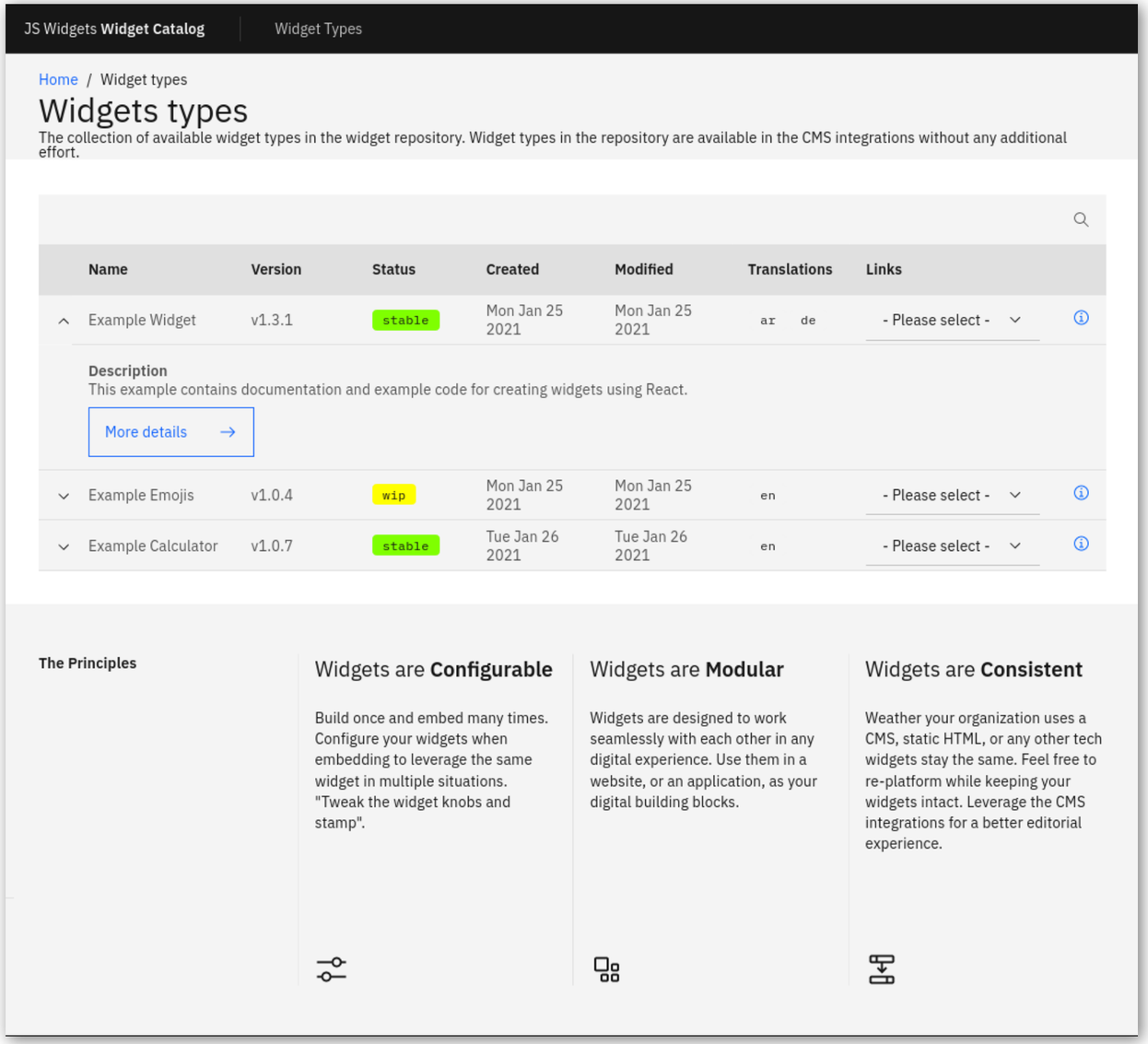
`https://github.com/js-widgets`

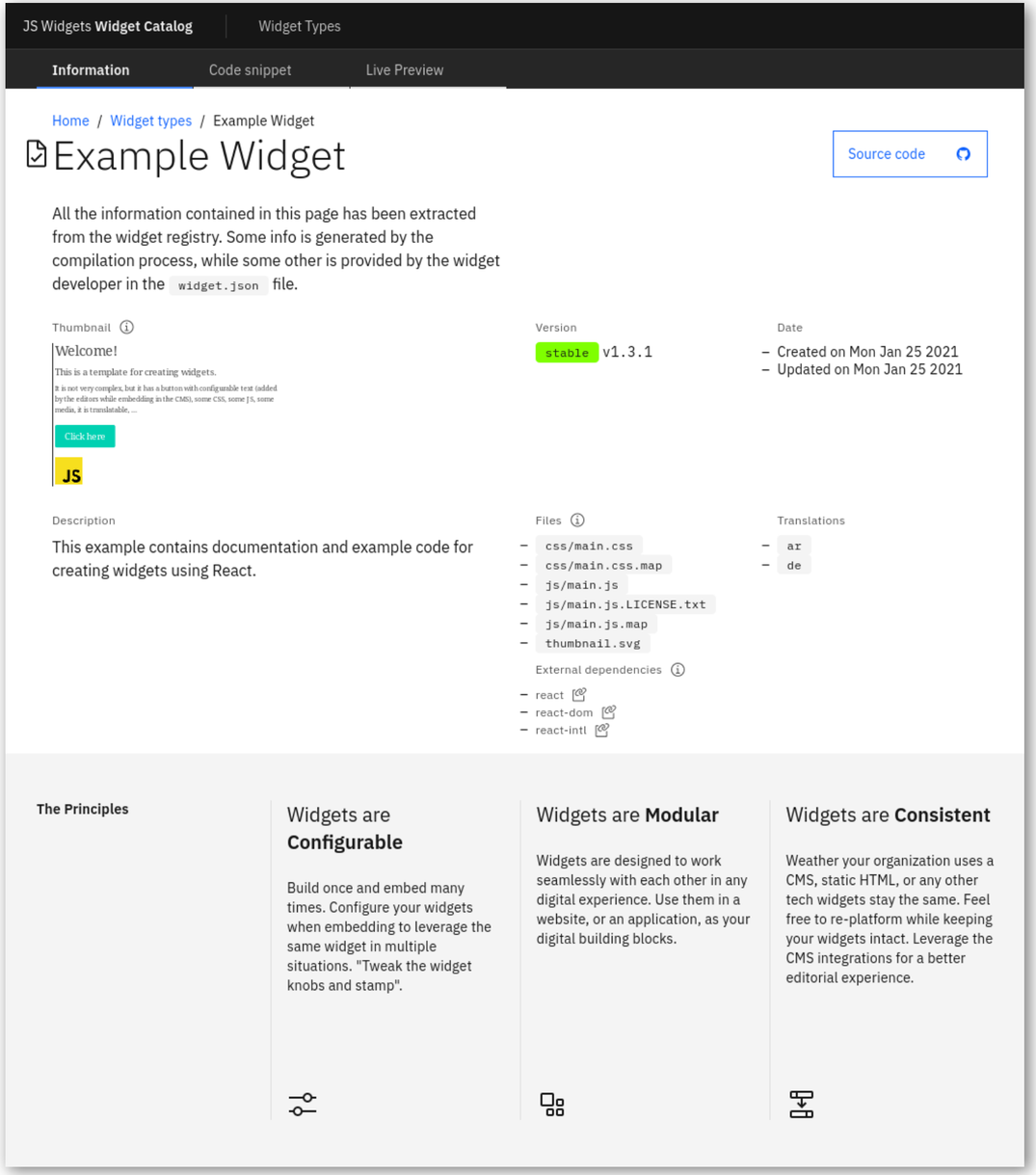
