

API

路由

umi/link

通过声明的方式做路由跳转。

例子：

```
import Link from 'umi/link';

export default () => {
  <div>
    /* 普通使用 */
    <Link to="/list">Go to list page</Link>

    /* 带参数 */
    <Link to="/list?a=b">Go to list page</Link>

    /* 包含子组件 */
    <Link to="/list?a=b"><button>Go to list page</button></Link>
  </div>
}
```

html

umi/router

通过编程的方式做路由切换，包含以下 4 个 API。

router.push(path)

推一个新的页面到 history 里。

例子：

```
import router from 'umi/router';
```

js

≡ UmiJS

```
// 带参数
router.push('/list?a=b');
router.push({
  pathname: '/list',
  query: {
    a: 'b',
  },
});
# 对象且不包含 pathname 会报错
router.push({
  query: {}
});
```

router.replace(path)

替换当前页面，参数和 [router.push\(\)](#) 相同。

router.go(n)

往前或往后跳指定页数。

例子：

```
import router from 'umi/router';

router.go(-1);
router.go(2);
```

js

router.goBack()

后退一页。

例子：

```
import router from 'umi/router';

router.goBack();
```

js

≡ UmiJS

umi/navlink

详见: <https://reacttraining.com/react-router/web/api/NavLink> 

umi/redirect

重定向用。

例子:

```
import Redirect from 'umi/redirect';  
<Redirect to="/login" />;
```

js

详见: <https://reacttraining.com/react-router/web/api/Redirect> 

umi/prompt

例子:

```
import Prompt from 'umi/prompt';  
  
export default () => {  
  return (  
    <>  
      <h1>Prompt</h1>  
      <Prompt  
        when={true}  
        message={location => {  
          return window.confirm(`confirm to leave to ${location.pathname}?`);  
        }}  
      />  
    </>  
  );  
};
```

js

详见: <https://reacttraining.com/react-router/web/api/Prompt> 

umi/withRouter

≡ UmiJS

Locale

umi-plugin-locale

无需单独引入 `umi-plugin-locale` 依赖，当你使用 `umi-plugin-react` 时，就已经被自动引入了。

setLocale(lang, realReload = true)

指定当前使用语言，`realReload = false` 时，可以无刷新更新多语言设置。

举例:

```
import { setLocale } from 'umi-plugin-locale';

// 十秒后设置当前语言为en-US
setTimeout(() => {
  setLocale('en-US');
}, 10000);
```

js

getLocale()

获取当前使用语言

举例:

```
import { getLocale } from 'umi-plugin-locale';

// 打印当前使用语言
console.log(getLocale());
```

js

使用 [react-intl](#) 提供的组件

你可以直接从 `umi-plugin-locale` 引入由 [react-intl](#) 提供的如下组件：

≡ UmiJS

```
FormattedTime,  
FormattedRelative,  
FormattedNumber,  
FormattedPlural,  
FormattedMessage,  
FormattedHTMLMessage,  
} from 'umi-plugin-locale';  
  
export default () => {  
  return <FormattedMessage id="TEST_TITLE" />;  
};
```

使用 [react-intl](#) 提供的方法

你可以直接从 `umi-plugin-locale` 引入由 [react-intl](#) 提供的如下方法：

```
import {  
  formatDate,  
  formatTime,  
  formatRelative,  
  formatNumber,  
  formatPlural,  
  formatMessage,  
  formatHTMLMessage  
} from 'umi-plugin-locale';  
  
export default () => {  
  return <p>{formatMessage({ id="TEST_TITLE" })}</p>;  
}
```

js

性能

umi/dynamic

动态加载组件，基于 [react-loadable](#) 实现。

dynamic(options)

≡ UmiJS

```
import dynamic from 'umi/dynamic';

// 延时 1s 渲染的组件。
const App = dynamic({
  loader: () => {
    return new Promise((resolve) => {
      setTimeout(() => {
        resolve(() => <div>I will render after 1s</div>);
      }, /* 1s */1000);
    });
  },
});

// 或者用 async 语法
const delay = (timeout) => new Promise(resolve => setTimeout(resolve, timeout));
const App = dynamic({
  loader: async function() {
    await delay(/* 1s */1000);
    return () => <div>I will render after 1s</div>;
  },
});
```

JS

构建

umi/babel

让用户可基于 umi 的 babel 配置进行扩展。

[在 GitHub 上编辑此页](#) 

Last Updated: 2019/6/19 下午2:16:14