

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 expire from 'store/plugins/expire';
import { observable } from 'mobx';

// 过期插件
store.addPlugin(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, 注意需要重开一次chrome的调试窗口才能看到
        store.set('token',
          response.data.token,
          (new Date()).getTime() + (8 * 3600 * 1000));
        this.loggedin = true; //+ 修改被观察者
      }
    ).catch(
      error => {
        console.log(error);
      }
    )
  }

  reg(name, email, password) {
    console.log(name, 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, 注意需要重开一次chrome的调试窗口才能看到
    store.set('token',
      response.data.token,
      (new Date()).getTime() + (8 * 3600 * 1000));
    this.loggedin = true; //+ 修改被观察者
  }
).catch(
  error => {
    console.log(error);
  }
)
}
}

```

Reg组件类

```

import React from 'react';
import '../css/login.css';
import {Link, Redirect} from 'react-router-dom';
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 {
  validate(password, confirmpwd) {
    // 表单验证函数
    return password.value === confirmpwd.value;
  }

  handleClick(event) {
    event.preventDefault();
    const [name, email, password, confirmpwd] = event.target.form;
    console.log(this.validate(password, confirmpwd)); // +要验证表单数据后才发往后台
    this.props.service.reg(name.value, email.value, password.value);
  }

  render() {
    if (this.props.service.loggedin){ // 已经登录的用户不允许注册
      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>
  );
}
}

```

测试

重新登录，登录成功后，点击注册链接，看看是否跳转到了首页？或先注册，注册成功后，点击登录，看看是否还能登录？

原因是什么？

原因就是构造并使用了不同的UserService实例。如何使用同一个？

在service/user.js中导出唯一的UserService实例即可，其他模块直接导入并使用这个实例即可。

```

// service/user.js
class UserService {
  // 此处省略
}

const userService = new UserService();

export {userService};

```

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antd的组件
import 'antd/lib/list/style/css'; // 加载 组件的样式 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 expire from 'store/plugins/expire';
import { observable } from 'mobx';

// 过期插件
store.addPlugin(expire);

class UserService {
  @observable loggedin = false; //+ 被观察者
  @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，注意需要重开一次chrome的调试窗口才能看到
        store.set('token',
          response.data.token,
          (new Date()).getTime() + (8 * 3600 * 1000));
        this.loggedin = true; //+ 修改被观察者
      }
    ).catch(
      error => {
        console.log(error);
        this.errMsg = '登录失败' //+ 信息显示
      }
    )
  }
}
```

```

reg(name, email, password) {
  console.log(name, 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, 注意需要重开一次chrome的调试窗口才能看到
      store.set('token',
        response.data.token,
        (new Date()).getTime() + (8 * 3600 * 1000));
      this.loggedin = true; //+ 修改被观察者
    }
  ).catch(
    error => {
      console.log(error);
      this.errMsg = '登录失败' //+ 信息显示
    }
  )
}
}

const userService = new UserService();

export { userService };

```

component/_Login组件, 增加Antd的message组件

```

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';
import { message } from 'antd';

import 'antd/lib/message/style';

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

@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() {
    if (this.props.service.loggedin) {
      return <Redirect to="/" />; //+ 跳转
    }

    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='wayne@magedu.com' />
          <input type="password" placeholder="密码" value='abc1' />
          <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>);
  }
}

```

上面代码执行到用户登录失败时，浏览器控制台会抛出一个警告

```

Warning: _renderNewRootComponent(): Render methods should be a pure function of props and state;
triggering nested component updates from render is not allowed. If necessary, trigger nested
updates in componentDidUpdate. Check the render method of _Login.

```

解决方法

将消息的清除代码移动到 componentDidUpdate 中。

```

//component/login.js
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';
import { message } from 'antd';

import 'antd/lib/message/style';

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

```

```

    }
  }

  @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() {
      if (this.props.service.loggedin) {
        return <Redirect to="/" />; //+ 跳转
      }

      let em = this.props.service.errMsg;

      return (<div className="login-page">
        <div className="form">
          <form className="login-form">
            <input type="text" placeholder="邮箱" value='wayne@magedu.com' />
            <input type="password" placeholder="密码" value='abc1' />
            <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>);
    }

    componentDidUpdate(prevProps, prevState) { //+ 渲染后显示消息组件
      if (prevProps.service.errMsg) {
        message.info(prevProps.service.errMsg, 3, () => prevProps.service.errMsg = '');
      }
    }
  }
}

```

component/_Reg组件同样增加message组件

```

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

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

```

```

    }
  }

  @observer
  class _Reg extends React.Component {
    validate(password, confirmpwd) {
      // 表单验证函数
      return password.value === confirmpwd.value;
    }

    handleClick(event) {
      event.preventDefault();
      const [name, email, password, confirmpwd] = event.target.form;
      console.log(this.validate(password, confirmpwd)); // +要验证表单数据后才发往后台
      this.props.service.reg(name.value, email.value, password.value);
    }

    render() {
      if (this.props.service.loggedin){ // 已经登录的用户不允许注册
        return <Redirect to="/" />;
      }

      let em = this.props.service.errMsg;

      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>
      );
    }

    componentDidUpdate(prevProps, prevState) { //+ 渲染后显示消息组件
      if (prevProps.service.errMsg) {
        message.info(prevProps.service.errMsg, 3, () => prevProps.service.errMsg = '');
      }
    }
  }
}

```