Stakeholder ready

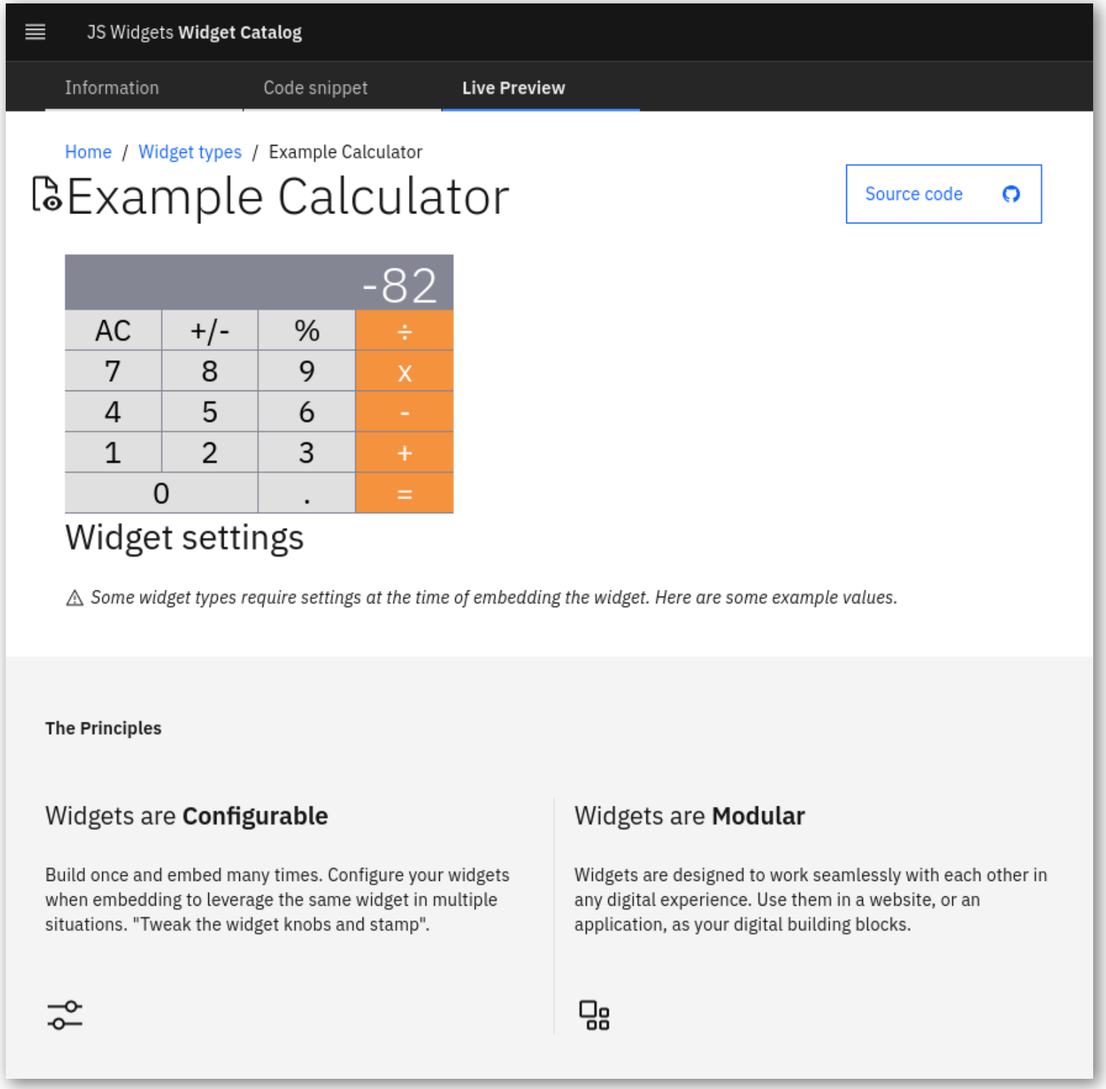
Show the progress on the JS apps to stakeholders soon. Also allow editors to browse all available widgets easily.

