App({  
 onLaunch: function () {  
 wx.hideTabBar()  
 wx.setStorage({  
 data: this.globalData.iconlist,  
 key: 'icon',  
 })  
 var logs = wx.getStorageSync('logs') || []  
 logs.unshift(Date.now())  
 wx.setStorageSync('logs', logs)  
 wx.login({  
 success: res => {  
 }  
 })  
 wx.getSetting({  
 success: res => {  
 if (res.authSetting['scope.userInfo']) {  
 wx.getUserInfo({  
 success: res => {  
 this.globalData.userInfo = res.userInfo  
 if (this.userInfoReadyCallback) {  
 this.userInfoReadyCallback(res)  
 }  
 }  
 })  
 }  
 }  
 })  
 },  
 editTabbar: function () {  
 let tabbar = this.globalData.tabBar;  
 let currentPages = getCurrentPages();  
 let \_this = currentPages[currentPages.length - 1];  
 let pagePath = \_this.route;  
 (pagePath.indexOf('/') != 0) && (pagePath = '/' + pagePath);  
 for (let i in tabbar.list) {  
 tabbar.list[i].selected = false;  
 (tabbar.list[i].pagePath == pagePath) && (tabbar.list[i].selected = true);  
 }  
 \_this.setData({  
 tabbar: tabbar  
 });  
 },  
 tabBar: {  
 "backgroundColor": "#ffffff",  
 "color": "#979795",  
 "selectedColor": "#1c1c1b",  
 "list": [{  
 "pagePath": "/pages/money/money",  
 "text": "明细",  
 "iconPath": "icon/detail.png",  
 "selectedIconPath": "icon/detail1.png"  
 },  
 {  
 "pagePath": "/pages/addmoney/addmoney",  
 "text": "添加",  
 "isSpecial": true,  
 "iconPath": "icon/add1.png",  
 "selectedIconPath": "icon/add1.png"  
 },  
 {  
 "pagePath": "/pages/figure/figure",  
 "text": "图表",  
 "iconPath": "icon/figure.png",  
 "selectedIconPath": "icon/figure1.png"  
 }  
 ]  
 }  
 }  
})const app = getApp()  
import {  
 addMoney  
} from './../service/api'  
var util = require('../../utils/util')  
Page({  
 data: {  
 val: "",  
 time: '2020-07-30',  
 time\_copy: '',  
 switchlist: ["支出", "收入"],  
 money: {  
 name: '奖金',  
 flag: 0,  
 tag: 0,  
 money: 0,  
 time: '',  
 remark: '',  
 usr: '',  
 year: '',  
 month: '',  
 day: ''  
 },  
 isTag: 0  
 },  
 onLoad: function () {  
 var r0 = util.formatYMD(new Date(), 1)  
 var r1 = util.formatYMD(new Date(), 0)  
 this.data.money["usr"] = app.globalData.userInfo.nickName  
 this.data.money["time"] = r0  
 this.data.money["year"] = r1[0]  
 this.data.money["month"] = r1[1]  
 this.data.money["day"] = r1[2]  
 this.setData({  
 money: this.data.money,  
 time: r0,  
 time\_copy: r0  
 })  
 },  
 moneyFun: async function (e) {  
 this.data.money.money = Number(e.detail.val)  
 var r0 = util.formatYMD(new Date(), 1)  
 var r1 = util.formatYMD(new Date(), 0)  
 this.data.money["usr"] = app.globalData.userInfo.nickName  
 this.data.money["time"] = r0  
 this.data.money["year"] = r1[0]  
 this.data.money["month"] = r1[1]  
 this.data.money["day"] = r1[2]  
 this.setData({  
 val: e.detail.val,  
 money: this.data.money  
 })  
 if (e.detail.sub == 1) {  
 var res = await addMoney(this.data.money)  
 console.log(res);  
 this.data.money.flag = 0  
 this.setData({  
 isTag: 0,  
 money: this.data.money  
 })  
 this.goBack()  
 }  
 },  
 iconSwitch: function (e) {  
 this.data.money.tag = e.detail.index  
 this.data.money.name = e.detail.name  
 this.setData({  
 money: this.data.money  
 })  
 },  
 switchTag: function (e) {  
 this.data.money.flag = e.detail  
 this.setData({  
 money: this.data.money  
 })  
 },  
 bindDateChange: function (e) {  
 var res = e.detail.value.split('-')  
 this.data.money.year = res[0]  
 this.data.money.month = res[1]  
