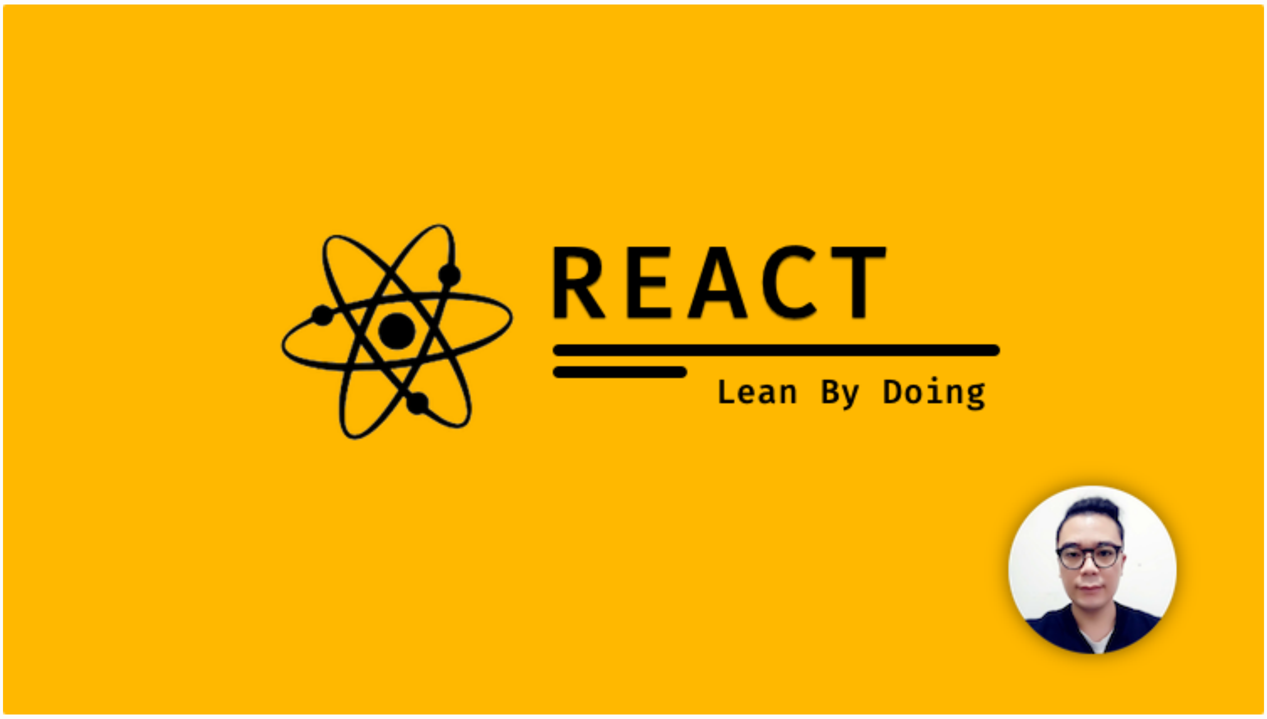
REACT 边做边学

—— 密叔



1、工具和编辑器

- 安装 Node 环境

  https://nodejs.org/en/

- 安装 Vscode 编辑器

  https://code.visualstudio.com/

- 安装 Yarn 包管理工具

  https://yarnpkg.com/en/docs/install#mac-stable

- 安装 React Developer Tools 插件

2、REACT 是什么?

用于构建用户界面的 JavaScript 库

- 是一个 JavaScript 库
    - 用于构建 JavaScript 驱动的 web 应用程序
    - 运行在浏览器端，而非服务器端
        - 数据变化即可响应
        - 不用从服务器端重新获取一个新页面
- 用户界面
    - 一个 React 应用由多个 React 组件组合嵌套构成
    - 每个组件接收外部参数，提供特定功能

3 、创建 REACT 项目

- 使用 create-react-app 创建 react project
    - 使用create-react-app 快速创建 React 项目
        - 全局安装 create-react-app
        - npx 命令
        - npm 命令
        - yarn 命令
    - react 项目目录介绍
    - 清理简化项目
    - React Hello Word 程序

4、REACT 项目结构说明

5、 第一个 REACT 组件

- Login 组件展示介绍
- 清理生成文件—》新建 index.js index.html
- react 项目运行机制
- index 中写 Login 组件编写

- 分割 Login.js
- React Developer Tools 查看

6、使用 JSX 编写 HTML

7、引入 CSS

app.scss

```scss
@charset "utf-8";

// Import a Google Font
@import url('https://fonts.googleapis.com/css?family=Nunito:400,700');
// Set Colors
$family-sans-serif: 'Nunito', sans-serif;
$purple: #c9168d;
$pink: #fa7c91;
$dark: #333;
$beige-lighter: rgb(245, 245, 245);

// update global variables
$primary: $purple;
$link: $dark;
$danger: $pink;

// update component variables
$body-background-color: $beige-lighter;

@import '../../node_modules/bulma/bulma.sass';

.login-wrapper {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  .login-box {
    width: 350px;
    padding: 1.5rem 1.5rem 2rem 1.5rem;
  }
}
```

8、首页面布局

App.js

```
import React from 'react';
import Header from 'components/Header';
import Products from 'components/Products';

class App extends React.Component {
  render() {
    return (
      <div className="main">
        <Header />
        <Products />
      </div>
    );
  }
}

export default App;
```

index.js

```
import React from 'react';
import { render } from 'react-dom';
import App from './components/App';

import './css/app.scss';
import './css/style.scss';

render(<App />, document.querySelector('#root'));
```

Header.js

```
import React from 'react';

class Header extends React.Component {
  render() {
    return (
      <div className="header">
        <div className="grid">
          <div className="start">
            <a href="/">Home</a>
          </div>
          <div className="end">
            <a href="/">LOGIN</a>
            <a href="/">REGISTER</a>
          </div>
        </div>
      </div>
    );
  }
}

export default Header;
```

Products.js

```
import React from 'react';

class Products extends React.Component {
  render() {
    return (
      <div className="products">
        <p>Products Component</p>
      </div>
    );
  }
}

export default Products;
```

jsconfig.json

```
{
  "compilerOptions": {
    "baseUrl": "src"
  },
  "include": ["src"]
}
```

9、PROPS 组件传参

```
<img src="shoe.jpg"  alt="shoe"/>

<input type="text" />

<!--
  props 像公交车或者汽车(两点直接传输)
-->
```

App.js

```
class App extends React.Component {
  render() {
    return (
      <div className="main">
      <Header nickname="Admin" age={28} marry={true} />
        <Products />
      </div>
    );
  }
}
```

Header.js

```
class Header extends React.Component {

  renderLink() {
    const nickname = this.props.nickname;
    if (!nickname) {
      return (
        <React.Fragment>
          <a href="/">LOGIN</a>
          <a href="/">REGISTER</a>
        </React.Fragment>
      );
    } else {
      return (
        <span className="nickname">
          <i className="far fa-user"></i>
          {this.props.nickname}
        </span>
      );
    }
  }

  render() {
    return (
      <div className="header">
        <div className="grid">
          <div className="start">
            <a href="/">Home</a>
          </div>
          <div className="end">{this.renderLink()}</div>
        </div>
      </div>
    );
  }
}
```

10、函数式组件

Header.js

```
const Header = props => (
  <div className="header">
    <div className="grid">
      <div className="start">
        <a href="/">Home</a>
      </div>
      <div className="end">
        {props.nickname ? (
          <span className="nickname">
            <i className="far fa-user"></i>
            {props.nickname}
          </span>
        ) : (
          <React.Fragment>
            <a href="/">Login</a>
            <a href="/">Register</a>
          </React.Fragment>
        )}
      </div>
    </div>
  </div>
);
```

11、组件路由

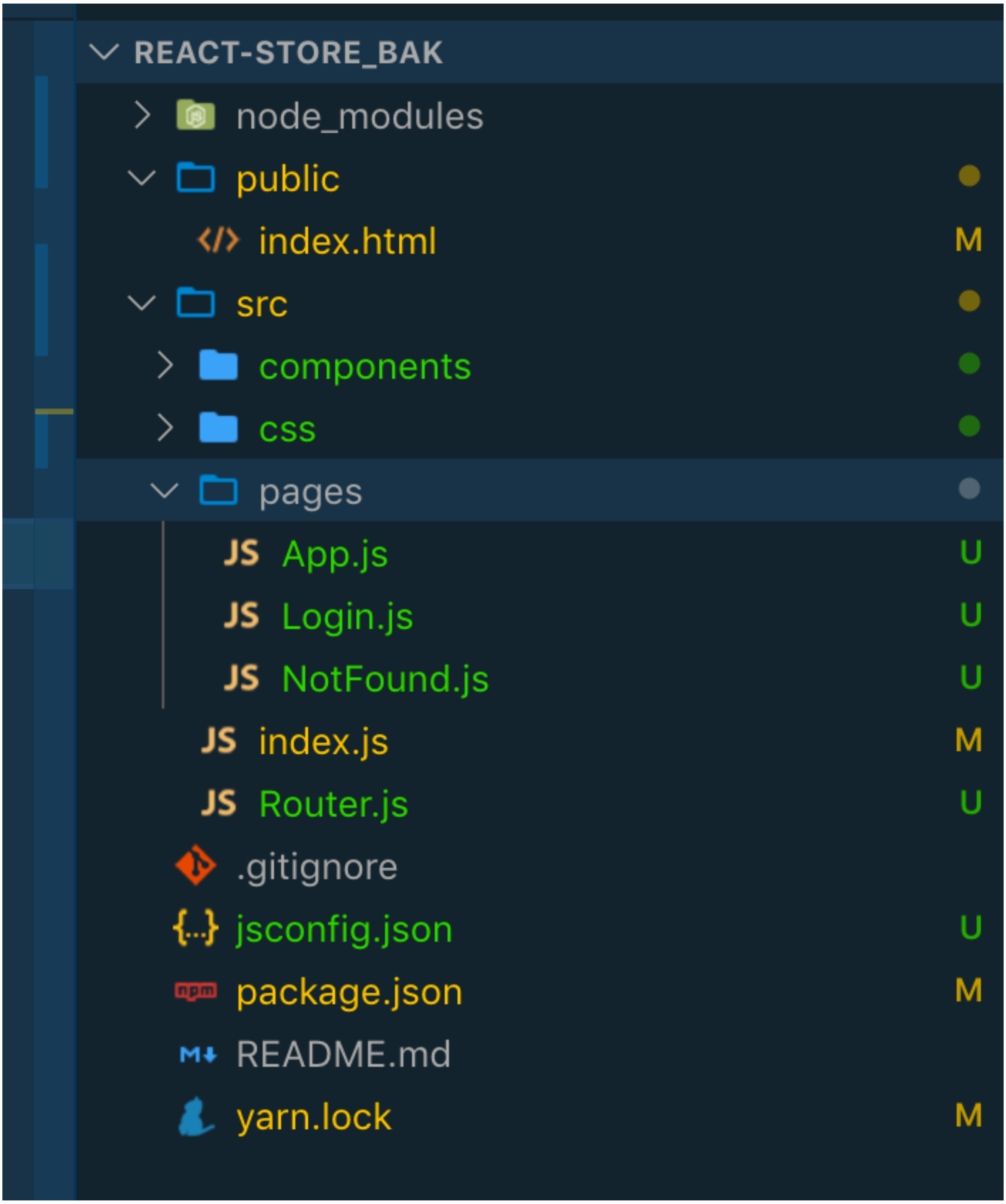
```
yarn add react-router-dom
```

Router.js

```
import React from 'react';
import { BrowserRouter, Switch, Route } from 'react-router-dom';
import App from 'pages/App';
import Login from 'pages/Login';
import NotFound from 'pages/NotFound';

const Router = () => (
  <BrowserRouter>
    <Switch>
      <Route path="/" exact component={App} />
      <Route path="/login" component={Login} />
      <Route component={NotFound} />
    </Switch>
  </BrowserRouter>
);
export default Router;
```

