

前端开发

前端注册功能实现

在service/user.js中增加reg注册函数

```
import axios from 'axios';
import store from 'store'
import {observable} from 'mobx';

// 过期插件
store.addPlugin(require('store/plugins/expire'));

export default class UserService {
  @observable loggedin = false; //+ 被观察者

  login (email, password) {
    console.log(email, password);

    axios.post('/api/user/login', {
      email:email,
      password:password
    })/* dev server会代理 */
    .then(
      response => { // 此函数要注意this的问题
        console.log(response.data);
        console.log(response.status);
        //+ 存储token
        store.set('token',
          response.data.token,
          (new Date()).getTime() + (8*3600*1000));
        this.loggedin = true;
      }
    ).catch(
      function (error) {
        console.log(error);
      }
    )
  }

  reg (name, email, password) {
    console.log(email, password);

    axios.post('/api/user/reg', {
      email:email,
      password:password,
      name:name
    })/* dev server会代理 */
    .then(
```

```

        response => { // 此函数要注意this的问题
            console.log(response.data);
            console.log(response.status);
            //+ 存储token
            store.set('token',
                response.data.token,
                (new Date()).getTime() + (8*3600*1000));
            // 注册成功, 返回token即成功登录
            this.loggedin = true; //+ 修改被观察者
        }
    ).catch(
        function (error) {
            console.log(error);
        }
    )
}
}

```

Reg组件类

```

import React from 'react';
import { Link, Redirect } from 'react-router-dom';
import './css/login.css'
import UserService from '../service/user';
import { observer } from 'mobx-react';

const userService = new UserService();

export default class Reg extends React.Component {
    render() {
        return <_Reg service={userService} />;
    }
}

@observer
class _Reg extends React.Component {
    handleClick(event) {
        event.preventDefault();
        let fm = event.target.form;
        this.props.service.reg(fm[0].value, fm[1].value, fm[2].value);
    }

    render() {
        if (this.props.service.loggedin){
            return <Redirect to="/" />;
        }
        return (
            <div className="login-page">
                <div className="form">
                    <form className="register-form">
                        <input type="text" placeholder="姓名" />

```

```
        <input type="text" placeholder="邮箱" />
        <input type="password" placeholder="密码" />
        <input type="password" placeholder="确认密码" />
        <button onClick={this.handleClick.bind(this)}>注册</button>
        <p className="message">如果已经注册 <Link to="/login">请登录</Link></p>
      </form>
    </div>
  </div>
);
}
```

Ant Design

Ant Design 蚂蚁金服开源的 React UI 库。

官网 <https://ant.design/index-cn>

官方文档 <https://ant.design/docs/react/introduce-cn>

安装

```
$ npm install antd
```

使用

```
import { List } from 'antd'; // 加载
import 'antd/lib/list/style/css'; // 加载 CSS

ReactDOM.render(<List />, mountNode);
```

每一种组件的详细使用例子，官网都有，需要时查阅官方文档即可

信息显示

网页开发中，不管操作成功与否，有很多提示信息，目前信息都是控制台输出，用户看不到。

使用Antd的message组件显示友好信息提示。

在service/user.js中增加一个被观察对象

```
import axios from 'axios';
import store from 'store'
import {observable} from 'mobx';

// 过期插件
store.addPlugin(require('store/plugins/expire'));

export default class UserService {
  @observable loggedIn = false;
  @observable errMsg = ''; // 被观察者

  login (email, password) {
    console.log(email, password);
```

```

    axios.post('/api/user/login', {
      email:email,
      password:password
    })/ * dev server会代理 */
    .then(
      response => { // 此函数要注意this的问题
        console.log(response.data);
        console.log(response.status);
        //+ 存储token
        store.set('token',
          response.data.token,
          (new Date()).getTime() + (8*3600*1000));
        this.loggedin = true;
      }
    ).catch(
      function (error) {
        console.log(error);
        this.errMsg = '登陆失败'; //+ 信息显示
      }
    )
  }
}

reg (name, email, password) {
  console.log(email, password);

  axios.post('/api/user/reg', {
    email:email,
    password:password,
    name:name
  })/ * dev server会代理 */
  .then(
    response => { // 此函数要注意this的问题
      console.log(response.data);
      console.log(response.status);
      //+ 存储token
      store.set('token',
        response.data.token,
        (new Date()).getTime() + (8*3600*1000));
      // 注册成功, 返回token即成功登录
      this.loggedin = true; //+ 修改被观察者
    }
  ).catch(
    error => {
      console.log(error);
      this.errMsg = '注册失败'; //+ 信息显示
    }
  )
}
}
}

```

component/_Login组件, 增加Antd的message组件

// 部分代码

```
import { message } from 'antd';
import 'antd/lib/message/style';

@observer
class _Login extends React.Component {
  handleClick(event) {
    event.preventDefault();
    let fm = event.target.form;
    this.props.service.login(
      fm[0].value, fm[1].value
    );
  }

  render() {
    console.log("~~~~~")
    if (this.props.service.loggedin) {
      return <Redirect to="/" />;
    }

    if (this.props.service.errMsg) {
      message.info(this.props.service.errMsg, 3,
        () => setTimeout(()=>this.props.service.errMsg = ''),1000);
    }

    return (<div className="login-page">
      <div className="form">
        <form className="login-form">
          <input type="text" placeholder="邮箱" value='tom@magedu.com' />
          <input type="password" placeholder="密码" />
          <button onClick={this.handleClick.bind(this)}>登录</button>
          <p className="message">还未注册? <Link to="/reg">请注册</Link></p>
        </form>
      </div>
    </div>);
  }
}
```

component/_Reg组件同样增加message组件

// 部分代码

```
import { message } from 'antd';
import 'antd/lib/message/style';

@observer
class _Reg extends React.Component {
  handleClick(event) {
    event.preventDefault();
    let fm = event.target.form;
    this.props.service.reg(fm[0].value, fm[1].value, fm[2].value);
  }
}
```

```
render() {  
  if (this.props.service.loggedin){  
    return <Redirect to="/" />;  
  }  
  
  if (this.props.service.errMsg) {  
    message.info(this.props.service.errMsg, 3,  
      () => setTimeout(()=>this.props.service.errMsg = ''),1000);  
  }  
  
  return (  
    <div className="login-page">  
      <div className="form">  
        <form className="register-form">  
          <input type="text" placeholder="姓名" />  
          <input type="text" placeholder="邮箱" />  
          <input type="password" placeholder="密码" />  
          <input type="password" placeholder="确认密码" />  
          <button onClick={this.handleClick.bind(this)}>注册</button>  
          <p className="message">如果已经注册 <Link to="/login">请登录</Link></p>  
        </form>  
      </div>  
    </div>  
  );  
}
```

