React扬帆起航

@vdfor https://github.com/vdfor

2018.5.14

读者要求

- 熟悉ES2015+主要知识点
- 对typescript有一定的了解
- 对react有基本的了解

Hello world

开始

```
npm i create-react-app -g
create-react-app react-demo --scripts-version=react-scripts-ts
cd react-demo
npm start
```

高级配置

- npm run eject
- react-app-rewired

Feature

- 声明式 (Tell it what you want, do not care how to do)
- 组件化
- Learn once, write anywhere
- MVVM(Model-View-ViewModel)

注：声明式与组件化是react的核心思想

组件的生命周期

1. Start
   - getDefaultProps
   - getInitialState
   - componentWillMount
   - render
   - componentDidMount(仅客户端渲染)

2. Changed
   - componentWillReceiveProps(仅props变更)
   - shouldComponentUpdate
   - componentWillUpdate
   - render
   - componentDidUpdate

3. Umount
   - componentWillUnmount

前后端数据交互

- fetch
- axios（推荐）

路由

- 路由属于组件
- 路由级别的代码拆分 (react-loadable)

状态管理(mobx) - 基本

1. 为什么需要引入状态管理工具

2. redux vs mobx
   - 单一store — 多store
   - 函数式编程 — 面向对象的响应式编程
   - immutable — mutable
   - ...

3. with react
   - mobx + mobx-react

4. 核心API(mobx)
   - @observable/@action
   - autorun/when
   - @computed
   - ...

5. 核心API(mobx-react)
   - @observer
   - Provider/inject
   - componentWillReact
   - ...

状态管理(mobx) - 组织store

- 按角色组织 vs 按功能组织(推荐)
- 单例应用
- 定义store → export store → 实例化
- Provider/inject注入

with antd

- css module
- 定制主题
- 按需加载
- 国际化

with echarts

- 按需引入
- refs

拓展与交流

- 复杂应用下states与actions的拆分(mobx)
- css module的缺陷
- 组件级别的代码拆分
- 服务端渲染
- PWA
- 多页应用
- 单元测试
- react性能分析与优化(本身优化与基于mobx的优化)

react-sail

Seed of react with mobx

https://github.com/vdfor/react-sail

注：本文档说述大部分内容在react-sail中有示例