Diagram, schematic

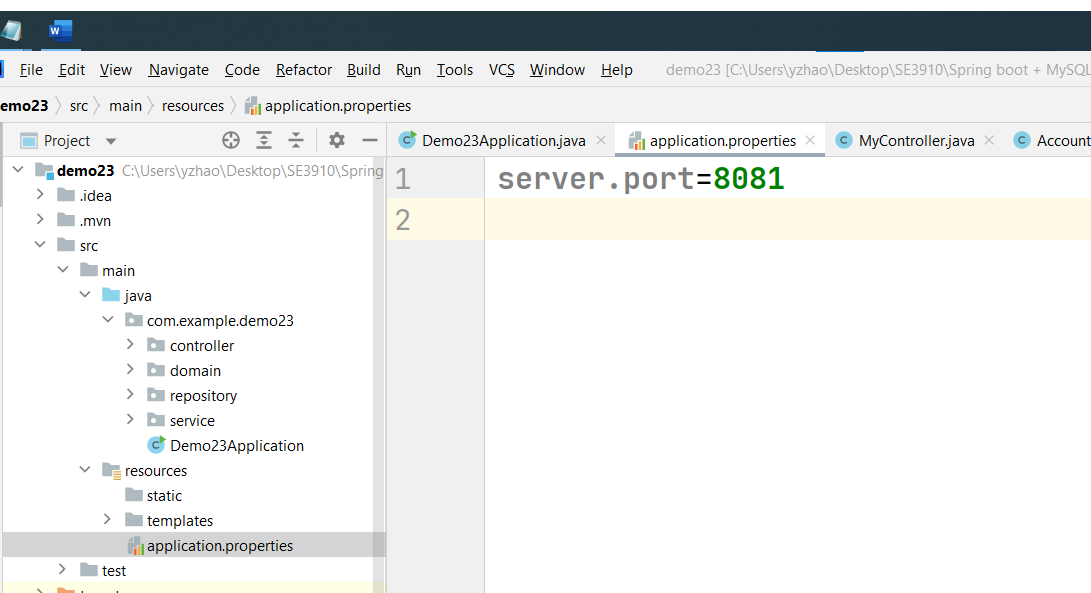
Description automatically generated

Controller 收发逻辑

Service： business logic

Domain： 数据结构

Repository： 数据库逻辑

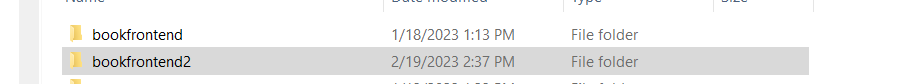


这是要和TK的reactjs的隔开。

####

我把tk的存在这里了

C:\Users\yzhao\class3910\reactjsapp\bookfrontend>



新的让我存在bookfrontend2 里面了

Graphical user interface, application

Description automatically generated

重新open一下



简单介绍一下这个code：



一般来说，这个index都是程序的起点。

Text

Description automatically generated

这个id是root连在后面js上。

Text

Description automatically generated

首先会进入这个index.js里面。

A screenshot of a computer

Description automatically generated with medium confidence

这个getElementById(‘root’)是语法，意思是这个要打开这个App



这个app在这里呢

Graphical user interface, application

Description automatically generated

这个app里面有一些link

A screenshot of a computer

Description automatically generated with medium confidence

path="/login"

