

React项目



React项目

课堂目标

资源

项目效果

开发步骤

step01 - 迁出种子项目

step02 - 创建布局页面

step03 - 用户登录认证

step04 - 商品列表

step05 - 添加购物车

回顾

课堂目标

1. 掌握umi项目开发流程
2. Ant-design-pro应用

资源

1. [umi](#)
2. [dva](#)
3. [antd-pro](#)

项目效果

```
git clone https://github.com/57code/umi-test.git
```

开发步骤

step01 - 迁出种子项目

```
git clone -b step01 https://github.com/57code/umi-test.git
```

step02 - 创建布局页面

- [antd布局组件使用](#)
- layouts/index.js

```
import { Layout } from "antd";
import styles from "../index.css";

const { Header, Footer, Content } = Layout;

export default function(props) {
  return (
    <Layout>
      { /* 页头 */ }
      <Header className={styles.header}>导航</Header>
      { /* 内容 */ }
      <Content className={styles.content}>
        <div className={styles.box}>{props.children}</div>
      </Content>
      { /* 页脚 */ }
      <Footer className={styles.footer}>开课吧</Footer>
    </Layout>
  );
}
```

- 样式设置, index.css

```
.header {
  color: white;
}
.content {
  margin: 16px;
}
.box {
  padding: 24px;
  background: #fff;
  min-height: 500px;
}
.footer {
  text-align: center;
}
```

- 配置路由, config.js

```
{
  path: "/",
  component: "../layouts",
  routes: [
    // 移动之前路由配置到这里
  ]
}
```

- 顶部导航菜单

```

<Menu theme="dark" mode="horizontal"
  defaultSelectedKeys={['2']}
  style={{ lineHeight: "64px" }}>
  <Menu.Item key="1"><Link to="/">商品</Link></Menu.Item>
  <Menu.Item key="2"><Link to="/users">用户</Link></Menu.Item>
  <Menu.Item key="3"><Link to="/about">关于</Link></Menu.Item>
</Menu>

```

step03 - 用户登录认证

- 引入ant-design-pro, 安装: `npm install ant-design-pro --save`
- 修改404页面提示内容, 404.js

```

// umi的配置, 已经自动支持antd-pro的按需加载
import {Exception} from 'ant-design-pro'
export default function() {
  return (
    <Exception type="404" backText="返回首页"></Exception>
  );
}

```

- 登录页构建, login.js:

```

import React, { Component } from "react";
import styles from "./login.css";
import router from "umi/router";
import { Login } from "ant-design-pro";

const { UserName, Password, Submit } = Login; // 通用的用户名、密码和提交组件

// 改为类形式组件, 可持有状态
export default class extends Component {
  // let from = props.location.state.from || "/"; // 重定向地址
  onSubmit = (err, values) => {
    console.log(err, values);
  };
  render() {
    return (
      <div className={styles.loginForm}>
        </* logo */>
        
        </* 登录表单 */>
        <Login onSubmit={this.onSubmit}>
          <UserName
            name="username"
            placeholder="kaikeba"
            rules={[{ required: true, message: "请输入用户名" }]}
          />

```

```

        <Password
          name="password"
          placeholder="123"
          rules={[{ required: true, message: "请输入密码" }]}
        />
        <Submit>登录</Submit>
      </Login>
    </div>
  );
}
}

```

样式, login.css

```

.loginForm {
  width: 50%;
  margin: 100px auto;
  text-align: center;
}
.logo {
  margin-bottom: 30px;
}

```

- 接口mock, 创建./mock/login.js
- 用户信息保存和登录动作编写, 创建./src/models/user.js
- 请求登录, login.js
- 失败处理

step04 - 商品列表

- 数据mock, ./mock/goods.js
- 图片素材, ./public/courses/*.png

```

// 从远程分支step04-seed迁出新分支step04
git pull origin step04-seed:step04

```

- 修改商品数据模型, src\pages\goods\models\goods.js
- 显示课程分类页签, src\pages\goods\index.js

```

import { TagSelect } from "ant-design-pro";

@connect(
  state => ({
    courses: state.goods.courses, // 映射课程数据
    tags: state.goods.tags, // 映射标签数据
  }),