index.js

```
import React from 'react';
import { render } from 'react-dom';
import Router from 'Router';

import 'css/app.scss';
import 'css/style.scss';

render(<Router />, document.querySelector('#root'));
```



```
REACT-STORE_BAK
  > node_modules
  v public                 ●
      index.html           M
  v src                    ●
    > components           ●
    > css                  ●
    v pages                ●
        App.js             U
        Login.js           U
        NotFound.js        U
      index.js             M
      Router.js            U
    .gitignore
    jsconfig.json          U
    package.json           M
    README.md
    yarn.lock              M
```

12、组件中处理事件

Login.js

```
import React from 'react';
class Login extends React.Component {
  /**
    1. 命名和绑定
    2. event
    3. this
    4. 传递参数
   */
  handleClick(msg, event) {
    event.preventDefault();
    console.log(this);
    alert(msg);
  }

  handleSubmit = event => {
    // 1. 阻止表单默认行为
    event.preventDefault();
    // 2. 获取表单数据,

    // 3. 处理登录逻辑

    // 4. 跳转到首页
    this.props.history.push('/');
  };

  render() {
    return (
      <div className="login-wrapper">
        {/* <a href="/login" onClick={event => this.handleClick('clicked',
event)}>
          click me
        </a> */}
        <a href="/login" onClick={this.handleClick.bind(this, 'click')}>
          click me
        </a>

        <form className="box login-box" onSubmit={this.handleSubmit}>
          <div className="field">
            <label className="label">Email</label>
            <div className="control">
              <input className="input" type="text" placeholder="Email" />
            </div>
          </div>
          <div className="field">
            <label className="label">Password</label>
            <div className="control">
              <input className="input" type="password" placeholder="Password"
/>
            </div>
          </div>
          <div className="control">
            <button className="button is-fullwidth is-primary">Login</button>
          </div>
        </form>
      </div>
    );
  }
}
```

13、REFS 访问元素

- Refs 提供了一种方式，允许我们访问 DOM 节点或在 render 方法中创建的 React 元素。
- 当 `ref` 属性用于 HTML 元素时，构造函数中使用 `React.createRef()` 创建的 `ref` 接收底层 DOM 元素作为其 `current` 属性
- 当 `ref` 属性用于自定义 class 组件时， `ref` 对象接收组件的挂载实例作为其 `current` 属性
- **你不能在函数组件上使用 ref 属性**，因为他们没有实例。
- 避免使用 refs 来做任何可以通过声明式实现来完成的事情

14、理解组件 STATE

React 把组件看成是一个状态机（State Machines）。通过与用户的交互，实现不同状态，然后渲染 UI，让用户界面和数据保持一致。

React 里，只需更新组件的 state，然后根据新的 state 重新渲染用户界面（不要操作 DOM）

Login.js

```
import React from 'react';
class Login extends React.Component {

  /*constructor() {
    super();
    this.state = {
      isLike: false,
      count: 0
    };
  }*/

   state = {
      isLike: false,
      count: 0
   }

  handleSubmit = event => {
   ...
  };

  handleClick = () => {
    this.setState({
      isLike: !this.state.isLike
    });

    this.setState({
      count: this.state.count + 1
    });

    this.setState(prevState => {
      return { count: prevState.count + 2 };
    });
  };

  render() {
    return (
      <div className="login-wrapper">
        <form className="box login-box" onSubmit={this.handleSubmit}>
          ....
        </form>
        <div className="control">
          <button
            className="button is-fullwidth is-link"
            onClick={this.handleClick}
          >
            {this.state.isLike ? 'No' : '👍'}
          </button>
        </div>
      </div>
    );
  }
}
```

```
export default Login;
```

15、受控组件

- React 的 State 成为唯一数据源
- 渲染该组件的 React 元素可以控制其 State

Login.js

```
class Login extends React.Component {
  state = {
      email: '',
      password: ''
    }

  handleSubmit = event => {
   ...
  };

  handleChange = e => {
    this.setState({
      [e.target.name]: e.target.value.toUpperCase()
    });
  };

  render() {
    return (
      <div className="login-wrapper">
        <form className="box login-box" onSubmit={this.handleSubmit}>
          <div className="field">
            <label className="label">Email</label>
            <div className="control">
              <input
                className="input"
                type="text"
                placeholder="Email"
                name="email"
                value={this.state.email}
                onChange={this.handleChange}
              />
            </div>
          </div>
          <div className="field">
            <label className="label">Password</label>
            <div className="control">
              <input
                className="input"
                type="password"
                placeholder="Password"
                name="password"
                value={this.state.password}
                onChange={this.handleChange}
              />
            </div>
          </div>
          <div className="control">
            <button className="button is-fullwidth is-primary">Login</button>
          </div>
        </form>
      </div>
    );
  }
}
```

16、商品列表组件布局

Products.js

```
import React from 'react';
import ToolBox from 'components/ToolBox';
import Product from 'components/Product';
class Products extends React.Component {
  render() {
    return (
      <div>
        <ToolBox />
        <div className="products">
          <div className="columns is-multiline is-desktop">
            <div className="column is-3">
              <Product />
            </div>
            <div className="column is-3">
```

```
              <Product />
            </div>
            <div className="column is-3">
              <Product />
            </div>
            <div className="column is-3">
              <Product />
            </div>
            <div className="column is-3">
              <Product />
            </div>
          </div>
        </div>
      </div>
    );
  }
}

export default Products;
```

ToolBox.js

```
import React from 'react';

class ToolBox extends React.Component {
  render() {
    return (
      <div className="tool-box">
        <div className="logo-text">Store</div>
        <div className="search-box">
          <div className="field has-addons">
            <div className="control">
              <input
                type="text"
                className="input search-input"
                placeholder="Search Product"
              />
            </div>
            <div className="control">
              <button className="button">X</button>
            </div>
          </div>
        </div>
        <div className="cart-box">
          <i className="fas fa-shopping-cart"></i>
          <span className="cart-num">(0)</span>
        </div>
      </div>
    );
  }
}

export default ToolBox;
```

Product.js

```
import React from 'react';

class Product extends React.Component {
  render() {
    return (
      <div className="product">
        <p>Product</p>
      </div>
    );
  }
}

export default Product;
```

17、商品组件布局

Product.js

```
import React from 'react';

class Product extends React.Component {
  render() {
    return (
      <div className="product">
        <div className="p-content">
          <div className="img-wrapper">
            <figure className="image is4by3">
              <img src="images/2.jpg" alt="Nike Paul George PG 3" />
            </figure>
            <p className="p-tags">42 Colors</p>
            <p className="p-name">Nike Paul George PG 3</p>
          </div>
        </div>
        <div className="p-footer">
          <p className="price">￥538.00</p>

          <button className="add-cart">
            <i className="fas fa-shopping-cart"></i>
            <i className="fas fa-exclamation "></i>
          </button>
        </div>
      </div>
    );
  }
}

export default Product;
```

18、商品组件数据渲染

React 哲学

- 是先用已有的数据模型渲染一个不包含交互功能的 UI
- 将渲染 UI 和添加交互这两个过程分开

props 和 State

`props` （“properties” 的缩写）和 `state` 都是普通的 JavaScript 对象。它们都是用来保存信息的，这些信息可以控制组件的渲染输出，而它们的一个重要的不同点就是： `props` 是传递给组件的（类似于函数的形参），而 `state` 是在组件内被组件自己管理的（类似于在一个函数内声明的变量）。

- Props 倾向用于组件初始化渲染时，提供数据 — 数据初始化
- State 倾向用于控制组件本身不同状态的切换 — 状态管理

Products.js

```
import React from 'react';
import ToolBox from 'components/ToolBox';
import Product from 'components/Product';
```

```
class Products extends React.Component {
  product = {
    name: 'Air Jordan 4',
    image: 'images/1.jpg',
    tags: '45 Colors',
    price: 59440,
    status: 'available'
  };

  products = [
    {
      id: 1,
      name: 'Air Jordan 4',
      image: '/images/1.jpg',
      tags: '92 ',
      price: 59440,
      status: 'available'
    },
    {
      id: 2,
      name: 'Nike Paul George PG 3',
      image: '/images/2.jpg',
      tags: '25 Colors',
      price: 53800,
      status: 'available'
    },
    {
      id: 3,
      name: 'Jordan Why Not Zer0.2',
      image: '/images/3.jpg',
      tags: '16 Color, y',
      price: 48900,
      status: 'available'
    },
    {
      id: 4,
      name: 'Nike Air Foamposite One',
      image: '/images/4.jpg',
      tags: '84 Colors',
      price: 73148,
      status: 'available'
    },
    {
      id: 5,
      name: 'adidas Harden Vol.3',
      image: '/images/5.jpg',
      tags: '34 Colors',
      price: 46900,
      status: 'unavailable'
    }
  ];
  render() {
    return (
      <div>
        <ToolBox />
        <div className="products">
          <div className="columns is-multiline is-desktop">
            {this.products.map(p => (
              <div className="column is-3" key={p.id}>
                <Product product={p} />
              </div>
            ))}
            {/* <div className="column is-3">
              <Product />
            </div>
            <div className="column is-3">
              <Product />
            </div>
            <div className="column is-3">
              <Product />
            </div>
            <div className="column is-3">
              <Product />
            </div> */}
          </div>
        </div>
      </div>
    );
  }
}

export default Products;
```