这个path指的是

Graphical user interface, Word

Description automatically generated



这个component指的



Graphical user interface, application

Description automatically generated with medium confidence

这玩意在这呢

A screenshot of a computer

Description automatically generated with medium confidence

里面非常简单，这也是为什么Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

这个header里面是有menu的

Text

Description automatically generated

后面要是加个menu item就在这加

比如说，我们想加个menu item，

Student

Text

Description automatically generated

我们就先加这么一行

不用restart，这个会自动变。

Graphical user interface, text, application

Description automatically generated

只不过点了student还是会去login

Graphical user interface, application

Description automatically generated

我们接下来要改造这个Login page

Graphical user interface, text, application

Description automatically generated

Rename it to be student

Text

Description automatically generated

把里面的东西都给改成student

Text

Description automatically generated

要去app.js里面加一下链接

Text

Description automatically generated

回到header再给改成

Text

Description automatically generated

就可以用了。

Graphical user interface, text, application, chat or text message

Description automatically generated

这回点击student会出现student page

<https://www.w3schools.com/REACT/react_usestate.asp>

Graphical user interface, text, application

Description automatically generated

Color是变量，

setColor是个函数，这个函数可以update 这个变量

Graphical user interface, text, application

Description automatically generated

这个color可以直接调用

Background pattern

Description automatically generated with low confidence

还可以找个按钮用，setColor来改。

Graphical user interface, text, application

Description automatically generated

还可以做attribute

Text

Description automatically generated

用这个updateColor可以做到只改一个attribute color



updateColor

  const[student, setStudent] = useState({

    first:'',

    middle:'',

    last:'',

  });

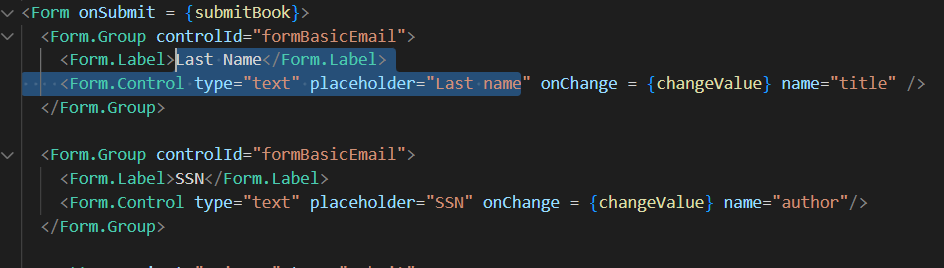
我加了一段userState

      <p> Student first name: {student.first}</p>

      <p> Student middle name: {student.middle}</p>

      <p> Student Last name: {student.last}</p>

在结尾也加了几个print



接着改了几个text

Graphical user interface, application

Description automatically generated

接下来看一看form的。

<https://www.w3schools.com/REACT/react_forms.asp>

Text

Description automatically generated

这个onchange的意思是把name的text编程他输入的。

Text

Description automatically generated

Handlesubmit的意思是点完按钮就会run这个handleSubmit

Chrome 如何打开console

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

Table

Description automatically generated

这个changevalue每点一下都会往console上发一个

Graphical user interface, diagram

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

They the same name

Text

Description automatically generated

这个会让默认的进入Home

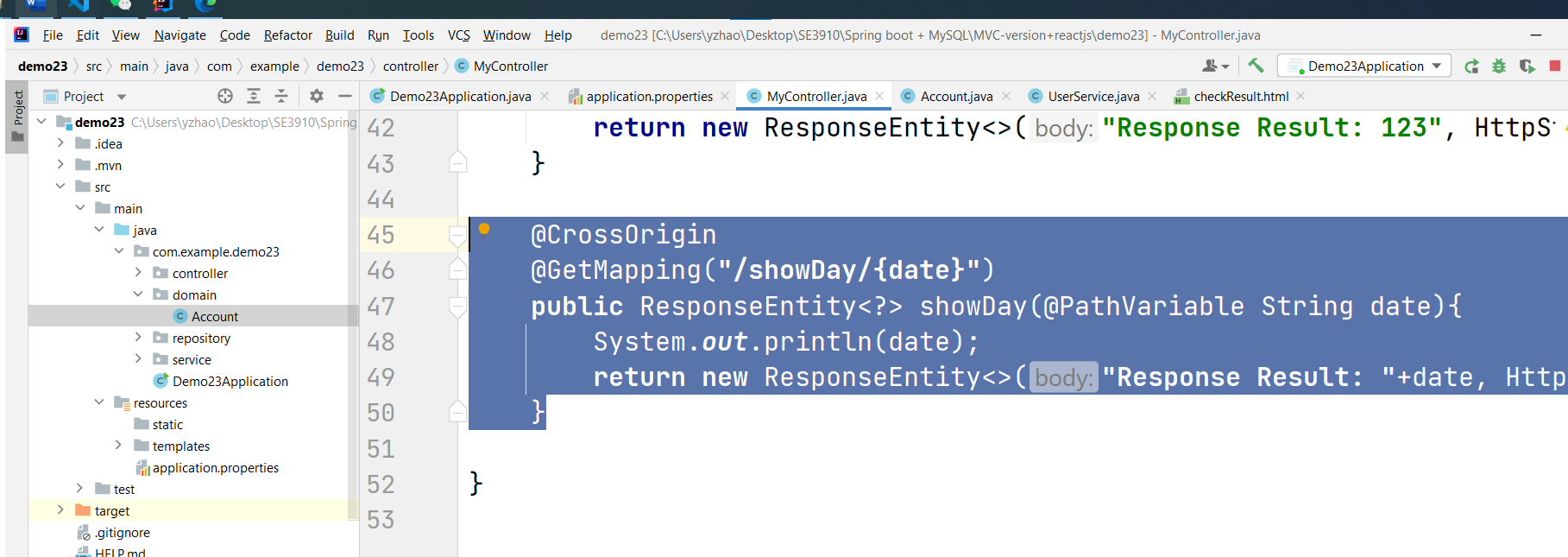
Graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated

