

title: 多语言支持  
date: 2019-10-09  
tag: React, i18n

## description: 网站支持多个语言切换

使用 [react-intl](#) 提供翻译支持。

## 安装

```
yarn add react-intl -D
```

## 使用思路

1. 使用 `localStorage` 缓存当前语言，并在切换时 `reload` 到更新语言的界面；
2. 使用 `IntlProvider`，`FormattedMessage` 两个组件达到更新后新的文本在各个组件之间的渲染。参照示例：

```
// App.jsx
import React from 'react';
import ReactDOM from 'react-dom';
import { IntlProvider, FormattedMessage } from 'react-intl';

const App = () => {
  return (
    <div>
      <FormattedMessage id="welcome"/>
    </div>
  )
}

export default App;

// index.jsx
ReactDOM.render(
  <IntlProvider locale="en" messages={{ welcome: 'Welcome !' }}>
    <App />
  </IntlProvider>,
  document.getElementById('container')
);
```

## 具体实现

相关 [pr](#)

## 对应语言的翻译

在 `locales` 下面添加 `en-US`、`zh-CN` 两个文件，并在其中补充键值对翻译，翻译颗粒度细化到组件级别，例如 `SideBar` 对应英文翻译：

```
export default {
  'app.sidebar.overview': 'Overview',
  'app.sidebar.nav1': 'Navigation-1',
  'app.sidebar.basic': 'Basic',
  'app.sidebar.list': 'List',
}
```

`LocaleProvider` 提供多语言翻译

通过 `IntlProvider` 的 `locale` 和 `message` 提供语言以及对应的翻译：

```
import React from 'react';
import { IntlProvider } from 'react-intl';
import zh_CN from '../locales/zh-CN';
import en_US from '../locales/en-US';
const localeInfo = {
  'zh-CN': {
    appLocale: 'zh',
    appMessages: zh_CN,
  },
  'en-US': {
    appLocale: 'en',
    appMessages: en_US,
  },
};

export default props => {
  const { locale, children } = props;
  const myLocale = localeInfo[locale];
  return (
    <IntlProvider locale={myLocale.appLocale} messages={myLocale.appMessages}>
      {React.Children.only(children)}
    </IntlProvider>
  );
}
```

## 文本使用 `FormattedMessage` 组件替换

`FormattedMessage` 替换原有文本，比如：

```
<FormattedMessage id="app.sidebar.overview"/>
```