Product.js

```
import React from 'react';
import { formatPrice } from 'commons/helper';

class Product extends React.Component {
  // formatPrice(cents) {
  //   return (cents / 100).toLocaleString('zh', {
  //     style: 'currency',
  //     currency: 'CNY'
  //   });
  // }
  // // ￥538.00

  render() {
    const { name, image, tags, price, status } = this.props.product;
    return (
      <div className="product">
        <div className="p-content">
          <div className="img-wrapper">
            {/* <div className="out-stock-text">Out Of Stock</div> */}
            <figure className="image is4by3">
              <img src={image} alt={name} />
            </figure>
          </div>
          <p className="p-tags">{tags}</p>
          <p className="p-name">{name}</p>
        </div>
        <div className="p-footer">
          <p className="price">{formatPrice(price)}</p>

          <button className="add-cart">
            <i className="fas fa-shopping-cart"></i>
            <i className="fas fa-exclamation "></i>
          </button>
        </div>
      </div>
    );
  }
}

export default Product;
```

helper.js

```
export const formatPrice = cents => {
  return (cents / 100).toLocaleString('zh', {
    style: 'currency',
    currency: 'CNY'
  });
};
```

19、商品库存 UI 逻辑处理

Product.js

```
import React from 'react';
import { formatPrice } from 'commons/helper';

class Product extends React.Component {
  render() {
    const { name, image, tags, price, status } = this.props.product;
    const _pClass = {
      unavailable: 'product out-stock',
      available: 'product'
    };
    return (
      <div className={_pClass[status]}>
        <div className="p-content">
          <div className="img-wrapper">
            <div className="out-stock-text">Out Of Stock</div>
            <figure className="image is4by3">
              <img src={image} alt={name} />
            </figure>
          </div>
          <p className="p-tags">{tags}</p>
          <p className="p-name">{name}</p>
        </div>
        <div className="p-footer">
          <p className="price">{formatPrice(price)}</p>

          <button className="add-cart" disabled={status === 'unavailable'}>
            <i className="fas fa-shopping-cart"></i>
            <i className="fas fa-exclamation "></i>
          </button>
        </div>
      </div>
    );
  }
}

export default Product;
```

20、JSON-SERVER 构建 REST API

- json-server 把 json 文件作为一个数据库，快速构建 REST API
- 多用于原型开发提供 API

1、创建 `react-store-api` 文件夹

2、根目录中构建 `db.json` 文件

3、安装 `json-server` 包

4、`package.json` 文件增加启动脚本

```
{
  "dependencies": {
    "json-server": "^0.15.1"
  },
  "scripts": {
    "start": "json-server --watch db.json --port 3003"
  },
  "license": "MIT"
}
```

5、`yarn start` 启动 API

21、请求商品列表 API

Products.js

```
import React from 'react';
import axios from 'commons/axios';
import ToolBox from 'components/ToolBox';
import Product from 'components/Product';
class Products extends React.Component {
  state = {
    products: []
  };

  componentDidMount() {
    // fetch('http://localhost:3003/products')
    //   .then(response => response.json())
    //   .then(data => {
    //     this.setState({
    //       products: data
    //     });
    //   });
    axios.get('/products').then(response => {
      console.log(response.data);
      this.setState({
        products: response.data
      });
    });
  }

  render() {
    return (
      <div>
        <ToolBox />
        <div className="products">
          <div className="columns is-multiline is-desktop">
            {this.state.products.map(p => (
              <div className="column is-3" key={p.id}>
                <Product product={p} />
              </div>
            ))}
          </div>
        </div>
      </div>
    );
  }
}

export default Products;
```

安装 axios

Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中

```
yarn add axios
```

axios.js

```
import _axios from 'axios';

const axios = baseURL => {
  const instance = _axios.create({
    baseURL: baseURL || 'http://localhost:3003',
    timeout: 1000
  });

  // 配置拦截器

  return instance;
};
```

```
export default axios();
```

22、商品列表搜索

ToolBox.js

```
import React from 'react';

class ToolBox extends React.Component {
  state = {
    searchText: ''
  };

  handleChange = e => {
    const value = e.target.value;
    this.setState({
      searchText: value
    });
    this.props.search(value);
  };

  clearSearchText = () => {
    this.setState({
      searchText: ''
    });
    this.props.search('');
  };

  render() {
    return (
      <div className="tool-box">
        <div className="logo-text">Store</div>
        <div className="search-box">
          <div className="field has-addons">
            <div className="control">
              <input
                type="text"
                className="input search-input"
                placeholder="Search Product"
                value={this.state.searchText}
                onChange={this.handleChange}
              />
            </div>
            <div className="control">
```

```
              <button className="button" onClick={this.clearSearchText}>
                X
              </button>
            </div>
          </div>
        </div>
        <div className="cart-box">
          <i className="fas fa-shopping-cart"></i>
          <span className="cart-num">(0)</span>
        </div>
      </div>
    );
  }
}

export default ToolBox;
```

Products.js

```
import React from 'react';
import axios from 'commons/axios';
import ToolBox from 'components/ToolBox';
import Product from 'components/Product';
class Products extends React.Component {
  state = {
    products: [],
    sourceProducts: []
  };

  componentDidMount() {
    axios.get('/products').then(response => {
      this.setState({
        products: response.data,
        sourceProducts: response.data
      });
    });
  }

  // search
  search = text => {
    // 1. Get New Array
    let _products = [...this.state.sourceProducts];

    // 2. Filter New Array
    _products = _products.filter(p => {
      // name : Abcd   text: ab   ==> ['Ab']
      // text : '' ==> ["", "", "", "", ""]
      const matchArray = p.name.match(new RegExp(text, 'gi'));
      return !!matchArray;
    });

    // 3. setState
    this.setState({
      products: _products
    });
  };

  render() {
    return (
      <div>
        <ToolBox search={this.search} />
        <div className="products">
          <div className="columns is-multiline is-desktop">
            {this.state.products.map(p => (
              <div className="column is-3" key={p.id}>
                <Product product={p} />
              </div>
            ))}
          </div>
        </div>
      </div>
    );
  }
}

export default Products;
```

23、商品组件添加动画

React Transition Group 动画组件库

安装

```
yarn add react-transition-group
```

主要组件：

- TransitionGroup
- CSSTransition

Products.js

```
import React from 'react';
import axios from 'commons/axios';
import ToolBox from 'components/ToolBox';
import Product from 'components/Product';
import { TransitionGroup, CSSTransition } from 'react-transition-group';
class Products extends React.Component {
  state = {
    products: [],
    sourceProducts: []
  };

  componentDidMount() {
    axios.get('/products').then(response => {
      this.setState({
        products: response.data,
        sourceProducts: response.data
      });
    });
  }

  // search
  search = text => {
    // 1. Get New Array
    let _products = [...this.state.sourceProducts];

    // 2. Filter New Array
    _products = _products.filter(p => {
      // name : Abcd   text: ab   ==> ['Ab']
      // text : '' ==> ["", "", "", "", ""]
      const matchArray = p.name.match(new RegExp(text, 'gi'));
      return !!matchArray;
    });

    // 3. setState
    this.setState({
      products: _products
    });
  };

  render() {
    return (
      <div>
        <ToolBox search={this.search} />
        <div className="products">
          <div className="columns is-multiline is-desktop">
            <TransitionGroup component={null}>
              {this.state.products.map(p => (
                <CSSTransition
                  classNames="product-fade"
                  timeout={300}
                  key={p.id}
                >
                  <div className="column is-3" key={p.id}>
                    <Product product={p} />
                  </div>
                </CSSTransition>
              ))}
            </TransitionGroup>
          </div>
        </div>
      </div>
    );
  }
}

export default Products;
```

style.scss

```
.product-fade-enter {
  opacity: 0.1;
}
.product-fade-enter-active {
  opacity: 1;
  transition: all 0.3s ease-in;
}

.product-fade-exit {
  opacity: 1;
}

.product-fade-exit-active {
  opacity: 0.1;
  transition: all 0.3s ease-in;
}
```

24、弹出组件(一)

- 一次渲染，随需调用
- 装载组件

Panel.js

```
import React from 'react';
import { render } from 'react-dom';

class Panel extends React.Component {
  state = {
    active: false
  };

  open = () => {
    this.setState({
      active: true
    });
  };

  close = () => {
    this.setState({
      active: false
    });
  };

  render() {
    const _class = {
      true: 'panel-wrapper active',
      false: 'panel-wrapper'
    };
    return (
      <div className={_class[this.state.active]}>
        <div className="over-layer" onClick={this.close}></div>
        <div className="panel">
          <div className="head">
            <span className="close" onClick={this.close}>
              ×
            </span>
            <p className="has-text-centered">Children Component</p>
          </div>
        </div>
      </div>
    );
  }
}

const _div = document.createElement('div');
document.body.appendChild(_div);
const _panel = render(<Panel />, _div);
export default _panel;
```

Products.js

```
import React from 'react';
//...
import Panel from 'components/Panel';
class Products extends React.Component {
  state = {
    products: [],
    sourceProducts: []
  };

  componentDidMount() {
    //...
  }

  // search
  search = text => {
    //...
  };

  // add
  toAdd = () => {
    Panel.open();
  };

  render() {
    return (
      <div>
        <ToolBox search={this.search} />
        <div className="products">
          <div className="columns is-multiline is-desktop">
            ...
          </div>

          {/* add */}
          <button className="button is-primary add-btn" onClick={this.toAdd}>
            add
          </button>
        </div>
      </div>
    );
  }
}

export default Products;
```

25、弹出组件(二)

- 装载组件
  - 子组件作为参数传递并被渲染
  - 子组件可以关闭弹出层
  - 子组件与调用者可以通讯

Products.js

```
import React from 'react';
...
class Products extends React.Component {
 ...
  // add
  toAdd = () => {
    Panel.open({
      component: AddInventory,
      callback: data => {
        console.log('callback data: ', data);
      }
    });
  };

  render() {
    return (
      <div>
        ...
      </div>
    );
  }
}

export default Products;
```

