import React, { Component } from 'react'

import { NavBar, Icon, WhiteSpace, WingBlank, List, Radio, Button, Result, Card } from 'antd-mobile'

import { getquestion } from '../../api/api'

import './homework.sass'

// 单选框

const RadioItem = Radio.RadioItem;

// 支付结果

const myImg = src => <img src={src} className="spe am-icon am-icon-lg" alt="" />;

// 正确值

let rightvalue, currentscore;

export default class Homework extends Component {

constructor() {

super();

this.state = {

questions: [

{ id: "1", question: "", a: "", b: "", c: "", d: '', score: 1 }

],

index: 1,

// 选项

value: '',

btnnext: '下一题',

btnpre: '上一题',

score:0,

right:0,

btnstatus: true,

valuebox: [

{ num: 1, value: '', rightvalue: '', currentscore: 0 },

{ num: 2, value: '', rightvalue: '', currentscore: 0 },

{ num: 3, value: '', rightvalue: '', currentscore: 0 },

{ num: 4, value: '', rightvalue: '', currentscore: 0 },

{ num: 5, value: '', rightvalue: '', currentscore: 0 }

]

}

}

componentDidMount() {

getquestion().then(res => {

this.setState({

questions: res.data

})

})

}

onChange = (value) => {

this.setState({

value,

});

};

// 渲染题目

works() {

let { index, questions, valuebox, score, right } = this.state;

// 用索引值index来判断内容显示题目，还是结果

if (index <= 5) {

let obj = questions[index - 1];

let answers = [

{ value: 'a', label: obj.a },

{ value: 'b', label: obj.b },

{ value: 'c', label: obj.c },

{ value: 'd', label: obj.d }

];

// 保存正确答案

rightvalue = obj.answer;

// 保存分数

currentscore = obj.score

// 返回渲染页面

return (

<WingBlank size="lg">

<List renderHeader={() => obj.question}>

{answers.map(i => (

<RadioItem key={i.value} checked={this.state.value === i.value} onChange={() => this.onChange(i.value)}>

{i.label}

</RadioItem>

))}

</List>

<WhiteSpace size="lg" />

<Button onClick={this.precommit.bind(this)} disabled={this.state.btnstatus} style={{ background: "#67C23A", color: "#fff" }}>{this.state.btnpre}</Button>

<WhiteSpace size="lg" />

<Button onClick={this.nextcommit.bind(this)} style={{ background: "#67C23A", color: "#fff" }}>{this.state.btnnext}</Button>

</WingBlank>

)

} else {

return (

<WingBlank size="lg">

<Result

img={myImg(require('../../assets/images/icon-homework.png'))}

title="答题结束"

message={<p style={{ color: 'red', fontSize: '1.5rem' }}>本次作业得分：{score}分，答对{right}题</p>}

/>

{/\* 题目渲染 \*/}

<WhiteSpace size="lg" />

<Card>

<Card.Header

title="解析"

/>

<Card.Body>

<ul>

{

this.state.questions.map(obj => (

<li>

<p>题目：{obj.question}</p>

<p style={{ color: 'red' }}>正确答案：{obj[obj.answer]}</p>

</li>

))

}

</ul>

</Card.Body>

</Card>

<WhiteSpace size="lg" />

</WingBlank>

)

}

}

// 下一题提交事件

nextcommit() {

let { index, score, right, valuebox, value, btnnext } = this.state;

// 判断上一题按钮的状态

if (index > 0) {

this.setState({

btnstatus: false

})

}

// 提交作业

if (index === 4) {

this.setState({

btnnext: '结束答题'

})

}

// 改变索引值

this.setState({

index: index + 1

})

// 改变状态值

this.setState({

// 更新答案盒子

valuebox: valuebox.map(obj => {

if (obj.num === index) {

obj.value = value;

obj.rightvalue = rightvalue;

obj.currentscore = currentscore;

}

return obj;

})

})

if (btnnext !== '结束答题') {

// 值状态

this.setState({

value: valuebox[index].value

})

}else{

// 判断作业对错

for (const obj of valuebox) {

if (obj.value === obj.rightvalue) {

score = score + parseInt(obj.currentscore);

right=right+1;

}

}

this.setState({

score: score,

right: right

})

}

}

// 上一题提交事件

precommit() {

let { index, valuebox, value } = this.state;

// 当第一题时，禁用上一题按钮

if (index === 2) {

this.setState({

btnstatus: true

})

}

// 将结束按钮变为下一题

if (index === 5) {

this.setState({

btnnext: '下一题'

})

}

// 得到当前对象的答案框

let currentvalue = valuebox.filter(obj => obj.num === index - 1)

this.setState({

index: index - 1,

value: currentvalue[0].value, // 读取当前题目存放的value值

// 更新答案盒子

valuebox: valuebox.map(obj => {

if (obj.num === index) {

obj.value = value;

obj.rightvalue = rightvalue;

obj.currentscore = currentscore;

}

return obj;

})

})

}

render() {

return (

<div className='homeworkbox'>

<NavBar

mode="dark"

icon={<Icon type="left" />}

style={{ background: "#67C23A" }}

onLeftClick={() => { this.props.history.push('/') }}

rightContent={`${this.state.index < 5 ? this.state.index : 5}/5`}//判断索引是否大于5

>今日作业</NavBar>

{/\* 今日作业\*/}

<WhiteSpace size="lg" />

<WhiteSpace size="lg" />

<WhiteSpace size="lg" />

{this.works()}

</div>

)

}

}