我们可以把这个useEffect删掉，省着天天报错



import {useState} from 'react';

import Calendar from 'react-calendar';

//import './App.css';

import 'react-calendar/dist/Calendar.css'

function Student(props) {

  const [date, setDate] = useState(new Date())

    let [showResult, setShowResult] = useState("")

    const submitDay =()=>{

        console.log("day " + date.toDateString())

        fetch('http://localhost:8081/showDay/'+ date.toDateString(),{

            method:'GET',

        })

            .then(res=> {

                    if (res.ok) {

                        res.text().then(text => {

                            console.log(text);

                            setShowResult(text);

                        });

                    }

                }

            )

    }

    return (

        <div className="app">

            <h1 className="header">React Calendar</h1>

            <div className="calendar-container">

                <Calendar onChange={setDate} value={date} onClickDay={() => submitDay()}/>

            </div>

            <div className="text-center">

                Selected date: {date.toDateString()}

                Result date: {showResult}

            </div>

        </div>

    )

}

export default Student;

Calender

import {useState} from 'react';

import Calendar from 'react-calendar';

//import './App.css';

import ReactDOM from "react-dom/client";

import 'react-calendar/dist/Calendar.css'

function App(props) {

    const [date, setDate] = useState(new Date())

    let [showResult, setShowResult] = useState("")

    const submitDay =()=>{

        console.log("day " + date.toDateString())

        fetch('http://localhost:8080/showDay/'+ date.toDateString(),{

            method:'GET',

        })

            .then(res=> {

                    if (res.ok) {

                        res.text().then(text => {

                            console.log(text);

                            setShowResult(text);

                        });

                    }

                }

            )

    }

    return (

        <div className="app">

            <h1 className="header">React Calendar</h1>

            <div className="calendar-container">

                <Calendar onChange={setDate} value={date} onClickDay={() => submitDay()}/>

            </div>

            <div className="text-center">

                Selected date: {date.toDateString()}

                Result date: {showResult}

            </div>

        </div>

    )

}

const root = ReactDOM.createRoot(document.getElementById('root1'));

root.render(<App />);

这个是work的，体力不支。要休息休息

Text, application, chat or text message

Description automatically generated

Graphical user interface, application

Description automatically generated

Student

import React, { useState } from 'react';

import { Form, Button } from 'react-bootstrap';

import { propTypes } from 'react-bootstrap/esm/Image';

function Student(props) {

  const[student, setStudent] = useState({

    firstname:'',

    lastname:'',

    ssn:'',

  });

  const[book, setBook] = useState({

    title:'',

    author:'',

  });

  const changeValue=(e)=>{

    console.log(e);

    setStudent({

     ...student, [e.target.name]:e.target.value

    });

    console.log(e.target.name + " name "  );

    console.log(e.target.value + " value " );

  }

  const submitBook =(e)=>{

    e.preventDefault();

    fetch("http://localhost:8081/student", {

      method:"PUT",

      headers:{

        "Content-Type" : "application/json"

      },

      body: JSON.stringify(student)

    })

    .then(res=>{

        console.log(1,res);

        if(res.status === 201){

          return res.json();

        }else{

          return null;

        }

      })

    .then(res=>{

      console.log(res)

      if(res!==null){

        props.history.push('/login');

      }else{

        alert('fails');

      }

    });

  }

  return (

    <div>

<Form onSubmit = {submitBook}>

  <Form.Group controlId="formBasicEmail">

    <Form.Label>Last Name</Form.Label>

    <Form.Control type="text" placeholder="Last name" onChange = {changeValue} name="last" />