Panel.js

```
import React from 'react';
import { render } from 'react-dom';

/**
 * 1. 一次渲染，随需调用
 * 2. 装载组件
    (1)、子组件作为参数传递并被渲染
    (2)、子组件可以关闭弹出层
    (3)、子组件与调用者可以通讯
 */

class Panel extends React.Component {
  state = {
    active: false,
    component: null,
    callback: () => {}
  };

  open = options => {
    const { component, callback } = options;
    const _key = new Date().getTime();
    const _component = React.createElement(component, {
      close: this.close,
      key: _key
    });
    this.setState({
      active: true,
      component: _component,
      callback: callback
    });
  };

  close = data => {
    this.setState({
      active: false
    });
    this.state.callback(data);
  };

  render() {
    const _class = {
      true: 'panel-wrapper active',
      false: 'panel-wrapper'
    };
    return (
      <div className={_class[this.state.active]}>
        <div className="over-layer" onClick={this.close}></div>
        <div className="panel">
          <div className="head">
            <span className="close" onClick={this.close}>
              ×
            </span>
            {this.state.component}
          </div>
        </div>
      </div>
    );
  }
}
```

```
const _div = document.createElement('div');
document.body.appendChild(_div);
const _panel = render(<Panel />, _div);
export default _panel;
```

AddInventory.js

```
import React from 'react';

class AddInventory extends React.Component {
  render() {
    return (
      <div className="inventory">
        <p className="title has-text-centered">Inventory</p>
        <br />
        <div className="control">
          <input type="text" className="input" />
        </div>
        <br />
        <div className="control">
          <button
            className="button"
            onClick={() => {
              this.props.close('AddInventory data');
            }}
          >
            Cancel
          </button>
        </div>
      </div>
    );
  }
}

export default AddInventory;
```

26、商品库存新增

AddInventory.js

```
import React from 'react';
import axios from 'commons/axios';

class AddInventory extends React.Component {
  state = {
    name: '',
    price: 0,
    tags: '',
    image: '',
    status: 'available'
  };

  handleChange = e => {
    const name = e.target.name;
    let value = e.target.value;
    if (name === 'price') {
      value = parseInt(value);
    }
    this.setState({
      [name]: value
    });
  };

  submit = e => {
    e.preventDefault();
    const product = {
      ...this.state
    };
    console.log(product);
    axios.post('/products', product).then(res => {
      this.props.close(res.data);
      alert('Add Success');
    });
  };

  render() {
    return (
      <div className="inventory">
        <p className="title has-text-centered">Inventory</p>
        <form onSubmit={this.submit}>
          <div className="field">
            <div className="control">
              <label className="label">Name</label>
              <textarea
                className="textarea"
                name="name"
                onChange={this.handleChange}
              />
            </div>
          </div>
          <div className="field">
            <div className="control">
              <label className="label">price</label>
              <input
                type="number"
                className="input"
                name="price"
                onChange={this.handleChange}
              />
            </div>
          </div>
          <div className="field">
            <div className="control">
              <label className="label">Tags</label>
              <input
                type="text"
                className="input"
                name="tags"
                onChange={this.handleChange}
              />
            </div>
          </div>
          <div className="field">
            <div className="control">
              <label className="label">Image</label>
              <input
                type="text"
                className="input"
                name="image"
                onChange={this.handleChange}
              />
            </div>
          </div>
          <div className="field">
            <div className="control">
              <label className="label">Status</label>
              <div className="select is-fullwidth">
                <select name="status" onChange={this.handleChange}>
                  <option>available</option>
                  <option>unavailable</option>
                </select>
              </div>
            </div>
          </div>
          <br />
          <div className="field is-grouped is-grouped-centered">
            <div className="control">
              <button className="button is-link">Submit</button>
            </div>
            <div className="control">
```

```
              <button className="button" type="button" onClick={() =>
this.props.close()}>Cancel</button>
            </div>
          </div>
        </form>
      </div>
    );
  }
}

export default AddInventory;
```

Products.js

```
import React from 'react';
...
class Products extends React.Component {
 ...
  // add
  toAdd = () => {
    Panel.open({
      component: AddInventory,
      callback: data => {
        if (data) {
          this.add(data);
        }
      }
    });
  };


 add = product => {
    const _products = [...this.state.products];
    _products.push(product);
    const _sProducts = [...this.state.sourceProducts];
    _sProducts.push(product);
    this.setState({
      products: _products,
      sourceProducts: _sProducts
    });
  };

  render() {
    return (
      <div>
        ...
      </div>
    );
  }
}

export default Products;
```

27、使用通知组件

通知组件：react-toastify

1、安装

```
$ npm install --save react-toastify
$ yarn add react-toastify
```

2、引入配置

index.js

```
...
import { ToastContainer } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';
...

render(
  <div>
    <ToastContainer
      position="top-right"
      autoClose={5000}
      hideProgressBar
      newestOnTop={false}
      closeOnClick
      rtl={false}
      pauseOnVisibilityChange
      draggable
      pauseOnHover
    />
    <Router />
  </div>,
  document.querySelector('#root')
);
```

3、使用

AddInventory.js

```
//...
import { toast } from 'react-toastify';

class AddInventory extends React.Component {
  state = {
    //...
  };

  handleChange = e => {
   //...
  };

  submit = e => {
    e.preventDefault();
    const product = {
      ...this.state
    };
    console.log(product);
    axios.post('/products', product).then(res => {
      this.props.close(res.data);
      toast.success('Add Success');
    });
  };

  showToast = () => {
    toast('default');
    toast.info('info');
    toast.success('success');
    toast.warning('warning');
    toast.error('error');
  };

  render() {
    return (
      <div className="inventory">
          {/*...*/}
          <button
             className="button is-primary"
             type="button"
             onClick={this.showToast}
             >
             Show
          </button>
      </div>
    );
  }
}

export default AddInventory;
```

28、商品库存修改

EditInventory.js

```
import React from 'react';
import axios from 'commons/axios';
import { toast } from 'react-toastify';

class EditInventory extends React.Component {
  state = {
    id: '',
    name: '',
    price: '',
    tags: '',
    image: '',
    status: 'available'
    // ...this.props.product
  };

  componentDidMount() {
    const { id, name, image, tags, price, status } = this.props.product;
    this.setState({
      id,
      name,
      image,
      tags,
      price,
      status
    });
  }

  handleChange = e => {
   //...
  };

  submit = e => {
    e.preventDefault();
    const product = {
      ...this.state
    };
    axios.put(`/products/${this.state.id}`, product).then(res => {
      this.props.close(res.data);
      toast.success('Edit Success');
    });
  };

  render() {
    return (
      <div className="inventory">
        {/*...*/}
      </div>
    );
  }
}

export default EditInventory;
```

Product.js

```
import React from 'react';
import Panel from 'components/Panel';
import { formatPrice } from 'commons/helper';
import EditInventory from 'components/EditInventory';

class Product extends React.Component {
  toEdit = () => {
    Panel.open({
      component: EditInventory,
      props: { product: this.props.product },
      callback: data => {
        if (data) {
          this.props.update(data);
        }
      }
    });
  };

  render() {
    const { name, image, tags, price, status } = this.props.product;
    const _pClass = {
      unavailable: 'product out-stock',
      available: 'product'
    };
    return (
      <div className={_pClass[status]}>
        <div className="p-content">
          <div className="p-head has-text-right" onClick={this.toEdit}>
            <span className="icon edit-btn">
              <i className="fas fa-sliders-h"></i>
            </span>
          </div>
           {/*...*/}
      </div>
    );
  }
}

export default Product;
```

Panel.js

```
import React from 'react';
import { render } from 'react-dom';



class Panel extends React.Component {
  //...

  open = (
    options = {
      props: {}
    }
  ) => {
    const { props, component, callback } = options;
    const _key = new Date().getTime();
    const _component = React.createElement(component, {
      ...props,
      close: this.close,
      key: _key
    });
    this.setState({
      active: true,
      component: _component,
      callback: callback
    });
  };

 //...
}

const _div = document.createElement('div');
```

```
document.body.appendChild(_div);
const _panel = render(<Panel />, _div);
export default _panel;
```

Products.js

```
import React from 'react';
//...
class Products extends React.Component {
  state = {
    products: [],
    sourceProducts: []
  };

  componentDidMount() {
   //...
  }

  // search
  search = text => {
    //...
  };

  // add
  toAdd = () => {
   //...
  };

  add = product => {
    //...
  };

  update = product => {
    const _products = [...this.state.products];
    const _index = _products.findIndex(p => p.id === product.id);
    _products.splice(_index, 1, product);
    const _sProducts = [...this.state.sourceProducts];
    const _sIndex = _sProducts.findIndex(p => p.id === product.id);
    _sProducts.splice(_sIndex, 1, product);
    this.setState({
      products: _products,
      sourceProducts: _sProducts
    });
  };

  render() {
    return (
      <div>
```

```
        <ToolBox search={this.search} />
        <div className="products">
           {/* ... */}

          <Product product={p} update={this.update} />

          {/* ... */}

        </div>
      </div>
    );
  }
}

export default Products;
```

29、商品库存删除

EditInventory.js

```
import React from 'react';
import axios from 'commons/axios';
import { toast } from 'react-toastify';

class EditInventory extends React.Component {
  state = {
    id: '',
    name: '',
    price: '',
    tags: '',
    image: '',
    status: 'available'
    // ...this.props.product
  };

  componentDidMount() {
    const { id, name, image, tags, price, status } = this.props.product;
    this.setState({
      id,
      name,
      image,
      tags,
      price,
      status
    });
  }

  handleChange = e => {
   //...
  };

  submit = e => {
   //...
  };
  onDelete = () => {
    axios.delete(`/products/${this.state.id}`).then(res => {
      this.props.deleteProduct(this.state.id);
      this.props.close();
      toast.success('Delete Success');
    });
  };

  render() {
    return (
      <div className="inventory">
        <p className="title has-text-centered">Inventory</p>
        <form onSubmit={this.submit}>
          ...

          <div className="field is-grouped is-grouped-centered">
            ...
            <div className="control">
              <button
                className="button is-danger"
                type="button"
                onClick={this.onDelete}
              >
                Delete
              </button>
            </div>
            ...
          </div>
        </form>
      </div>
    );
  }
}

export default EditInventory;
```

Products.js

```
import React from 'react';
//...
class Products extends React.Component {
  state = {
    products: [],
    sourceProducts: []
  };

  componentDidMount() {
   //...
  }

  // search
  search = text => {
    //...
  };

  // add
  toAdd = () => {
   //...
  };

  add = product => {
    //...
  };

  update = product => {
    //...
  };

  delete = id => {
    const _products = this.state.products.filter(p => p.id !== id);
    const _sProducts = this.state.sourceProducts.filter(p => p.id !== id);
    this.setState({
      products: _products,
      sourceProducts: _sProducts
    });
  };

  render() {
    return (
      <div>
        <ToolBox search={this.search} />
        <div className="products">
           {/* ... */}

          <Product product={p} update={this.update} delete={this.delete} />

          {/* ... */}

        </div>
      </div>
    );
  }
}

export default Products;
```

