

CSS Modules + Internationalization for Embroider

github.com/ijlee2

terms

- *CSS modules - a way to colocate styles and prevent leaks*
- *Internationalization - a way to help users interact with an app in their preferred language*
- *Embroider project - an app or addon (v2) that is built using @embroider/* packages*
- *Bleeding-edge Ember - an Embroider app that enabled every `static*` flag, route splitting, `glint`, and `<template>` tag*

outcomes

- *Style components in v2 addons*
- *Provide translations from v2 addons*
- *Refactor code*
- *Get sneak previews*

1a

embroider- css-modules

latest version 2.0.5

goals

- *Provide a smooth transition from ember-css-modules*
- *Shift the “how” from Ember to JavaScript*
- *Minimize cost to run on bleeding-edge Ember*
- *Delegate improving DX to glint, stylelint*

1b

ember-intl

latest version 6.5.3

goals

- *Provide a smooth transition from ember-intl (v1 addon)*
- *Shift the “how” from Ember to JavaScript*
- *Minimize cost to run on bleeding-edge Ember*
- *Delegate improving DX to glint, eslint-plugin-ym1, ember-intl-analyzer*

1c

**live
demo**

● ● ● File: addon/components/contracts/add-contract-button.hbs

```
1 <Ui::Button @onClick={{@onClick}} local-class="button">
2   <span>
3     <Ui::Icon @source="plus" local-class="icon" />
4   </span>
5
6   <span local-class="label">
7     {{t "contracts.add-contract-button.label"}}
8   </span>
9 </Ui::Button>
```

● ● ● File: addon/components/contracts/add-contract-button.ts

```
1 import templateOnlyComponent from '@ember/component/template-only';
2
3 interface ContractsAddContractButtonSignature {
4   Args: {
5     onClick: () => void;
6   };
7 }
8
9 const ContractsAddContractButtonComponent =
10   templateOnlyComponent<ContractsAddContractButtonSignature>();
11
12 export default ContractsAddContractButtonComponent;
```

2

refactors

File: addon/components/hello.hbs

```
1 <div
2   local-class="
3     message
4     hide
5   "
6 >
7   Hello!
8 </div>
```

File: src/components/hello.hbs

```
1 <div
2   class={{concat
3     this.styles.message
4     " "
5     this.styles.hide
6   }}
7 >
8   Hello!
9 </div>
```

● ● ● File: src/components/hello.hbs

```
1 <div
2   class={{local
3     this.styles
4     "message"
5     "hide"
6   }}
7 >
8   Hello!
9 </div>
```

● ● ● File: src/components/hello.hbs

```
1 <div
2   class={{local
3     this.styles
4     "message"
5     (if this.someCondition "hide")
6   }}
7 >
8   Hello!
9 </div>
```

● ● ● File: src/components/hello.hbs

```
1 <div
2   class={{local
3     this.styles
4     "message"
5     (if this.someCondition (array "hide" "after-3-sec"))
6   }}
7 >
8   Hello!
9 </div>
```

codemod

Get started:

```
npx ember-codemod-remove-ember-css-modules  
--type v2-addon
```

latest version 2.0.1

File: src/components/list.ts

```
1  import Component from '@glimmer/component';
2
3  import styles from './list.css';
4
5  type ItemSpace = 'loose' | 'normal' | 'tight';
6
7  interface ListSignature {
8    Args: {
9      itemSpace?: ItemSpace;
10   };
11   Blocks: {
12     default: [];
13   };
14 }
15
16 export default class ListComponent extends Component<ListSignature> {
17   styles = styles;
18
19   get itemSpace() {
20     return this.args.itemSpace ?? 'normal';
21   }
22 }
```

● ● ● File: src/components/list.hbs

```
1 <ul
2   class={{local
3     this.styles
4     "list"
5     (concat 'item-space-' this.itemSpace)
6   }}
7 >
8   {{yield}}
9 </ul>
```

● ● ● File: src/components/list.hbs

```
1 <ul
2   class={{local
3     this.styles
4     "list"
5     this.itemSpaceClass
6   }}
7 >
8   {{yield}}
9 </ul>
```

File: src/components/list.ts

```
1  import Component from '@glimmer/component';
2
3  import styles from './list.css';
4
5  type ItemSpace = 'loose' | 'normal' | 'tight';
6
7  interface ListSignature { /* ... */ }
8
9  export default class ListComponent extends Component<ListSignature> {
10    styles = styles;
11
12    get itemSpaceClass() {
13      switch (this.args.itemSpace) {
14        case 'loose': {
15          return 'item-space-loose';
16        }
17
18        case 'tight': {
19          return 'item-space-tight';
20        }
21
22        default: {
23          return 'item-space-normal';
24        }
25      }
26    }
27  }
```