  </Form.Group>

  <Form.Group controlId="formBasicEmail">

    <Form.Label>SSN</Form.Label>

    <Form.Control type="text" placeholder="SSN" onChange = {changeValue} name="ssn"/>

  </Form.Group>

  <Button variant="primary" type="submit">

    Submit

  </Button>

</Form>

      <p> Student first name: {student.firstname}</p>

      <p> Student Last name: {student.lastname}</p>

    </div>

  );

}

export default Student;

I just change student to calendar

Text

Description automatically generated with medium confidence

A picture containing logo

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

App.js

Text

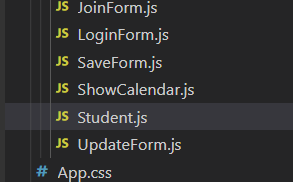
Description automatically generated

Header.js

Text

Description automatically generated

I created a student.js again



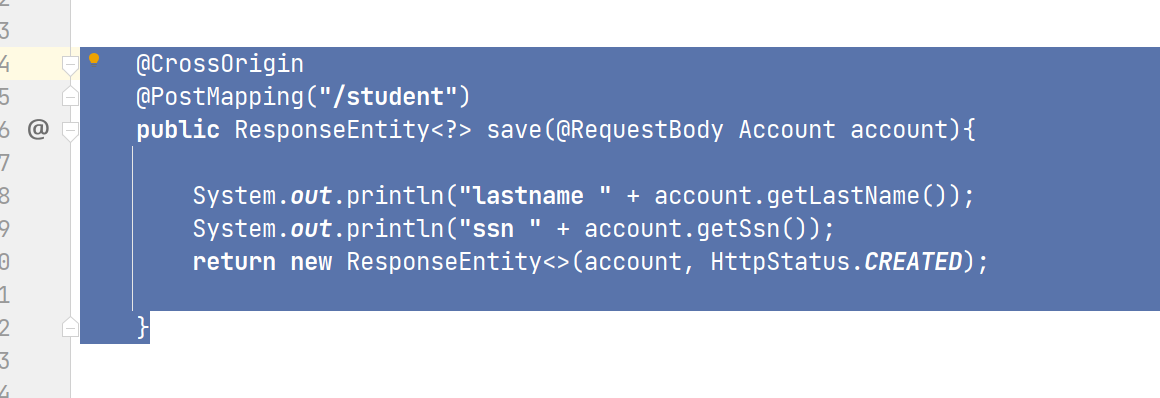
Text

Description automatically generated

Text

Description automatically generated

In the server:



This code can receive the request from the client.

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

However the server needs to have the same class attributes to client

Text

Description automatically generated

  const changeValue=(e)=>{

    console.log(e);

    setAcount({

     ...acount, [e.target.name]:e.target.value

    });

    console.log(e.target.name + " name "  );

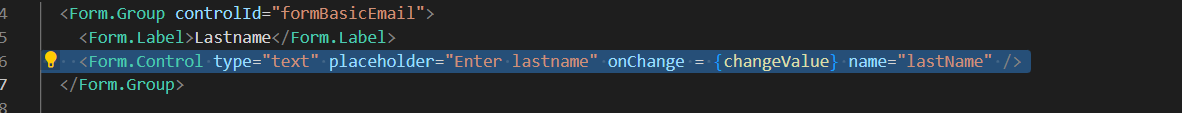
    console.log(e.target.value + " value " );

  }

这个地方是标准语法，可以来change

Text

Description automatically generated



调用的地方在这里，完全不用改。

@CrossOrigin  
@PostMapping(**"/student"**)  
**public** ResponseEntity<?> save(@RequestBody Account account){  
  
 System.***out***.println(**"lastname "** + account.getLastName());  
 System.***out***.println(**"ssn "** + account.getSsn());  
 account.setFirstName(**"ASD"**);  
 **return new** ResponseEntity<>(account, HttpStatus.***OK***);  
  
}

Text

Description automatically generated

Text

Description automatically generated

有不懂的问题可以问chatGPT

哈哈

Text

Description automatically generated

做一个button发出去

A screenshot of a computer

Description automatically generated

这个地方来接

然后做一个按钮，统计一下字数。