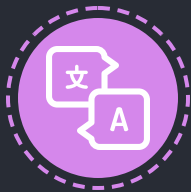


I18N WITHOUT PAIN

Anton Lyaptsev

GIVEN



5 languages

Translation management system (TMS)



A lot of legacy

WORKFLOW

Design → TMS → Code → Production

WORKFLOW

Design



Production



PROBLEM



Developers



Translators



Other staff

FIRST APPROACH

VCS-like TMS

FIRST APPROACH WORKFLOW

Design → TMS-branch → Code-branch → Merge → Production

+ CLI

FIRST APPROACH WORKFLOW

Design → TMS-bran → Merge → Production





BUT REACT HAPPENED...

NEW PROBLEMS

1. Incompatibility with React
2. Custom message formats

```
1 {  
2   "key.simple": "Show this {text}",  
3   "key.xml": "Show this <t:value>text</t:value>"  
4 }
```

NEW PROBLEMS

1. Incompatibility with React
2. Custom message formats
3. Dynamic keys

```
1 const statusType = data.status;  
2  
3 // hint: status.(started/processing/done)  
4 const statusName = i18n.t(`status.${statusType}`);
```

REQUIREMENTS

1. JS-api and React components
2. Isomorphic
3. Independent message format
4. Dynamic keys => static keys
5. Extracting keys

BONUS REQUIREMENT

Keys is not a developers responsibility

SECOND APPROACH

js-lingui

i18next

COMPARISON

	i18next	js-lingui
JS API	✓	✓
React API	plugin	✓
Message formats	plugin	ICU/PO
Dynamic keys	plugin	✓ (select)
Extraction	✓	✓
Size (gzipped)	10.1 kB	1.4 kB core
	+ plugins	2.5 kB react

```
1 i18n._({ id: 'keys.static' });  
2  
3 const name = 'Fred'  
4 i18n._({ id: 'keys.variable', values: { name } });  
5  
6 const count = 5;  
7 i18n._({ id: 'keys.plural', values: { count } });  
8  
9 const status = 'paid';  
10 i18n._({ id: 'keys.select', values: { status } });
```

```
1 const name = 'Fred';
2 const count = 5;
3 const status = 'paid';
4
5 <Trans id="keys.static" />
6
7 <Trans id="keys.variable" values={{ name }} />
8
9 <Trans id="keys.plural" values={{ count }} />
10
11 <Trans id="keys.select" values={{ status }} />
```

BONUS REQUIREMENT

babel-plugin-macros

Runtime transformations without babel plugin

KEYLESS APPROACH

```
1 import { t } from '@lingui/macro';
2
3 const name = 'Fred';
4
5 i18n._(/*i18n: some description*/t`My name id ${ name }`);
6 // ↓ ↓ ↓ ↓ ↓ ↓
7 i18n._(/*i18n: some description*/{
8   id: 'My name id {name}',
9   defaults: 'My name id {name}',
10  values: {
11    name: name
12  }
13 });
```



```
1 import { plural } from '@lingui/macro';
2
3 i18n._(/*i18n: some description*/plural({
4   value: count,
5   one: '# book',
6   other: '# books'
7 }));
8 // ↓ ↓ ↓ ↓ ↓ ↓
9 i18n._(/*i18n: some description*/{
10   id: '{count, plural, one {# book}, other {# books}}',
11   defaults: '{count, plural, one {# book}, other {# books}}',
12   values: {
13     count: count
14   }
15 });
```

```
1 import { select } from '@lingui/macro';
2
3 i18n._(/*i18n: some description*/select({
4   value: status,
5   paid: `Paid at ${paidAt}`,
6   canceled: `Canceled by ${userId}`
7 }));
8 // ↓ ↓ ↓ ↓ ↓ ↓
9 i18n._(/*i18n: some description*/{
10   id: '{status, select, paid {Paid at {paidAt}} ... ',
11   defaults: '{status, select, paid {Paid at {paidAt}} ... ',
12   values: {
13     status: status,
14     paidAt: paidAt,
15     canceled: canceled
16   }
17 });
```

```
1 import { Trans, Plural, Select } from '@lingui/macro';
2
3 <Trans>My name is {name}</Trans>
4
5 <Plural
6   id="keys.plural"
7   value={count}
8   one="# book"
9   other="# books"
10 />
11
12 <Select
13   value={status}
14   paid={<Trans>Paid at {paidAt}</Trans>}
15   canceled={<Trans>Canceled by {userId}</Trans>}
16 />
```

```
1 {  
2   "My name is {name}": {  
3     "translation": "Меня зовут {name}",  
4     "defaults": "My name is {name}",  
5     "description": "some description",  
6     "origin": [["src/App.jsx", 5]]  
7   }  
8 }
```