The widget catalog

The widget registry reads from the widget metadata and creates a static site catalog out-of-the-box. **Zero effort**.




JS Widgets Widget Catalog
Widget Types
Information
Code snippet
Live Preview
Home / Widget types / Example Widget
Example Widget
Source code
All the information contained in this page has been extracted from the widget registry. Some info is generated by the compilation process, while some other is provided by the widget developer in the widget.json file.
Thumbnail
Welcome!
This is a template for creating widgets.
It is not very complex, but it has a button with configurable text (added by the editors while embedding in the CMS), some CSS, some JS, some media, it is translatable, ...
Click here
JS
Version
stable v1.3.1
Date
Created on Mon Jan 25 2021
Updated on Mon Jan 25 2021
Description
This example contains documentation and example code for creating widgets using React.
Files
css/main.css
css/main.css.map
js/main.js
js/main.js.LICENSE.txt
js/main.js.map
thumbnail.svg
External dependencies
react
react-dom
react-intl
Translations
ar
de
The Principles
Widgets are Configurable
Build once and embed many times. Configure your widgets when embedding to leverage the same widget in multiple situations. "Tweak the widget knobs and stamp".
Widgets are Modular
Widgets are designed to work seamlessly with each other in any digital experience. Use them in a website, or an application, as your digital building blocks.
Widgets are Consistent
Weather your organization uses a CMS, static HTML, or any other tech widgets stay the same. Feel free to re-platform while keeping your widgets intact. Leverage the CMS integrations for a better editorial experience.


JS Widgets Widget Catalog
Information
Code snippet
Live Preview
Home / Widget types / Example Calculator
Example Calculator
Source code
-82
AC +/- % ÷
7 8 9 x
4 5 6 -
1 2 3 +
0 . =
Widget settings
Some widget types require settings at the time of embedding the widget. Here are some example values.
The Principles
Widgets are Configurable
Build once and embed many times. Configure your widgets when embedding to leverage the same widget in multiple situations. "Tweak the widget knobs and stamp".
Widgets are Modular
Widgets are designed to work seamlessly with each other in any digital experience. Use them in a website, or an application, as your digital building blocks.