Product.js

```
import React from 'react';
import Panel from 'components/Panel';
import { formatPrice } from 'commons/helper';
import EditInventory from 'components/EditInventory';

class Product extends React.Component {
  toEdit = () => {
    Panel.open({
      component: EditInventory,
      props: { product: this.props.product, deleteProduct: this.props.delete
},
      callback: data => {
        if (data) {
          this.props.update(data);
        }
      }
    });
  };

  render() {
    const { name, image, tags, price, status } = this.props.product;
    const _pClass = {
      unavailable: 'product out-stock',
      available: 'product'
    };
    return (
      <div className={_pClass[status]}>
        <div className="p-content">
          <div className="p-head has-text-right" onClick={this.toEdit}>
            <span className="icon edit-btn">
              <i className="fas fa-sliders-h"></i>
            </span>
          </div>
           {/*...*/}
      </div>
    );
  }
}
```

```
export default Product;
```

30、添加购物车(一)

Product.js

```
import React from 'react';
//...

class Product extends React.Component {
  toEdit = () => {
    //...
  };

  addCart = async () => {
    try {
      const { id, name, image, price } = this.props.product;
      const res = await axios.get('/carts', {
        params: {
          productId: id
        }
      });
      const carts = res.data;
      if (carts && carts.length > 0) {
        const cart = carts[0];
        cart.mount += 1;
        await axios.put(`/carts/${cart.id}`, cart);
      } else {
        const cart = {
          productId: id,
          name,
          image,
          price,
          mount: 1
        };
        await axios.post('/carts', cart);
      }
      toast.success('Add Cart Success');
      //this.props.updateCartNum();
    } catch (error) {
      toast.error('Add Cart Failed');
    }
  };

  render() {
    const { name, image, tags, price, status } = this.props.product;
    const _pClass = {
      unavailable: 'product out-stock',
      available: 'product'
    };
    return (
      <div className={_pClass[status]}>
        <div className="p-content">
          {/*...*/}
        <div className="p-footer">

          {/*...*/}
          <button
            className="add-cart"
            disabled={status === 'unavailable'}
            onClick={this.addCart}
          >
            <i className="fas fa-shopping-cart"></i>
            <i className="fas fa-exclamation "></i>
          </button>
        </div>
      </div>
    );
  }
}

export default Product;
```

31、添加购物车(二)

ToolBox.js

```
import React from 'react';

class ToolBox extends React.Component {
  state = {
    searchText: ''
  };

  handleChange = e => {
   //...
  };
  clearSearchText = () => {
    //...
  };

  render() {
    return (
      <div className="tool-box">
       {/*...*/}
        <div className="cart-box">
          <i className="fas fa-shopping-cart"></i>
          <span className="cart-num">({this.props.cartNum})</span>
        </div>
      </div>
    );
  }
}

export default ToolBox;
```

Products.js

```
import React from 'react';
import axios from 'commons/axios';
import { TransitionGroup, CSSTransition } from 'react-transition-group';
import ToolBox from 'components/ToolBox';
import Product from 'components/Product';
import Panel from 'components/Panel';
import AddInventory from 'components/AddInventory';
class Products extends React.Component {
  state = {
    products: [],
    sourceProducts: [],
    cartNum: 0
  };

  componentDidMount() {
    // ...
    this.updateCartNum();
  }

  // search
  search = text => {
   // ...
  };

  // add
  toAdd = () => {
    // ...
  };

  add = product => {
    // ...
  };

  update = product => {
    // ...
  };

  delete = id => {
    // ...
  };

  updateCartNum = async () => {
    const num = await this.initCartNum();
    this.setState({
      cartNum: num
    });
  };

  initCartNum = async () => {
    const res = await axios.get(`/carts`);
    const carts = res.data || [];
    const cartNum = carts
      .map(c => c.mount)
      .reduce((accumulator, value) => accumulator + value, 0);
    return cartNum || 0;
  };

  render() {
    return (
      <div>
        {/* ... */}
        <ToolBox search={this.search} cartNum={this.state.cartNum} />

         <Product
            product={p}
            update={this.update}
            delete={this.delete}
            updateCartNum={this.updateCartNum}
            />
          {/* ... */}
      </div>
    );
  }
}

export default Products;
```

Product.js

```
import React from 'react';
//...

class Product extends React.Component {
  toEdit = () => {
    //...
  };

  addCart = async () => {
    try {
      const { id, name, image, price } = this.props.product;
      const res = await axios.get('/carts', {
        params: {
          productId: id
        }
      });
      const carts = res.data;
      if (carts && carts.length > 0) {
        const cart = carts[0];
        cart.mount += 1;
        await axios.put(`/carts/${cart.id}`, cart);
      } else {
        const cart = {
          productId: id,
          name,
          image,
          price,
          mount: 1
        };
        await axios.post('/carts', cart);
      }
      toast.success('Add Cart Success');
      this.props.updateCartNum();
    } catch (error) {
      toast.error('Add Cart Failed');
    }
  };

  render() {
   //...
  }
}

export default Product;
```

32、购物车路由和布局

- withRouter

> 默认情况下必须经过路由匹配渲染的组件才存在this.props,才拥有路由参数
>
> 使用withRouter就可以给此未经过路由匹配渲染的组件传入路由参数

- props.children

ToolBox.js

```
import React from 'react';
import { withRouter } from 'react-router-dom';

class ToolBox extends React.Component {
  state = {
    searchText: ''
  };

 // ...

  goCart = () => {
    this.props.history.push('/cart');
  };

  render() {
    return (
      <div className="tool-box">
        {/* ... */}
        <div className="cart-box" onClick={this.goCart}>
          <i className="fas fa-shopping-cart"></i>
          <span className="cart-num">({this.props.cartNum})</span>
        </div>
      </div>
    );
  }
}

export default withRouter(ToolBox);
```

Cart.js

```
import React from 'react';
import Layout from 'Layout';

const Cart = props => {
  return (
    <Layout>
      <div className="cart-page">
        <p className="title has-text-centered">Cart Page</p>
      </div>
    </Layout>
  );
};

export default Cart;
```

Layout.js

```
import React from 'react';
import Header from 'components/Header';
const Layout = props => (
  <div className="main">
    <Header />
    {/* child component */}
    {props.children}
  </div>
);

export default Layout;
```

App.js

```
import React from 'react';
import Products from 'components/Products';
import Layout from 'Layout';

class App extends React.Component {
  render() {
    return (
      <Layout>
        <Products />
      </Layout>
    );
  }
}

export default App;
```

33、购物车列表(一)

Cart.js

```
import React from 'react';
import Layout from 'Layout';
import CartItem from 'components/CartItem';

const Cart = () => (
  <Layout>
    <div className="cart-page">
      <span className="cart-title">Shopping Cart</span>
      <div className="cart-list">
        <CartItem />
        <CartItem />
        <CartItem />
      </div>
      <div className="cart-total">
        Total:
        <span className="total-price">￥2345</span>
      </div>
    </div>
  </Layout>
);

export default Cart;
```

CartItem.js

```
import React from 'react';
```

```
const CartItem = () => {
  return (
    <div className="columns is-vcentered">
      <div className="column is-narrow">
        <span className="close">X</span>
      </div>
      <div className="column is-narrow">
        <img src="images/1.jpg" alt="" width="100" />
      </div>
      <div className="column cart-name is-narrow">Nike Paul George PG 3</div>
      <div className="column">
        <span className="price">￥538.00</span>
      </div>
      <div className="column">
        <input type="number" className="input num-input" />
      </div>
      <div className="column">
        <span className="sum-price">￥538.00</span>
      </div>
    </div>
  );
};

export default CartItem;
```

34、购物车列表(二)

Cart.js

```
import React, { useState, useEffect } from 'react';
import axios from 'commons/axios';
import Layout from 'Layout';
import CartItem from 'components/CartItem';
import { formatPrice } from 'commons/helper';

const Cart = () => {
  const [carts, setCarts] = useState([]);

  useEffect(() => {
    axios.get('/carts').then(res => setCarts(res.data));
  });

  const totalPrice = () => {
    const totalPrice = carts
      .map(cart => cart.mount * parseInt(cart.price))
      .reduce((a, value) => a + value, 0);
    return formatPrice(totalPrice);
  };

  return (
    <Layout>
      <div className="cart-page">
        <span className="cart-title">Shopping Cart</span>
        <div className="cart-list">
          {carts.map(cart => (
            <CartItem key={cart.id} cart={cart} />
          ))}
        </div>
        <div className="cart-total">
          Total:
          <span className="total-price">{totalPrice()}</span>
        </div>
      </div>
    </Layout>
  );
};

export default Cart;
```

CartItem.js

```
import React from 'react';
import { formatPrice } from 'commons/helper';

const CartItem = props => {
  const { name, image, price, mount } = props.cart || {};
  const sumPrice = formatPrice(mount * parseInt(price));
  return (
    <div className="columns is-vcentered">
      <div className="column is-narrow">
        <span className="close">X</span>
      </div>
      <div className="column is-narrow">
        <img src={image} alt={name} width="100" />
      </div>
      <div className="column cart-name is-narrow">{name}</div>
      <div className="column">
        <span className="price">{formatPrice(price)}</span>
      </div>
      <div className="column">
        <input type="number" className="input num-input" defaultValue={mount}
/>
      </div>
      <div className="column">
        <span className="sum-price">{sumPrice}</span>
      </div>
    </div>
  );
};

export default CartItem;
```

35、购物车列表(三)

Cart.js

```
import React, { useState, useEffect } from 'react';
import axios from 'commons/axios';
import Layout from 'Layout';
import CartItem from 'components/CartItem';
import { formatPrice } from 'commons/helper';

// Hook
const Cart = props => {
  const [carts, setCarts] = useState([]);

  useEffect(() => {
    axios.get('/carts').then(res => {
      setCarts(res.data);
    });
  }, []);

  const totalPrice = () => {
    const totalCent = carts
      .map(cart => cart.mount * parseInt(cart.price))
      .reduce((a, value) => a + value, 0);
    return formatPrice(totalCent);
  };

  const updateCart = cart => {
    const newCarts = [...carts];
    const _index = newCarts.findIndex(c => c.id === cart.id);
    newCarts.splice(_index, 1, cart);
    setCarts(newCarts);
  };
  const deleteCart = cart => {
    const _carts = carts.filter(c => c.id !== cart.id);
    setCarts(_carts);
  };

  return (
    <Layout>
      <div className="cart-page">
        <span className="cart-title">Shopping Cart</span>
        <div className="cart-list">
          {carts.map(cart => (
            <CartItem
              key={cart.id}
              cart={cart}
              updateCart={updateCart}
              deleteCart={deleteCart}
            />
          ))}
        </div>
        <div className="cart-total">
          Total:
          <span className="total-price">{totalPrice()}</span>
        </div>
      </div>
    </Layout>
  );
};

export default Cart;
```