 this.data.money.day = res[2]  
 this.data.money.time = e.detail.value  
 this.setData({  
 time: e.detail.value,  
 money: this.data.money  
 })  
 },  
 goBack: function () {  
 this.setData({  
 val: 0,  
 isTag: 0,  
 time: this.data.time\_copy  
 })  
 wx.switchTab({  
 url: '../money/money',  
 success: function () {  
 var page = getCurrentPages().pop();  
 if (page == undefined || page == null) return;  
 page.onLoad();  
 }  
 })  
 }  
})const app = getApp()  
import {  
 postBud  
} from '../service/api'  
Page({  
 data: {  
 budget: 500,  
 btnblur: 0,  
 had: 0  
 },  
 onLoad: function (option) {  
 console.log(option)  
 this.setData({  
 had: option.had,  
 budget: option.bud  
 })  
 },  
 changeShow: function () {  
 this.setData({  
 btnblur: 1  
 })  
 },  
 setSuc: async function () {  
 this.setData({  
 btnblur: 0  
 })  
 var res = await postBud(app.globalData.userInfo.nickName, this.data.budget)  
 console.log(res)  
 wx.reLaunch({  
 url: '/pages/figure/figure?budget'  
 })  
 }  
})const app = getApp()  
import {  
 addTodo  
} from './../../service/api'  
var util = require('../../../utils/util')  
Component({  
 properties: {  
 title: {  
 type: String  
 },  
 isClose: {  
 type: Boolean  
 }  
 },  
 data: {  
 isTag: -1,  
 detail: ""  
 },  
 methods: {  
 addSubmit: async function () {  
 console.log(this.data.detail);  
 var obj = {  
 name: app.globalData.userInfo.nickName,  
 time: util.formatYMD(new Date(), 1),  
 detail: this.data.isDetail,  
 flag: 0,  
 tag: this.data.isTag  
 }  
 const res = await addTodo(obj)  
 console.log(res)  
 this.triggerEvent('addSuc', {  
 obj: obj,  
 tag: this.data.isTag  
 })  
 this.changeDel()  
 },  
 changeTag: function (e) {  
 this.data.isTag = e.currentTarget.dataset.tags  
 this.setData({  
 isTag: this.data.isTag  
 })  
 },  
 changeDel: function () {  
 this.setData({  
 isClose: false,  
 isTag: -1  
 })  
 this.triggerEvent('addClose', {  
 add: false  
 })  
 }  
 }  
})Component({  
 properties: {  
 iconShow: {  
 type: Number,  
 value: 0  
 }  
 },  
 data: {},  
 methods: {  
 addSuc: function (e) {  
 var flag = e.currentTarget.dataset.flag  
 if (flag == 1) {  
 this.triggerEvent('addIcon', this.data.newTag)  
 }  
 this.setData({  
 iconShow: 0,  
 newTag: ''  
 })  
 }  
 }  
})var util = require('../../../utils/util')  
Component({  
 properties: {  
 income: {  
 type: Number,  
 value: 0  
 },  
 expenditure: {  
 type: Number,  
 value: 0  
 },  
 budget: {  
 type: Number,  
 value: 500  
 },  
 subbudget: {  
 type: Number,  
 value: 500  
 },  
 type: {  
 type: Number,  
 value: 0  
 },  
 expval:{  
 type:Number,  
 value:0  
 }  
 },  
 data: {  
 year: '',  
 month: ''  
 },  
 attached: function () {  
 var res = util.formatYM(new Date())  
 this.setData({  
 year: res[0],  
 month: res[1]  
 })  
 },  
 methods: {  
 bindDateChange: function (e) {  
 var [year, month] = e.detail.value.split('-')  
 this.setData({  
 year: year,  
 month: month  
 })  
 this.triggerEvent('reqMoney', {  
 year: this.data.year,  
 month: this.data.month  
 })  
 },  
 setBudget: function () {  
 wx.navigateTo({  
 url: `/pages/budget/budget?had=${this.data.expval}&bud=${this.data.budget}`,  
 })  
 }  
 }  
})Component({  
 properties: {  
 },  
 data: {  
 },  
 ready: function () {  
 let ring = this.selectComponent('#ring\_id')  
 ring.showRing()  
 },  
 methods: {  
 close: function () {  
