

运行时配置

为什么有运行时配置？

我们通过 `.umirc.js` 做编译时的配置，这能覆盖大量场景，但有一些却是编译时很难触及的。

比如：

- 在出错时显示个 message 提示用户
- 在加载和路由切换时显示个 loading
- 页面载入完成时请求后端，根据响应动态修改路由

这些在编译时就很难处理，或者不能处理了。

配置方式

umi 约定 `src` 目录下的 `app.js` 为运行时的配置文件。

```
+ src sh
- app.js      # 运行时配置文件
- package.json
```

配置列表

patchRoutes

用于运行时修改路由。

参数：

- routes: Array , 路由配置

e.g. 添加一个 `/foo` 的路由，

≡ UmiJS

```
    path: '/foo',  
    component: require('./routes/foo').default,  
  });  
}
```

可能的场景：

- 和 `render` 配合使用，请求服务端根据响应动态更新路由
- 修改全部路由，加上某个前缀
- ...

注：

1. 同样适用约定式路由
2. 直接修改 `routes` 就好，不要返回新的路由对象

render

用于改写把整个应用 `render` 到 `dom` 树里的方法。

参数：

- `oldRender`, `Function` , 原始 `render` 方法，需至少被调用一次

e.g. 延迟 1s 渲染应用，

```
export function render(oldRender) {  
  setTimeout(oldRender, 1000);  
}
```

js

可能的场景：

1. 渲染应用之前做权限校验，不通过则跳转到登录页

onRouteChange

用于在初始加载和路由切换时做一些事情。

参数：

≡ UmiJS

- routes: Array , 路由配置
- action: PUSH|POP|REPLACE|undefined , 初次加载时为 undefined

e.g.

```
export function onRouteChange({ location, routes, action }) {  
  bacon(location.pathname);  
}
```

js

可能的场景：

1. 埋点统计

注：

1. 初次加载时也会执行，但 action 为 undefined

rootContainer

用于封装 root container，可以取一部分，或者外面套一层，等等。

参数：

- container, ReactComponent , React 应用最上层的根组件

e.g.

```
export function rootContainer(container) {  
  const DvaContainer = require('@tmp/DvaContainer').default;  
  return React.createElement(DvaContainer, null, container);  
}
```

js

可能的场景：

1. dva、intl 等需要在外层有个 Provider 的场景

modifyRouteProps

修改传给路由组件的 props。

≡ UmiJS

• props, Object , 原始 props

- Object

- route, Object , 当前路由配置

e.g.

```
export function modifyRouteProps(props, { route }) {  
  return { ...props, foo: 'bar' };  
}
```

js

注:

1. 需返回新的 props

在 [GitHub](#) 上编辑此页 

Last Updated: 2019/3/20 上午9:25:02

← 按需加载

区块 →