```

```

    {...}
  )
  class Goods extends Component {
    // 页签变更
    tagSelectChange = (tags) => {
      console.log(tags);
    };
    render() {
      if (this.props.loading.models.goods) {
        return <div>加载中...</div>;
      }
      return (
        <div>
          { /* 分类标签 */ }
          <TagSelect onChange={this.tagSelectChange}>
            {this.props.tags.map(tag => {
              return (
                <TagSelect.Option key={tag} value={tag}>
                  {tag}
                </TagSelect.Option>
              );
            })}
          </TagSelect>
        </div>
      );
    }
  }
  export default Goods;

```

- 显示课程列表, src\pages\goods\index.js

```

{ /* 商品列表 */ }
<Row type="flex" justify="start">
  {this.state.displayCourses.map((item, index) => {
    return (
      <Col key={index} style={{ padding: 10 }} span={6}>
        {item.name ? (
          <Card
            hoverable
            title={item.name}
            cover={<img src={"/course/" + item.img} />}
          >
            <Card.Meta
              description={
                <div>
                  <span>¥{item.price}</span>
                  <span style={{ float: "right" }}>
                    <Icon type="user" /> {item.solded}
                  </span>
                </div>
              }
            </Card.Meta>
          </>
        ) : </>}
      </div>
    );
  })}
</div>

```

```

        </Card>
      ) : (
        <Skeleton active={true} />
      )}
    </Col>
  );
})}
</Row>

```

step05 - 添加购物车

- 创建购物车模型, ./src/models/cart.js

```

export default {
  namespace: "cart", // 可省略
  state: JSON.parse(localStorage.getItem("cart")) || [], // 初始状态: 缓存或空数组
  reducers: {
    addCart(cart, action) {
      const good = action.payload;
      const idx = cart.findIndex(v => v.id === good.id);
      if (idx > -1) {
        // 更新数量
        const cartCopy = [...cart];
        const itemCopy = { ...cartCopy[idx] };
        itemCopy.count += 1;
        cartCopy.splice(idx, 1, itemCopy);
        return cartCopy;
      } else {
        // 新增
        return [...cart, { ...good, count: 1 }];
      }
    }
  }
};

```

- 请求添加购物车, src/pages/goods/index.js
- 购物车数据同步到localStorage, 新建./src/app.js, 配置dva

```

export const dva = {
  config: {
    onStateChange(state, a) {
      if (localStorage) {
        localStorage.setItem("cart", JSON.stringify(state.cart));
      }
    }
  }
};

```

- 页头添加购物车信息, ./src/layouts/index.js

```

export default class extends Component {
  render(){
    ...
    <div style={{float:'right'}}>
      <Icon type="shopping-cart" style={{fontSize:18}}/>
      <span>我的购物车</span>
      <Badge count={5} offset={[-4,-18]}/>
    </div>
  }
}

```

~ 徽章尺寸调整，添加全局样式，./src/global.css

```

.ant-badge-count{
  height: 16px;
  border-radius: 8px;
  min-width: 16px;
  line-height: 16px;
  padding: 0;
}

```

~ 显示购物车数据

```

@connect(state => ({ // 连接购物车状态
  count: state.cart.length,
  cart: state.cart
}))
export default class extends Component {
  render(){
    // 构造购物车列表菜单
    const menu = (
      <Menu>
        {this.props.cart.map((item, index) => (
          <Menu.Item key={index}>
            {item.name}×{item.count} <span>¥{item.count * item.price}</span>
          </Menu.Item>
        ))}
      </Menu>
    );
    ...
    return (
      { /* 购物车信息放在Dropdown以便展示 */ }
      <Dropdown overlay={menu} placement="bottomRight">
        <div className={styles.cart}>
          { /* 购物车项目数量 */ }
          <Badge count={this.props.count} offset={[-4, -18]} />
        </div>
      </Dropdown>
    )
  }
}

```

回顾

React项目

课堂目标

资源

项目效果

开发步骤

step01 - 迁出种子项目

step02 - 创建布局页面

step03 - 用户登录认证

step04 - 商品列表

step05 - 添加购物车

回顾