 this.triggerEvent('close')  
 }  
 }  
})const app = getApp()  
Component({  
 properties: {  
 itemList: {  
 type: Object,  
 value: {}  
 },  
 type: {  
 type: Number,  
 value: 0  
 },  
 total: {  
 type: Number,  
 value: 0  
 },  
 typeTitle:{  
 type:String  
 }  
 },  
 data: {  
 },  
 lifetimes: {  
 ready: function () {   
 var that=this  
 setTimeout(function () {  
 that.data.typeTitle = app.globalData.iconlist[that.data.itemList.tag].name  
 that.data.percent = (that.data.itemList.val / that.data.total \* 100).toFixed(2)  
 that.setData({  
 typeTitle: that.data.typeTitle,  
 percent: that.data.percent  
 })  
 }, 1000)  
 },  
 },  
 methods: {  
 }  
})Component({  
 properties: {  
 num:{  
 type:String,  
 value:'0'  
 }  
 },  
 data: {  
 dot: false  
 },  
 methods: {  
 tapkey: function (v) {  
 var val = v.target.dataset.key  
 if (val == ".") {  
 if (this.data.dot) return  
 this.setData({  
 dot: true  
 })  
 }  
 this.setData({  
 num: this.data.num == '0' ? val : this.data.num + val  
 })  
 this.triggerEvent('moneySuc', {  
 val: this.data.num,  
 sub: 0  
 })  
 },  
 subkey: function () {  
 this.triggerEvent('moneySuc', {  
 val: this.data.num,  
 sub: 1  
 })  
 this.setData({  
 num: '0',  
 dot: false  
 })  
 },  
 delkey: function () {  
 if (this.data.num == '0') return  
 if (this.data.num[this.data.num.length - 1] == '.') this.setData({  
 dot: false  
 })  
 this.setData({  
 num: this.data.num.length == 1 ? '0' : this.data.num.substring(0, this.data.num.length - 1)  
 })  
 this.triggerEvent('moneySuc', {  
 val: this.data.num,  
 sub: 0  
 })  
 },  
 clearkey: function () {  
 this.setData({  
 num: '0',  
 dot: false  
 })  
 this.triggerEvent('moneySuc', {  
 val: this.data.num,  
 sub: 0  
 })  
 }  
 }  
})Component({  
 properties: {  
 curIndex: {  
 type: Number,  
 value: 0  
 }  
 },  
 data: {  
 iconShow: 0,  
 select: 0,  
 iconlist: []  
 },  
 attached: async function () {  
 var res = await wx.getStorageSync('icon')  
 this.setData({  
 iconlist: res  
 })  
 this.swiper(this.data.iconlist, 10)  
 },  
 methods: {  
 iconSuc: function (e) {  
 var index = e.currentTarget.dataset.index  
 if (index == 11) {  
 this.setData({  
 iconShow: 1  
 })  
 } else {  
 this.setData({  
 select: index  
 })  
 this.triggerEvent('iconSwitch', {  
 index: index,  
 name: e.currentTarget.dataset.name  
 })  
 }  
 },  
 addIcon: function (e) {  
 var obj = {  
 id: 12,  
 name: e.detail  
 }  
 var len = this.data.iconlist.length  
 this.data.iconlist.splice(len - 1, 0, obj)  
 wx.setStorageSync('icon', this.data.iconlist)  
 this.setData({  
 iconlist: this.data.iconlist  
 })  
 this.swiper(this.data.iconlist, 10)  
 },  
 swiper(arr, size) {  
 let that = this  
 var length = that.data.iconlist.length;  
 var page = Math.ceil(length / size);  
 var newData = []  
 var start = 0  
 for (var i = 1; i < page; i++) {  
 var newArr = arr.slice(start, i \* size)  
 newData.push(newArr)  
 var remainder = length - i \* size  
 start = i \* size  
 }  
 if (remainder > size) {  
 return  
 } else {  
 var newArr = arr.slice(start, length)  
 newData.push(newArr)  
 that.setData({  
 newData: newData  
 })  
 }  
 }  
 }  
})Component({  
 properties: {  
 moneylist:{  
 type:Object  
 }  
 },  
 data: {  
 },  
 methods: {  
 }  
})Component({  
 properties: {  
 switchlist: {  
 type: Array  
 },  
 isTag:{  
 type:Number  
 }  