CartItem.js

```
import React, { useState } from 'react';
import axios from 'commons/axios';
import { formatPrice } from 'commons/helper';

const CartItem = props => {
  const [mount, setMount] = useState(props.cart.mount);
  const { id, name, image, price } = props.cart || {};

  const handleChange = e => {
    const _mount = parseInt(e.target.value);
    setMount(_mount);
    const newCart = {
      ...props.cart,
      mount: _mount
    };
    axios.put(`/carts/${id}`, newCart).then(res => {
      props.updateCart(newCart);
    });
  };

  const sumPrice = () => {
    return formatPrice(mount * parseInt(price));
  };

  const deleteCart = () => {
    axios.delete(`/carts/${id}`).then(res => {
      props.deleteCart(props.cart);
    });
  };

  return (
    <div className="columns is-vcentered">
      <div className="column is-narrow" onClick={deleteCart}>
        <span className="close">X</span>
      </div>
      <div className="column is-narrow">
        <img src={image} alt={name} width="100" />
      </div>
      <div className="column is-narrow cart-name">{name}</div>
      <div className="column ">
        <span className="price"></span>
      </div>
      <div className="column">
        <input
          type="number"
          className="input num-input"
          min={1}
          value={mount}
          onChange={handleChange}
        />
      </div>
      <div className="column">
        <span className="sum-price">{sumPrice()}</span>
      </div>
    </div>
  );
};

export default CartItem;
```

36、购物车列表(四)

Cart.js

```
import React, { useState, useEffect, useMemo } from 'react';
import { TransitionGroup, CSSTransition } from 'react-transition-group';
import axios from 'commons/axios';
import Layout from 'Layout';
import CartItem from 'components/CartItem';
import { formatPrice } from 'commons/helper';

// Hook
const Cart = props => {
  const [carts, setCarts] = useState([]);

  useEffect(() => {
    axios.get('/carts').then(res => {
      setCarts(res.data);
    });
  }, []);

  const totalPrice = useMemo(() => {
    const totalCent = carts
      .map(cart => cart.mount * parseInt(cart.price))
      .reduce((a, value) => a + value, 0);
    return formatPrice(totalCent);
  }, [carts]);

  const updateCart = cart => {
    const newCarts = [...carts];
    const _index = newCarts.findIndex(c => c.id === cart.id);
    newCarts.splice(_index, 1, cart);
    setCarts(newCarts);
  };

  const deleteCart = cart => {
    const _carts = carts.filter(c => c.id !== cart.id);
    setCarts(_carts);
  };

  return (
    <Layout>
      <div className="cart-page">
        <span className="cart-title">Shopping Cart</span>
        <div className="cart-list">
          <TransitionGroup component={null}>
            {carts.map(cart => (
              <CSSTransition classNames="cart-item" timeout={500} key=
{cart.id}>
                <CartItem
                  key={cart.id}
                  cart={cart}
                  updateCart={updateCart}
                  deleteCart={deleteCart}
                />
              </CSSTransition>
            ))}
          </TransitionGroup>
        </div>
        {carts.length === 0 ? <p className="no-cart">NO GOODS</p> : ''}

        <div className="cart-total">
          Total:
          <span className="total-price">{totalPrice}</span>
        </div>
      </div>
    </Layout>
  );
};

export default Cart;
```

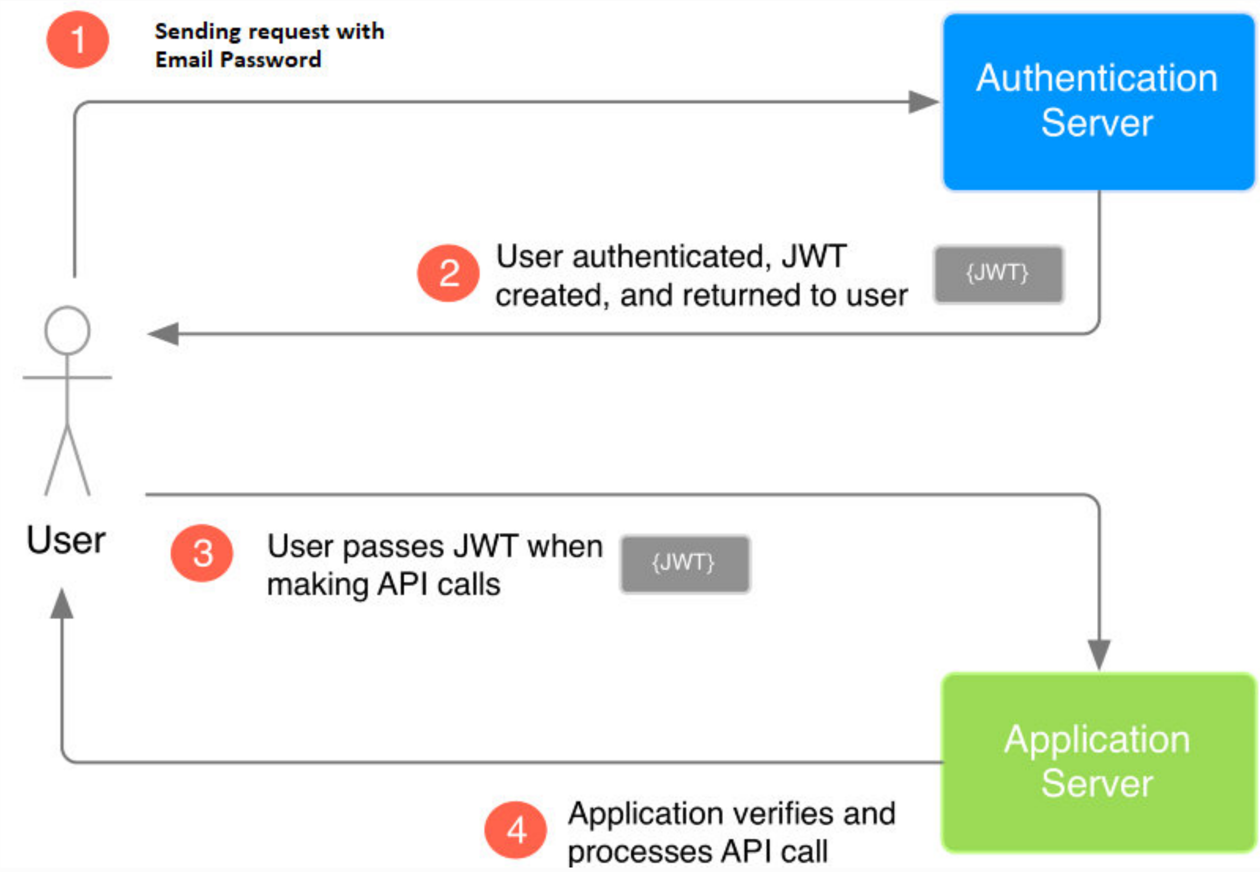
CartItem.js

```
import React, { useState, useMemo } from 'react';
import axios from 'commons/axios';
import { formatPrice } from 'commons/helper';

const CartItem = props => {
  const [mount, setMount] = useState(props.cart.mount);
  const { id, name, image, price } = props.cart || {};

  const handleChange = e => {
    const _mount = parseInt(e.target.value);
    setMount(_mount);
    const newCart = {
      ...props.cart,
      mount: _mount
    };
    axios.put(`/carts/${id}`, newCart).then(res => {
      props.updateCart(newCart);
    });
  };

  const sumPrice = useMemo(() => {
    return formatPrice(mount * parseInt(price));
  }, [mount, price]);

  const deleteCart = () => {
    axios.delete(`/carts/${id}`).then(res => {
      props.deleteCart(props.cart);
    });
  };

  return (
    <div className="columns is-vcentered">
      <div className="column is-narrow" onClick={deleteCart}>
        <span className="close">X</span>
      </div>
      <div className="column is-narrow">
        <img src={image} alt={name} width="100" />
      </div>
      <div className="column is-narrow cart-name">{name}</div>
      <div className="column ">
        <span className="price"></span>
      </div>
      <div className="column">
        <input
          type="number"
          className="input num-input"
          min={1}
          value={mount}
          onChange={handleChange}
        />
      </div>
      <div className="column">
        <span className="sum-price">{sumPrice}</span>
      </div>
    </div>
  );
};

export default CartItem;
```

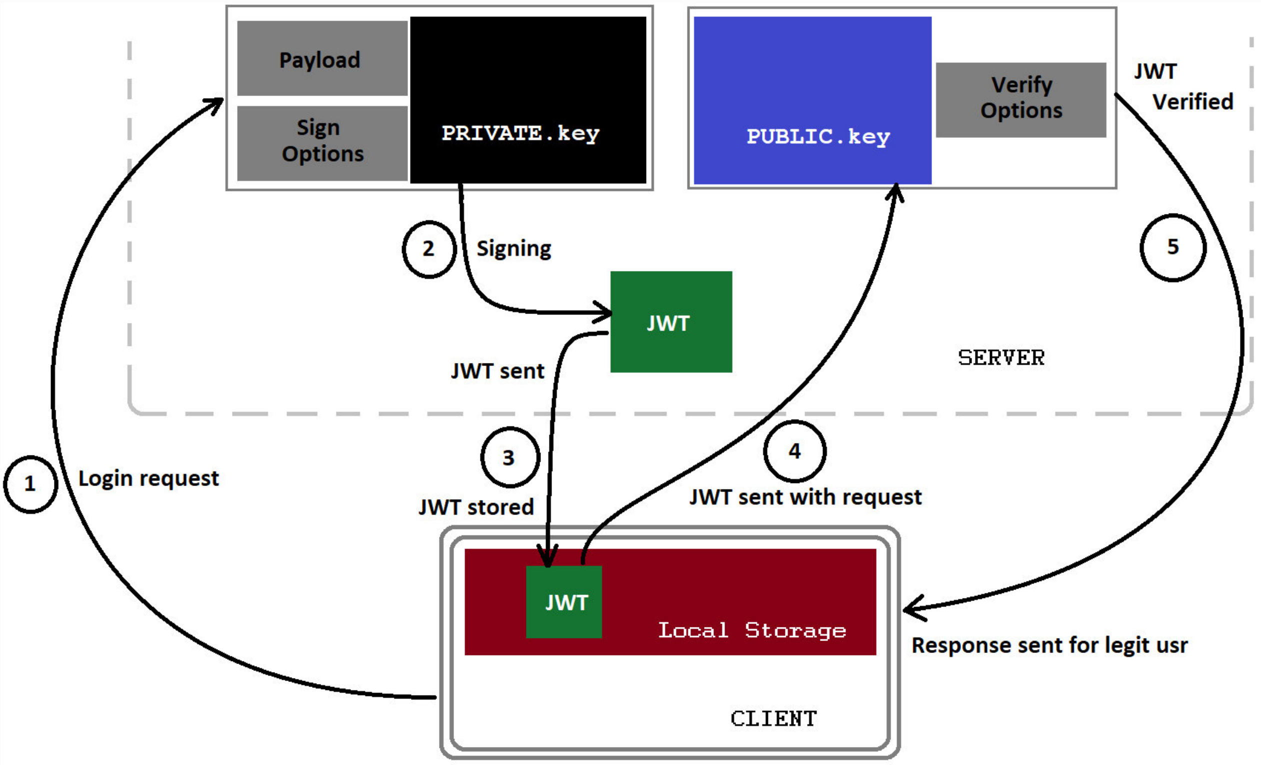
37、JWT 认证

JSON Web Token 入门教程

请求步骤：


1
Sending request with
Email Password
Authentication
Server
2
User authenticated, JWT
created, and returned to user
{JWT}
User
3
User passes JWT when
making API calls
{JWT}
Application
Server
4
Application verifies and
processes API call