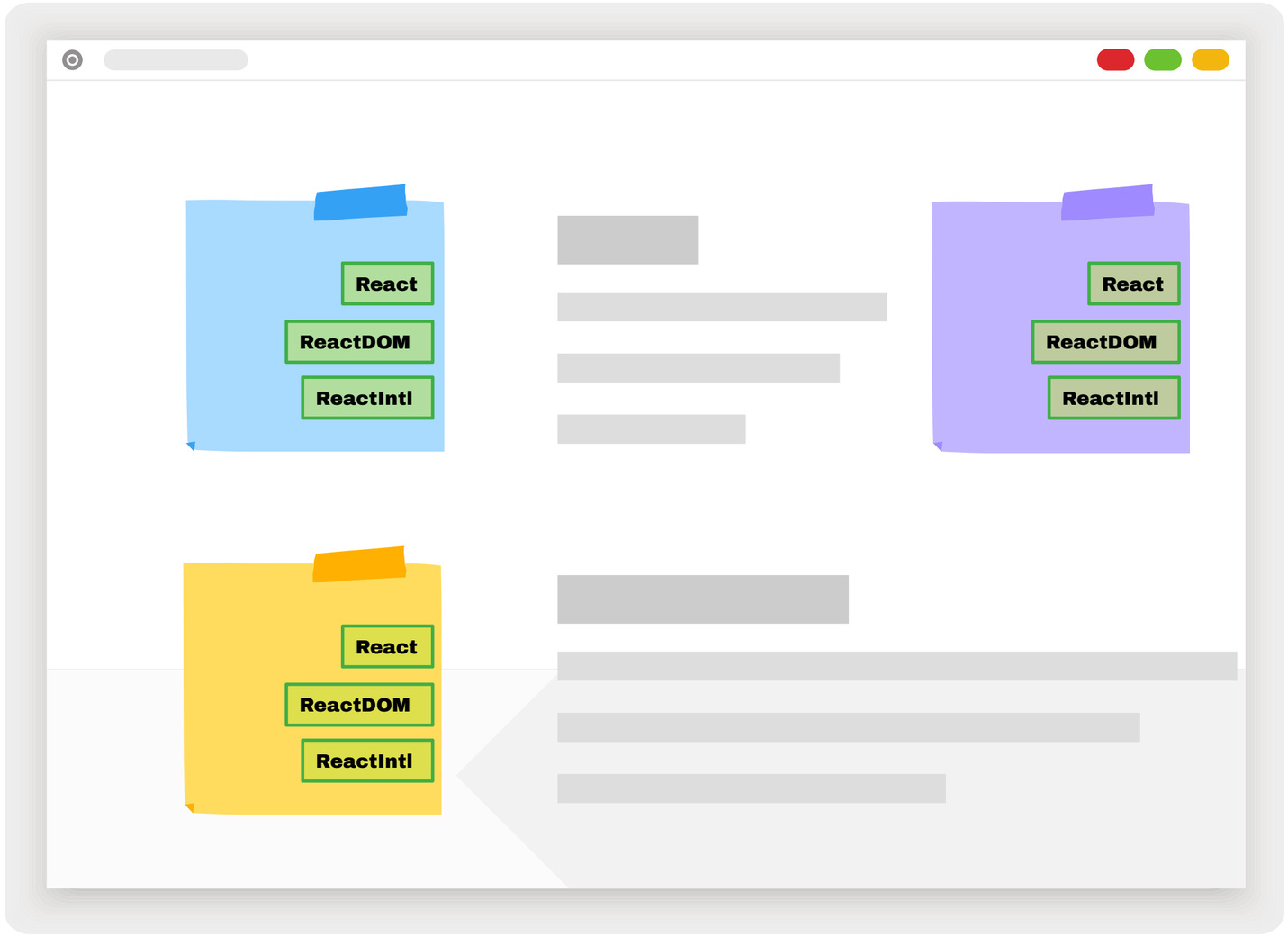
Governance like you need it

