

## Use umi with dva

自 `>= umi@2` 起, `dva` 的整合可以直接通过 [umi-plugin-react](#) 来配置。

### 特性

- 按目录约定注册 model, 无需手动 `app.model`
- 文件名即 namespace, 可以省去 model 导出的 `namespace` key
- 无需手写 router.js, 交给 umi 处理, 支持 model 和 component 的按需加载
- 内置 query-string 处理, 无需再手动解码和编码
- 内置 dva-loading 和 dva-immmer, 其中 dva-immmer 需通过配置开启
- 开箱即用, 无需安装额外依赖, 比如 dva、dva-loading、dva-immmer、path-to-regexp、object-assign、react、react-dom 等

### # 使用

用 yarn 安装依赖,

```
$ yarn add umi-plugin-react
```

sh

如果你用 npm, 执行 `npm install --save umi-plugin-react`。

然后在 `.umirc.js` 里配置插件:

```
export default {  
  plugins: [  
    [  
      'umi-plugin-react',  
      {  
        dva: true,  
      },  
    ],  
  ],  
};
```

js

## ≡ UmiJS

```
export default {
  plugins: [
    [
      'umi-plugin-react',
      {
        dva: {
          immer: true
        }
      }
    ],
  ],
};
```

JS

## model 注册

model 分两类，一是全局 model，二是页面 model。全局 model 存于 `/src/models/` 目录，所有页面都可引用；**页面 model 不能被其他页面所引用。**

规则如下：

- `src/models/**/*.js` 为 global model
- `src/pages/**/*.models/**/*.js` 为 page model
- global model 全量载入，page model 在 production 时按需载入，在 development 时全量载入
- page model 为 page js 所在路径下 `models/**/*.js` 的文件
- page model 会向上查找，比如 page js 为 `pages/a/b.js`，他的 page model 为 `pages/a/b/models/**/*.js` + `pages/a/models/**/*.js`，依次类推
- 约定 model.js 为单文件 model，解决只有一个 model 时不需要建 models 目录的问题，有 model.js 则不去找 `models/**/*.js`

举个例子，

```
+ src
+ models
- g.js
+ pages
+ a
+ models
- a.js
```

## UmiJS

```
src/
- page.js
+ c
- model.js
+ d
+ models
- d.js
- page.js
- page.js
```

如上目录：

- global model 为 `src/models/g.js`
- `/a` 的 page model 为 `src/pages/a/models/{a,b,ss/s}.js`
- `/c` 的 page model 为 `src/pages/c/model.js`
- `/c/d` 的 page model 为 `src/pages/c/model.js, src/pages/c/d/models/d.js`

## 配置及插件

之前在 `src/dva.js` 下进行配置的方式已 deprecated，下个版本会移除支持。

在 `src` 目录下新建 `app.js`，内容如下：

```
export const dva = {
  config: {
    onError(e) {
      e.preventDefault();
      console.error(e.message);
    },
  },
  plugins: [
    require('dva-logger')(),
  ],
};
```

js

## FAQ

## ≡ UmiJS

### url 变化了，但页面组件不刷新，是什么原因？

layouts/index.js 里如果用了 connect 传数据，需要用 umi/withRouter 高阶一下。

```
import withRouter from 'umi/withRouter';

export default withRouter(connect(mapStateToProps)(LayoutComponent));
```

js

### 如何访问到 store 或 dispatch 方法？

```
window.g_app._store
window.g_app._store.dispatch
```

js

### 如何禁用包括 component 和 models 的按需加载？

在 .umirc.js 里配置：

```
export default {
  plugins: [
    [
      'umi-plugin-react',
      {
        dva: {
          dynamicImport: undefined // 配置在dva里
        },
        dynamicImport: undefined // 或者直接写在react插件的根配置，写在这里也会被继承到
      }
    ],
  ],
};
```

js

### 全局 layout 使用 connect 后路由切换后没有刷新？

需用 withRouter 包一下导出的 react 组件，注意顺序。

# 参考

- [使用 umi 改进 dva 项目开发](#)
- [umi + dva, 完成用户管理的 CURD 应用](#)

在 [GitHub](#) 上编辑此页

Last Updated: 2018/10/30 下午3:58:21

← [Mock 数据](#)

[按需加载](#) →