认证步骤：



node-jsonwebtoken

JSON SERVER

38、JSON SERVER 添加认证

index.js

```
const fs = require('fs');
const path = require('path');
const jsonServer = require('json-server');
const jwt = require('jsonwebtoken');

// json server module
const server = jsonServer.create();
const router = jsonServer.router('db.json');
const middleWares = jsonServer.defaults();
server.use(jsonServer.bodyParser);
server.use(middleWares);

// get user data from json
const getUsersDb = () => {
  return JSON.parse(
    fs.readFileSync(path.join(__dirname, 'users.json'), 'UTF-8')
  );
};

const isAuthenticated = ({ email, password }) => {
  return (
    getUsersDb().users.findIndex(
      user => user.email === email && user.password === password
    ) !== -1
  );
};

// create json web token
const SECRET = '121SDKFJKSDF8908SDF';
const expiresIn = '1h';
const createToken = payload => {
  return jwt.sign(payload, SECRET, { expiresIn });
};

// login
server.post('/auth/login', (req, res) => {
  console.log(req.body);
  const { email, password } = req.body;
  console.log({ email, password });
  if (!isAuthenticated({ email, password })) {
    const status = 401;
    const message = 'Incorrect email or password';
    return res.status(401).json({ status, message });
  } else {
    const user = getUsersDb().users.find(
      u => u.email === email && u.password === password
    );
    const { nickname, type } = user;
    const jwToken = createToken({ nickname, type, email });
    return res.status(200).json(jwToken);
  }
});

// register

server.use(router);
server.listen(3003, () => {
  console.log('JSON Server is running');
});
```

users.json

```
{
  "users": [
    {
      "id": 1,
      "email": "admin@163.com",
      "password": "admin",
      "type": 1
    }
  ]
}
```

package.json

```
{
  "dependencies": {
    "json-server": "^0.15.1",
    "jsonwebtoken": "^8.5.1"
  },
  "scripts": {
    "start": "nodemon index.js",
    "server": "json-server --watch db.json --port 3003"
  },
  "license": "MIT",
  "devDependencies": {
    "nodemon": "^1.19.4"
  }
}
```

39、登录(一) 表单验证

React Hook Form

Login.js

```
import React from 'react';
import useForm from 'react-hook-form';

export default function Login(props) {
  const { register, handleSubmit, errors } = useForm();

  const onSubmit = data => {
    console.log(data);

    // 3. 处理登录逻辑
    // axios.post('/auth/login', { ...state }).then(res => {
    //   console.log('login response data :', res.data);
    // });
    // 4. 跳转到首页
    // props.history.push('/');
  };
  console.log(errors.password);

  return (
    <div className="login-wrapper">
      <form className="box login-box" onSubmit={handleSubmit(onSubmit)}>
        <div className="field">
          <label className="label">Email</label>
          <div className="control">
            <input
              className={`input ${errors.email && 'is-danger'}`}
              type="text"
              placeholder="Email"
              name="email"
              ref={register({
                required: 'password is required',
                pattern: {
                  value: /^[A-Za-z0-9]+([_\\.][A-Za-z0-9]+)*@([A-Za-z0-9\\-
]+\.)+[A-Za-z]{2,6}$/,
                  message: 'invalid email'
                }
              })}
            />
            {errors.email && (
              <p className="helper has-text-danger">{errors.email.message}
</p>
            )}
          </div>
        </div>
        <div className="field">
          <label className="label">Password</label>
          <div className="control">
            <input
              className={`input ${errors.password && 'is-danger'}`}
              type="password"
              placeholder="Password"
              name="password"
              ref={register({
                required: 'password is required',
                minLength: {
                  value: 6,
                  message: 'cannot be less than 6 digits'
                }
              })}
            />
            {errors.password && (
              <p className="helper has-text-danger">
                {errors.password.message}
              </p>
            )}
          </div>
        </div>
        <div className="control">
          <button className="button is-fullwidth is-primary">Login</button>
        </div>
      </form>
    </div>
  );
}
```

40、登录(二) JWT

Login.js

```
import React from 'react';
import useForm from 'react-hook-form';
import axios from 'commons/axios';
import { toast } from 'react-toastify';

export default function Login(props) {
  const { register, handleSubmit, errors } = useForm();

  const onSubmit = async data => {
    // 1. 处理登录逻辑
    try {
      const { email, password } = data;
      const res = await axios.post('/auth/login', { email, password });
      const jwToken = res.data;
      global.auth.setToken(jwToken);
      toast.success('Login Success');
      // 2. 跳转到首页
      props.history.push('/');
    } catch (error) {
      const { message } = error.response.data;
      toast.error(message);
    }
  };

  return (
    <div>
      {/* ... */}
    </div>
  );
}
```

Layout.js

```
import React, { useMemo } from 'react';
import Header from 'components/Header';
const Layout = props => {
  const user = useMemo(() => {
    const user = global.auth.getUser() || {};
    return user;
  }, []);

  return (
    <div className="main">
      <Header user={user} />
      {/* child component */}
      {props.children}
    </div>
  );
};

export default Layout;
```

auth.js

```
import decode from 'jwt-decode';

const JWT = 'store_token_id';

const setToken = token => {
  localStorage.setItem(JWT, token);
};

const getToken = () => {
  return localStorage.getItem(JWT);
};

const isLogin = () => {
  const jwToken = getToken();
  return !!jwToken && !isTokenExpired(jwToken);
};

const isTokenExpired = token => {
  try {
    const _info = decode(token);
    if (_info.exp < Date.now() / 1000) {
      return true;
    } else return false;
  } catch (error) {
    return false;
  }
};

const getUser = () => {
  const jwToken = getToken();
  if (isLogin()) {
    const user = decode(jwToken);
    return user;
  } else {
    return null;
  }
};

global.auth = {
  isLogin,
  setToken,
  getToken,
  getUser
};
```

Header.js

```
import React from 'react';
import { Link } from 'react-router-dom';

const Header = props => (
  <div className="header">
    <div className="grid">
      <div className="start">
        <Link to="/">Home</Link>
      </div>
      <div className="end">
        {props.user.nickname ? (
          <span className="nickname">
            <i className="far fa-user"></i>
            {props.user.nickname}
          </span>
        ) : (
          <>
            <Link to="/login">Login</Link>
            <Link to="/register">Register</Link>
          </>
        )}
      </div>
    </div>
  </div>
);

export default Header;
```

41、登出和用户信息

UserProfile.js

```
import React from 'react';
export default function UserProfile(props) {
  const logout = () => {
    global.auth.logout();
    props.close('logout');
  };

  return (
    <div className="user-profile">
      <p className="title has-text-centered">Profile</p>
      <fieldset disabled>
        <div className="field">
          <div className="control">
            <label className="label">Nickname</label>
            <input
              className="input"
              type="text"
              defaultValue={props.user.nickname}
            />
          </div>
        </div>

        <div className="field">
          <div className="control">
            <label className="label">Email</label>
            <input
              className="input"
              type="text"
              defaultValue={props.user.email}
            />
          </div>
        </div>

        <div className="field">
          <div className="control">
            <label className="label">Type</label>
            <input
              className="input"
              type="text"
              defaultValue={props.user.type === 1 ? 'Manager' : 'General
User'}
            />
          </div>
        </div>
      </fieldset>

      <br />
      <br />
      <div className="field is-grouped is-grouped-centered">
        <div className="control">
          <button className="button is-danger" onClick={logout}>
            Logout
          </button>
        </div>
        <div className="control">
          <button
            className="button"
            type="button"
            onClick={() => {
              props.close();
            }}
          >
            Cancel
          </button>
        </div>
      </div>
    </div>
  );
}
```

Header.js

```
import React from 'react';
import { Link, withRouter } from 'react-router-dom';
import Panel from 'components/Panel';
import UserProfile from 'components/UserProfile';

const Header = props => {
  const showProfile = () => {
    Panel.open({
      component: UserProfile,
      props: {
        user: props.user
      },
      callback: data => {
        console.log(data);
        if (data === 'logout') {
          props.history.go(0);
        }
      }
    });
  };

  return (
    <div className="header">
      {/* .... */}
    </div>
  );
};
```

```
export default withRouter(Header);
```

auth.js

```
import decode from 'jwt-decode';

//...

const logout = () => {
  localStorage.removeItem(JWT);
};

global.auth = {
  //...
  logout
};
```

42、用户注册

server.js

```

// ....

// Register New User
server.post('/auth/register', (req, res) => {
  const { email, password, nickname, type } = req.body;

  // ----- 1 step
  if (isAuthenticated({ email, password })) {
    const status = 401;
    const message = 'Email and Password already exist';
    return res.status(status).json({ status, message });
  }

  // ----- 2 step
  fs.readFile(path.join(__dirname, 'users.json'), (err, _data) => {
    if (err) {
      const status = 401;
      const message = err;
      return res.status(status).json({ status, message });
    }
    // Get current users data
    const data = JSON.parse(_data.toString());
    // Get the id of last user
    const last_item_id = data.users[data.users.length - 1].id;
    //Add new user
    data.users.push({ id: last_item_id + 1, email, password, nickname, type
}); //add some data
    fs.writeFile(
      path.join(__dirname, 'users.json'),
      JSON.stringify(data),
      (err, result) => {
        // WRITE
        if (err) {
          const status = 401;
          const message = err;
          res.status(status).json({ status, message });
          return;
        }
      }
    );
  });

  // Create token for new user
  const jwToken = createToken({ nickname, type, email });
  res.status(200).json(jwToken);
});

// ...
```