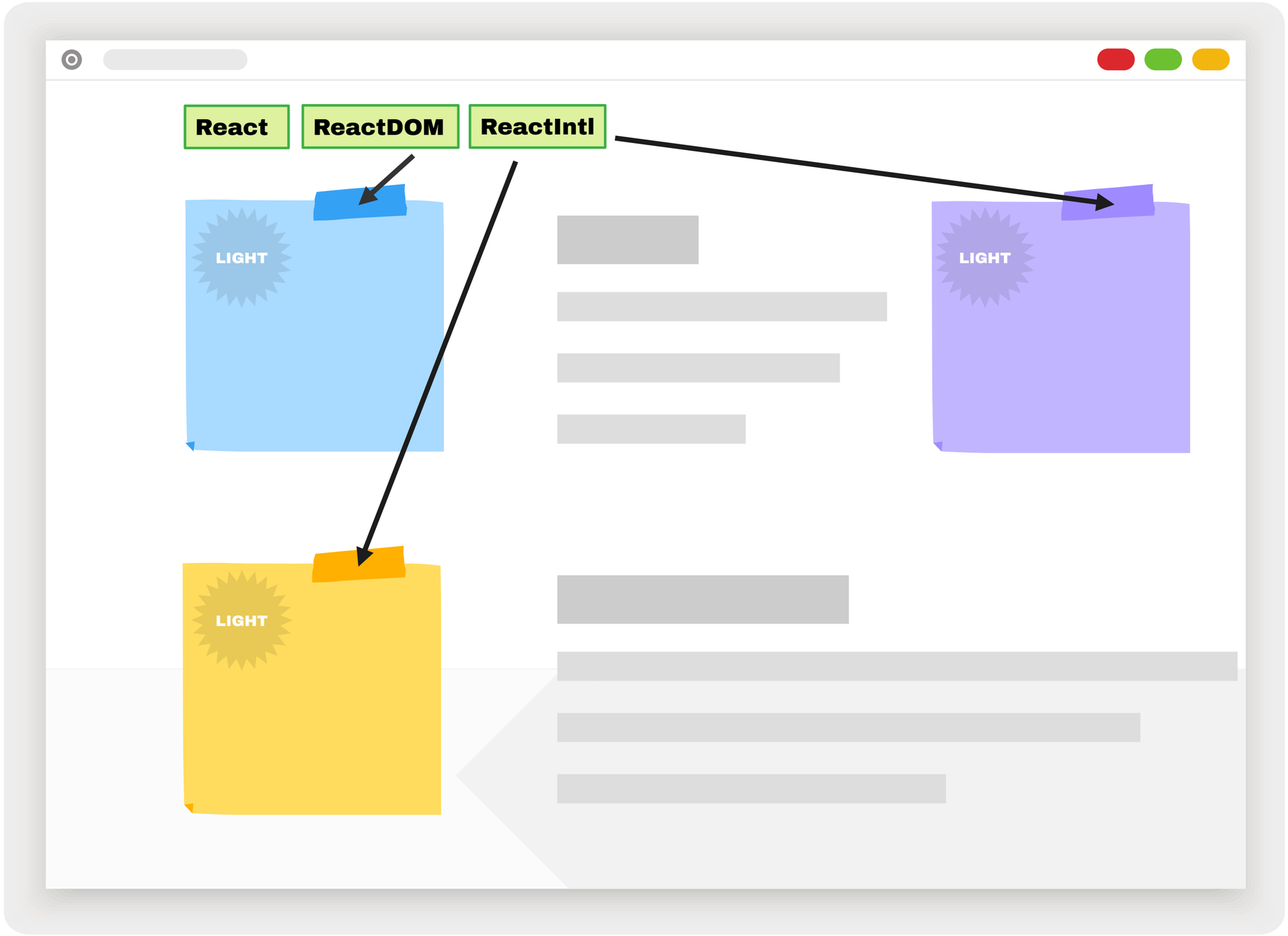
You decide what registries to accept into **your** CMS 1