● ● ● File: src/components/list.ts

```
1  import Component from '@glimmer/component';
2
3  import styles from './list.css';
4
5  const itemSpaceClasses = {
6    loose: 'item-space-loose',
7    normal: 'item-space-normal',
8    tight: 'item-space-tight',
9  } as const;
10
11  type ItemSpace = keyof typeof itemSpaceClasses;
12
13  interface ListSignature { /* ... */ }
14
15  export default class ListComponent extends Component<ListSignature> {
16    styles = styles;
17
18    get itemSpaceClass() {
19      const itemSpace = this.args.itemSpace ?? 'normal';
20
21      return itemSpaceClasses[itemSpace];
22    }
23  }
```

● ● ● File: src/components/list.ts

```
1  import Component from '@glimmer/component';
2
3  import styles from './list.css';
4
5  type ItemSpace = 'loose' | 'normal' | 'tight';
6
7  interface ListSignature { /* ... */ }
8
9  export default class ListComponent extends Component<ListSignature> {
10     styles = styles;
11
12     get itemSpaceClass() {
13         const itemSpace = this.args.itemSpace ?? 'normal';
14
15         return `item-space-${itemSpace}` as const;
16     }
17 }
```

● ● ● File: tests/integration/components/contracts/add-contract-button-test.ts

```
1  import { setupIntl, t } from 'ember-intl/test-support';
2
3  module(
4    'Integration | Component | contracts/add-contract-button',
5    function (hooks) {
6      setupRenderingTest(hooks);
7      setupIntl(hooks);
8
9      test('it renders', async function (assert) {
10        this.onClick = () => {};
11
12        await render(hbs`
13          <Contracts::AddContractButton
14            @onClick={{this.onClick}}
15          />
16        `);
17
18        assert
19          .dom('[data-test-button]')
20          .hasText(t('contracts.add-contract-button.label'));
21      });
22    },
23  );
```

● ● ● File: tests/integration/components/contracts/add-contract-button-test.ts

```
1  import { setupIntl } from 'ember-intl/test-support';
2
3  module(
4    'Integration | Component | contracts/add-contract-button',
5    function (hooks) {
6      setupRenderingTest(hooks);
7      setupIntl(hooks, 'de-de');
8
9      test('it renders', async function (assert) {
10        this.onClick = () => {};
11
12        await render(hbs`
13          <Contracts::AddContractButton
14            @onClick={{this.onClick}}
15          />
16        `);
17
18        assert
19          .dom('[data-test-button]')
20          .hasText('Vertrag hinzufügen');
21      });
22    },
23  );
```

3

**what's
next?**

ember-intl

- *v7-beta in Q2 2024*
- *Update tests and documentations*
- *Fix type errors from v5*
- *CTA: Help migrate to v2 format (Broccoli)*

embroider-css-modules

- *Quite stable for Embroider apps with Webpack*
- *Quite stable for v2 addons*
- *Non-Embroider projects can use `ember-css-modules`*
- *Support Embroider apps with Vite*

ViteApp

localhost:4200

Elements

Console

Sources

<!DOCTYPE html>
<html data-ember-extension="1">
 <head>
 <script src="/assets/vendor.js"></script>
 <script src="/assets/vite-app.js?t=1710508195843" type="module"></script>
 <div class="embroider-css-modules__0gS5i">
 <div class="embroider-css-modules__2lpSU embroider-css-modules__ZI7ws"> Hello Vite! </div>
 </div>
 </body>
</html>
...</div>
div.embroider-css-modules__2lpSU.embroider-css-modules__ZI7ws</div>
Filter: :hov .cls + -
element.style {
}
.embroider-css-modules__ZI7ws {<style>
 color: white;
 font-size: 8rem;
 font-style: italic;
 letter-spacing: 0.01rem;
}
.embroider-css-modules__2lpSU {<style>
 align-items: center;
 display: flex;
 height: 100%;
 justify-content: center;
}
* { vite-app.css:12

Hello Vite!

more info

- [*https://crunchingnumbers.live/2023/07/20/in-1-year*](https://crunchingnumbers.live/2023/07/20/in-1-year)
- [*https://github.com/ember-intl/ember-intl*](https://github.com/ember-intl/ember-intl)
- [*https://github.com/ijlee2/embroider-css-modules*](https://github.com/ijlee2/embroider-css-modules)