Register.js

```
import React from 'react';
import useForm from 'react-hook-form';
import axios from 'commons/axios';
import { toast } from 'react-toastify';

export default function Login(props) {
  const { register, handleSubmit, errors } = useForm();

  const onSubmit = async data => {
    // 3. 处理注册逻辑
    try {
      const { nickname, email, password } = data;
      const res = await axios.post('/auth/register', {
        email,
        password,
        nickname,
        type: 0
      });
      const jwToken = res.data;
      global.auth.setToken(jwToken);
      toast.success('Register Success');
      // 4. 跳转到首页
      props.history.push('/');
    } catch (error) {
      const { message } = error.response.data;
      toast.error(message);
    }
  };

  return (
    <div className="login-wrapper">
      <form className="box login-box" onSubmit={handleSubmit(onSubmit)}>
        <div className="field">
          <label className="label">Nickname</label>
          <div className="control">
            <input
              className={`input ${errors.nickname && 'is-danger'}`}
              type="text"
              placeholder="Nickname"
              name="nickname"
              ref={register({
                required: 'nickname is required'
              })}
            />
            {errors.nickname && (
              <p className="helper has-text-danger">
                {errors.nickname.message}
              </p>
            )}
          </div>
        </div>
        <div className="field">
          <label className="label">Email</label>
          <div className="control">
            <input
              className={`input ${errors.email && 'is-danger'}`}
              type="text"
              placeholder="Email"
              name="email"
              ref={register({
                required: 'email is required',
                pattern: {
                  value: /^[A-Za-z0-9]+([_\\.][A-Za-z0-9]+)*@([A-Za-z0-9\\-
]+\.)+[A-Za-z]{2,6}$/,
                  message: 'invalid email'
                }
              })}
            />
            {errors.email && (
              <p className="helper has-text-danger">{errors.email.message}
</p>
            )}
          </div>
        </div>
        <div className="field">
          <label className="label">Password</label>
          <div className="control">
            <input
              className={`input ${errors.password && 'is-danger'}`}
              type="password"
              placeholder="Password"
              name="password"
              ref={register({
                required: 'password is required',
                minLength: {
                  value: 6,
                  message: 'cannot be less than 6 digits'
                }
              })}
            />
            {errors.password && (
              <p className="helper has-text-danger">
                {errors.password.message}
              </p>
            )}
          </div>
        </div>
        <div className="control">
          <button className="button is-fullwidth is-primary">Submit</button>
        </div>
      </form>
    </div>
  );
}
```

43、视图权限控制

ToolBox.js

```
import React from 'react';
import { withRouter } from 'react-router-dom';
import { toast } from 'react-toastify';

class ToolBox extends React.Component {
 //....


  goCart = () => {
    if (!global.auth.isLogin()) {
      this.props.history.push('/login');
      toast.info('Please Login First');
      return;
    }
    this.props.history.push('/cart');
  };

  render() {
    return (
      <div className="tool-box">
        {/*  ..... */}
      </div>
    );
  }
}

export default withRouter(ToolBox);
```

Product.js

```
import React from 'react';
import { withRouter } from 'react-router-dom';
import { toast } from 'react-toastify';
import Panel from 'components/Panel';
import axios from 'commons/axios';
import { formatPrice } from 'commons/helper';
import EditInventory from 'components/EditInventory';

class Product extends React.Component {
  toEdit = () => {
    //...
  };

  addCart = async () => {
    if (!global.auth.isLogin()) {
      this.props.history.push('/login');
      toast.info('Please Login First');
      return;
    }
    // ...
  };

  renderManagerBtn = () => {
    const user = global.auth.getUser() || {};
    if (user.type === 1) {
      return (
        <div className="p-head has-text-right" onClick={this.toEdit}>
          <span className="icon edit-btn">
            <i className="fas fa-sliders-h"></i>
          </span>
        </div>
      );
    }
  };

  render() {
    const { name, image, tags, price, status } = this.props.product;
    const _pClass = {
      unavailable: 'product out-stock',
      available: 'product'
    };
    return (
      <div className={_pClass[status]}>
        <div className="p-content">
          {this.renderManagerBtn()}
         {/* ... */}
        </div>
         {/* ... */}
      </div>
    );
  }
}

export default withRouter(Product);
```

Products.js

```
import React from 'react';
//...

class Products extends React.Component {
  // ...

  render() {
    return (
      <div>
        {/* ... */}
          {(global.auth.getUser() || {}).type === 1 && (
            <button className="button is-primary add-btn" onClick=
{this.toAdd}>
              add
            </button>
          )}
        </div>
      </div>
    );
  }
}

export default Products;
```

44、购物车 API 访问控制

server.js

```
//...

/**
request headers --> Authorization
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
.eyJuaWNrbmFtZSI6ImFkbWluIiwidHlwZSI6MSwiZW1haWwiOiJhZG1pbkAxNjMuY29tIiwiaWF0
IjoxNTcyNzU3MjAzLCJleHAiOjE1NzI3NjA4MDN9
.f4hfN1IjU4E23Lo44N-2VLzc1qoyNu1oZg2iQreZTfU
*/

// server.use(/^(?!\/auth).*$/, (req, res, next) => {
// server.use(['/carts'], (req, res, next) => {
server.use('/carts', (req, res, next) => {
  if (
    req.headers.authorization === undefined ||
    req.headers.authorization.split(' ')[0] !== 'Bearer'
  ) {
    const status = 401;
    const message = 'Error in authorization format';
    res.status(status).json({ status, message });
    return;
  }
  try {
    const verifyTokenResult = verifyToken(
      req.headers.authorization.split(' ')[1]
    );
    if (verifyTokenResult instanceof Error) {
      const status = 401;
      const message = 'Access token not provided';
      res.status(status).json({ status, message });
      return;
    }
    next();
  } catch (err) {
    const status = 401;
    const message = 'Error token is revoked';
    res.status(status).json({ status, message });
  }
});
// Verify the token
const verifyToken = token => {
  return jwt.verify(token, SECRET, (err, decode) =>
    decode !== undefined ? decode : err
  );
};

//...
```

axios.js

```
import _axios from 'axios';

const axios = baseURL => {
  const instance = _axios.create({
    baseURL: baseURL || 'http://localhost:3003',
    timeout: 1000
  });

  // 配置拦截器
  instance.interceptors.request.use(
    config => {
      // Do something before request is sent
      const jwToken = global.auth.getToken();
      config.headers['Authorization'] = 'Bearer ' + jwToken;
      return config;
    },
    error => {
      // Do something with request error
      return Promise.reject(error);
    }
  );

  return instance;
};
export { axios };
export default axios();
```

45、购物车数据权限

Cart.js

```
import React, { useState, useEffect, useMemo } from 'react';
import { TransitionGroup, CSSTransition } from 'react-transition-group';
import axios from 'commons/axios';
import Layout from 'Layout';
import CartItem from 'components/CartItem';
import { formatPrice } from 'commons/helper';

// Hook
const Cart = props => {
  const [carts, setCarts] = useState([]);

  useEffect(() => {
    const user = global.auth.getUser() || {};
    axios.get(`/carts?userId=${user.email}`).then(res => {
      setCarts(res.data);
    });
  }, []);

 // ....

  return (
    <Layout>
      {/* ... */}
    </Layout>
  );
};

export default Cart;
```

Products.js

```
import React from 'react';
// ...
class Products extends React.Component {

  // ...

  initCartNum = async () => {
    const user = global.auth.getUser() || {};
    const res = await axios.get(`/carts`, {
      params: {
        userId: user.email
      }
    });
    const carts = res.data || [];
    const cartNum = carts
      .map(c => c.mount)
      .reduce((accumulator, value) => accumulator + value, 0);
    return cartNum || 0;
  };

  render() {
    return (
      <div>
        {/* ... */}
      </div>
    );
  }
}

export default Products;
```

Product.js

```
import React from 'react';
// ...

class Product extends React.Component {
  // ...

  addCart = async () => {
    if (!global.auth.isLogin()) {
      this.props.history.push('/login');
      toast.info('Please Login First');
      return;
    }
    try {
      const user = global.auth.getUser() || {};
      const { id, name, image, price } = this.props.product;
      const res = await axios.get('/carts', {
        params: {
          productId: id
        }
      });
      const carts = res.data;
      if (carts && carts.length > 0) {
        const cart = carts[0];
        cart.mount += 1;
        await axios.put(`/carts/${cart.id}`, cart);
      } else {
        const cart = {
          productId: id,
          name,
          image,
          price,
          mount: 1,
          userId: user.email
        };
        await axios.post('/carts', cart);
      }
      toast.success('Add Cart Success');
      this.props.updateCartNum();
    } catch (error) {
      toast.error('Add Cart Failed');
    }
  };

  //...

  render() {
   // ...
  }
}

export default withRouter(Product);
```

46、项目部署

（一） react-store 部署到 now

1. 安装 now

```
sudo npm i -g now
# 或者
yarn global add now
```

```
# 查看版本
now -v
```

2. react-store 安装 serve

```
npm install serve --save
# 或者
yarn add serve
```

3. 启动命令修改

package.json

```
...
"scripts": {
    "start": "serve --single ./build",
    "dev": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
...
```

4. 命令部署

```
cd react-store
now
```

（二）react-store-api 部署到 now

1. 项目根目录下创建 now.json

```
{
  "version": 2,
  "name": "react-store-api",
  "builds": [{ "src": "/server.js", "use": "@now/node-server" }],
  "routes": [
    {
      "src": "/(.*)",
      "dest": "/server.js"
    }
  ]
}
```

2. 部署到 now

```
cd react-store-api
now
```

（三）react-store 切换 API 域名

1. react-store 根目录增加环境变量配置文件 `.env.development` 和 `.env.production`

   .env.development

   ```
   REACT_APP_API_DOMAIN = http://localhost:3003
   ```

   .env.production

   ```
   REACT_APP_API_DOMAIN = https://react-store-api.lirenmi.now.sh
   ```

2. axios.js 修改api 访问域名

```
import _axios from 'axios';

const axios = baseURL => {
  const instance = _axios.create({
    baseURL:
      baseURL || process.env.REACT_APP_API_DOMAIN || 'http://localhost:3003',
    timeout: 1000
  });

  // 配置拦截器
  instance.interceptors.request.use(
   // ...
  );

  return instance;
};
export { axios };
export default axios();
```

3. 重新部署 react-store 项目

```
cd react-store
now
```

其他选择：

- netlify
- coding.net
- heroku
- 阿里云
- 其他