You decide what widgets and updates go into **your** registry 2

Production ready

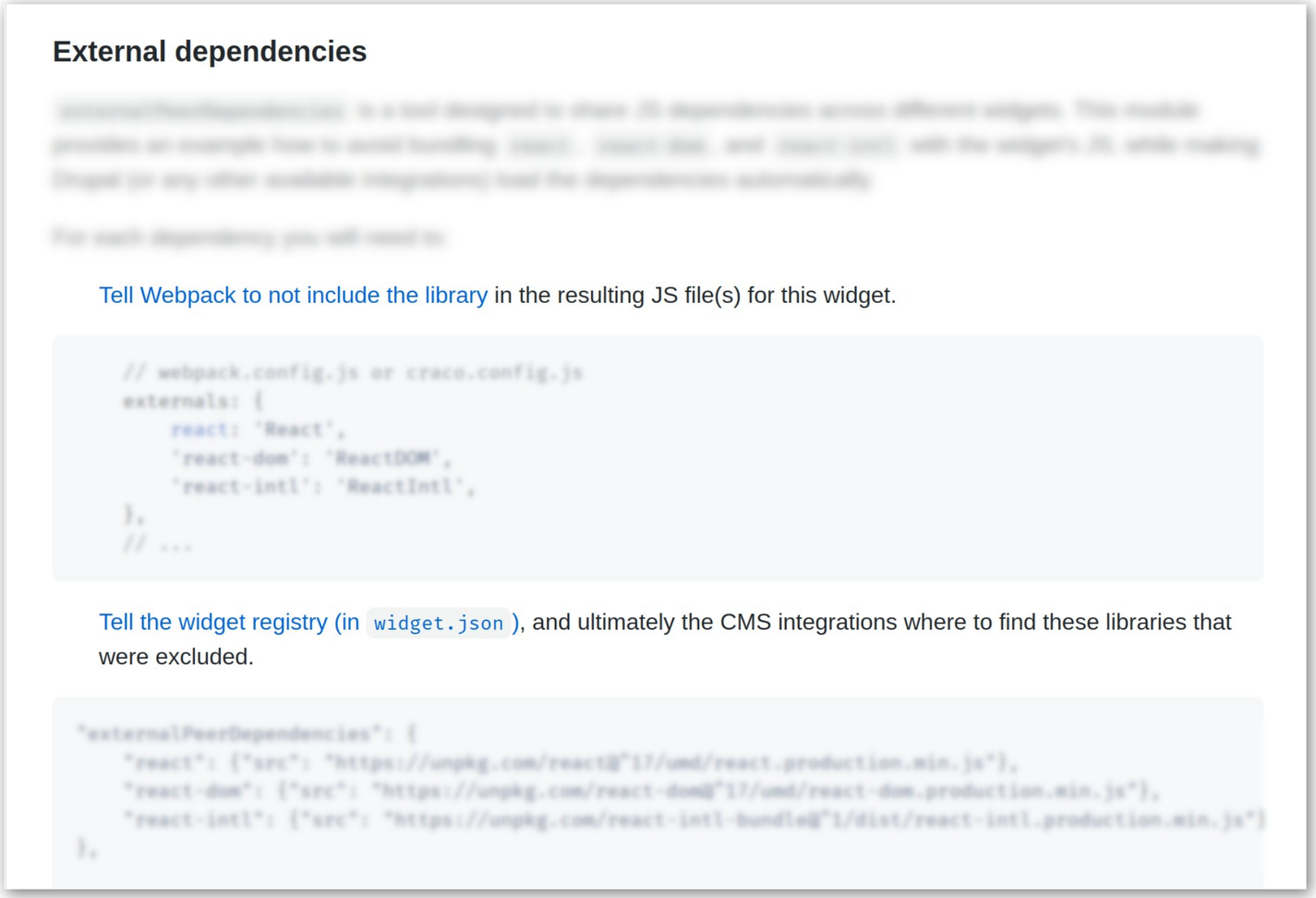
Battle tested in production with many out-of-the-box optimizations.




React
ReactDOM
ReactIntl
LIGHT
LIGHT
LIGHT

1. Exclude shared dependencies from all JS bundles via webpack

2. Have the CMS provide the deps as the bundles expect



External dependencies

[blurred] is a tool designed to share JS dependencies across different widgets. This module provides an example how to avoid bundling `react`, `react-dom`, and `react-intl` with the widget's JS, while making Drupal (or any other available integrations) load the dependencies automatically.

For each dependency you will need to:

Tell Webpack to not include the library in the resulting JS file(s) for this widget.

```
// webpack.config.js or craco.config.js
externals: {
    react: 'React',
    'react-dom': 'ReactDOM',
    'react-intl': 'ReactIntl',
},
// ...
```

Tell the widget registry (in `widget.json`), and ultimately the CMS integrations where to find these libraries that were excluded.

```
"externalPeerDependencies": {
    "react": {"src": "https://unpkg.com/react@17/umd/react.production.min.js"},
    "react-dom": {"src": "https://unpkg.com/react-dom@17/umd/react-dom.production.min.js"},
    "react-intl": {"src": "https://unpkg.com/react-intl-bundle@1/dist/react-intl.production.min.js"}
},
```

https://github.com/js-widgets/example-widget#external-dependencies

Documentation available

https://video.mateuaguilo.com

Thank you!