快速上手

> 更喜欢观看视频？可以点此观看

环境准备

首先得有 node，并确保 node 版本是 8.10 或以上。（mac 下推荐使用 nvm 来管理 node 版本）

```sh
$ node -v
8.1x
```

推荐使用 yarn 管理 npm 依赖，并使用国内源（阿里用户使用内网源）。

```sh
# 国内源
$ npm i yarn tyarn -g
# 后面文档里的 yarn 换成 tyarn
$ tyarn -v

# 阿里内网源
$ tnpm i yarn @ali/yarn -g
# 后面文档里的 yarn 换成 ayarn
$ ayarn -v
```

然后全局安装 umi，并确保版本是 2.0.0 或以上。

```sh
$ yarn global add umi
$ umi -v
2.0.0
```

> FAQ：如果提示 **umi: command not found**，你需要将 `yarn global bin` 路径配置到环境变量中，方法如下：

```
# 在 .bash_profile 中添加下面一行：
export PATH="$PATH:`yarn global bin`"

# windows系统：
# 获取 global bin 的路径
$ yarn global bin
C:\Users\Administrator\AppData\Local\Yarn\bin
# 复制上面的 global bin 的路径，添加到系统环境变量 PATH。
```

脚手架

先找个地方建个空目录。

```sh
$ mkdir myapp && cd myapp
```

然后通过 `umi g` 创建一些页面，

```sh
$ umi g page index
$ umi g page users
```

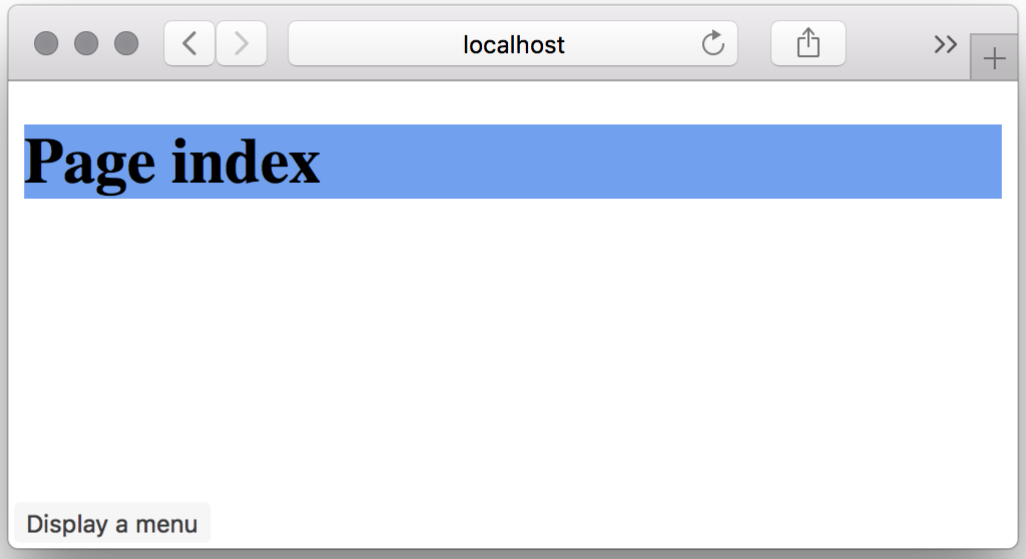
> `umi g` 是 `umi generate` 的别名，可用于快速生成 component、page、layout 等，并且可在插件里被扩展，比如 umi-plugin-dva 里扩展了 dva:model，然后就可以通过 `umi g dva:model foo` 快速 dva 的 model。

然后通过 tree 查看下目录，（windows 用户可跳过此步）

```sh
$ tree
.
└── pages
    ├── index.css
    ├── index.js
    ├── users.css
    └── users.js
```

然后启动本地服务器，

```sh
$ umi dev
```



约定式路由

启动 `umi dev` 后，大家会发现 pages 下多了个 `.umi` 的目录。这是啥？这是 umi 的临时目录，可以在这里做一些验证，但请不要直接在这里修改代码，umi 重启或者 pages 下的文件修改都会重新生成这个文件夹下的文件。

然后我们在 index 和 users 直接加一些路由跳转逻辑。

先修改 `pages/index.js` ，

```
+ import Link from 'umi/link';
import styles from './index.css';

export default function() {
  return (
    <div className={styles.normal}>
      <h1>Page index</h1>
+     <Link to="/users">go to /users</Link>
```

再修改 `pages/users.js` ，

```
+ import router from 'umi/router';
import styles from './index.css';

export default function() {
  return (
    <div className={styles.normal}>
      <h1>Page index</h1>
+     <button onClick={() => { router.goBack(); }}>go back</button>
    </div>
  );
}
```

然后浏览器验证，应该已经可以在 index 和 users 两个页面之间通过路由跳转了。

部署发布

构建

执行 `umi build` ，

```sh
$ umi build

DONE  Compiled successfully in 1729ms

File sizes after gzip:

  68.04 KB  dist/umi.js
  83 B      dist/umi.css
```

构建产物默认生成到 `./dist` 下，然后通过 tree 命令查看，(windows 用户可忽略此步)

```sh
$ tree ./dist
./dist
├── index.html
```

本地验证

发布之前，可以通过 `serve` 做本地验证，

```sh
$ yarn global add serve
$ serve ./dist

Serving!

- Local:            http://localhost:5000
- On Your Network:  http://{Your IP}:5000

Copied local address to clipboard!
```

访问 http://localhost:5000，正常情况下应该是和 `umi dev` 一致的。

部署

本地验证完，就可以部署了，这里通过 now 来做演示。

```sh
$ yarn global add now
$ now ./dist

> Deploying /private/tmp/sorrycc-1KVCmK/dist under chencheng
> Synced 3 files (301.93KB) [2s]
> https://dist-jtckzjjatx.now.sh [in clipboard] [1s]
> Deployment complete!
```

然后打开相应的地址就能访问到线上的地址了。

测试与配置检查

测试

umi 内置了基于 `jest` 的测试工具 umi-test：

```
Options:

    --coverage                    indicates that test coverage information should be
    --collectCoverageFrom=<glob>  a glob pattern relative to matching the files that
    --detectLeaks                 debug memory leaks
```

配置检查

使用 umi inspect 列出配置项的内容用以检查：

```sh
$ umi inspect

Options:

    --mode                specify env mode (development or production, default is de
    --rule <ruleName>     inspect a specific module rule
    --plugin <pluginName> inspect a specific plugin
    --rules               list all module rule names
    --plugins             list all plugin names
    --verbose             show full function definitions in output
```

在 GitHub 上编辑此页

Last Updated: 2019/5/22 下午10:44:20

← 介绍 通过脚手架创建项目 →