KEYLESS CAVEAT

New text is new key

WORKFLOW

Design → Code → Production

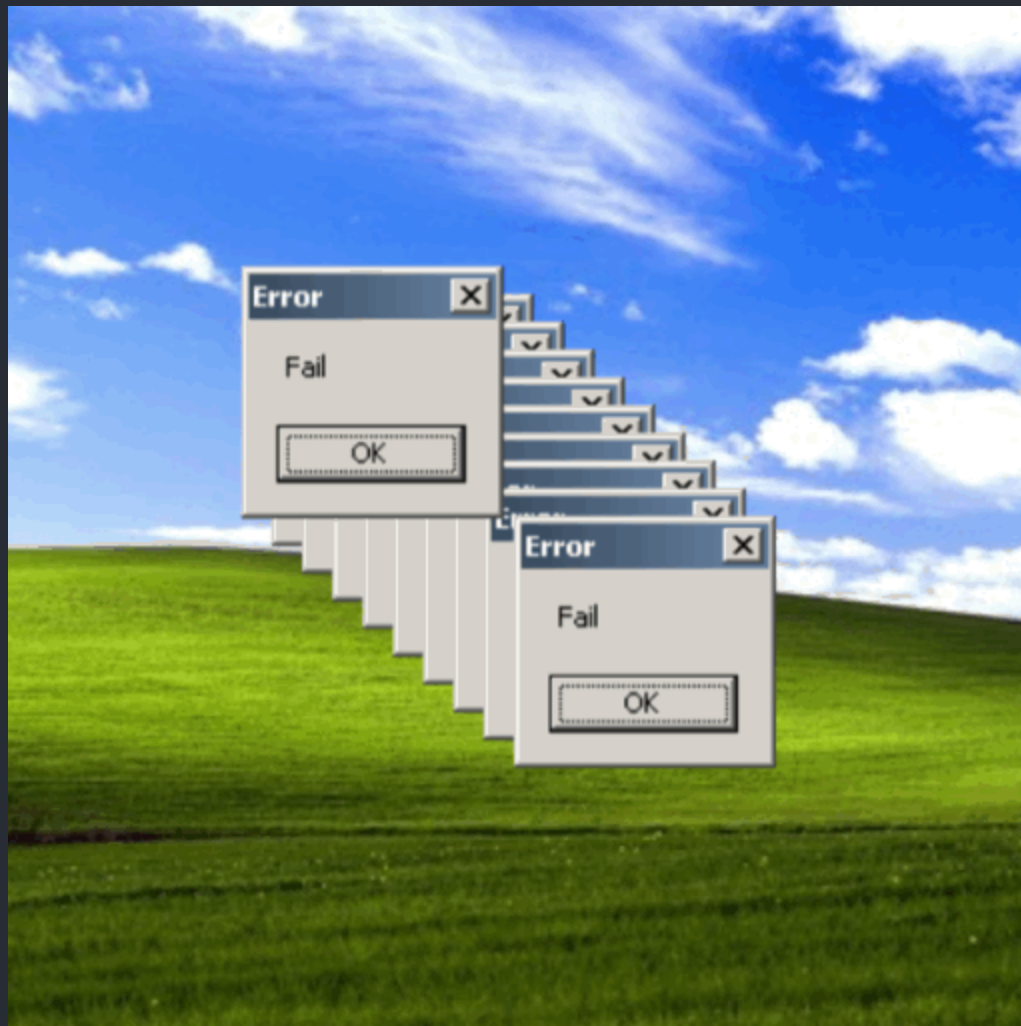
+ sync with TMS in **CI**

WORKFLOW






```
1 // FirstComponent.jsx
2 const FirstComponent = () => (
3   <button>
4     <Trans>
5       Click me!
6     </Trans>
7   </button>
8 )
9
10 // SecondComponent.jsx
11 const SecondComponent = () => (
12   <button>
13     <Trans>
14       Click me!
15     </Trans>
16   </button>
17 )
```



WHAT IF WE COULD CONTROL KEY GENERATION

```
1 // linguist.config.js
2 module.exports = {
3   extractId: function (notUniqID, file) {
4     // notUniqID can be passed id or default translation
5     const uniqID = notUniqID + file.path;
6
7     return uniqID;
8   }
9 };
```



IT'S NOT A SUCCESS STORY



AS A RESULT

1. You can forget about working with TMS
2. js-lingui is a good lightweight i18n-lib
3. Keyless approach is good if default translations is ok for your users
4. Keyless approach is good if your i18n-lib has custom key generation support

CONTACTS



Github: github.com/alyaptsev

Telegram: [@antonlyaptsev](https://t.me/antonlyaptsev)