 },  
 data: {  
 isTag: 0  
 },  
 methods: {  
 switchSuc: function (e) {  
 this.setData({  
 isTag: e.target.dataset.tag  
 })  
 this.triggerEvent('switchTag',this.data.isTag)  
 }  
 }  
})var windWidth = wx.getSystemInfoSync().windowWidth  
const app = getApp()  
import Animation from './../../../utils/animation'  
import {  
 measureText  
} from '../../../utils/util'  
Component({  
 properties: {  
 ringList: {  
 type: Array  
 },  
 total: {  
 type: Number,  
 value: 0  
 },  
 type: {  
 type: Number,  
 value: 0  
 }  
 },  
 data: {  
 width: windWidth,  
 canvasWidth: windWidth \* 0.5,  
 x\_position: windWidth / 2,  
 y\_position: 0,  
 height: windWidth \* 0.5 + 40,  
 index: 0,  
 exp\_color: ["#FCE38A", "#FFD487", "#FFC48E", "#ffde7d", "#f9ed69", "#fce38a", "#FFB59E"],  
 in\_color: ["#3c9099", "#1fab89", "#62d2a2", "#9df3c4", "#a7d7c5"],  
 angleList: [],  
 radius: 0,  
 empty: false  
 },  
 ready: function () {  
 var that = this  
 setTimeout(function () {  
 that.showRing(that.data.ringList, that.data.total, that.data.type)  
 }, 50)  
 },  
 methods: {  
 choicePart(e) {  
 var curx = e.touches[0].clientX  
 var cury = e.touches[0].clientY  
 var x = this.data.x\_position  
 var y = this.data.height / 2 + e.currentTarget.offsetTop  
 var radius\_in = this.data.height / 2 - this.data.radius / 2  
 var radius\_out = radius\_in + this.data.radius / 2  
 var angle = -1  
 var index = 0  
 var dis = Math.pow(curx - x, 2) + Math.pow(cury - y, 2)  
 if (dis <= Math.pow(radius\_out, 2) && dis >= Math.pow(radius\_in, 2)) {  
 angle = Math.atan(Math.abs(cury - y) / (Math.abs(curx - x)))  
 if (curx > x) {  
 if (cury < y) angle = 2 \* Math.PI - angle  
 } else {  
 if (cury < y) angle = Math.PI + angle  
 else angle = Math.PI - angle  
 }  
 }  
 for (let i = 0; i < this.data.angleList.length; i++) {  
 if (i == this.data.angleList.length - 1) {  
 if (angle >= this.data.angleList[i].startAngle) {  
 index = i  
 break  
 }  
 }  
 if (this.data.angleList[i].startAngle <= angle && this.data.angleList[i + 1].startAngle > angle) {  
 index = i  
 break  
 }  
 }  
 this.setData({  
 index: index  
 })  
 this.triggerEvent('figureSwitch', {  
 type: this.data.type,  
 index: this.data.index  
 })  
 },  
 cailPieAngle(series, count, process = 1) {  
 let startAngle = 0;  
 if (series.length == 0) {  
 return [{  
 startAngle: 0,  
 proportion: 1 \* process  
 }]  
 }  
 return series.map((item) => {  
 item.proportion = item.val / count \* process  
 item.startAngle = startAngle  
 startAngle += 2 \* Math.PI \* item.proportion  
 return item;  
 })  
 },  
 drawLegend(context, series, count, x, y, radius) {  
 var pieSeries = this.cailPieAngle(series, count, 1)  
 context.font = "35px sans-serif"  
 for (let i = 0; i < pieSeries.length; i++) {  
 var off\_x = 0,  
 off\_y = 0,  
 angle = 0,  
 legend\_x = 0,  
 legend\_y = 0,  
 text = ''  
 angle = pieSeries[i].startAngle + pieSeries[i].proportion \* Math.PI  
 text = `${app.globalData.iconlist[pieSeries[i].tag].name}${(pieSeries[i].proportion\*100).toFixed(2)}%`  
 off\_x = Math.cos(angle) \* radius  
 off\_y = Math.sin(angle) \* radius  
 var textwidth = measureText(text)  
 legend\_x = off\_x >= 0 ? off\_x + x : x + off\_x - textwidth  
 legend\_y = off\_y + y  
 context.beginPath()  
 context.fillStyle = '#999999'  
 context.fillText(text, legend\_x, legend\_y);  
 context.closePath();  
 }  
 },  
 drawPie(ctx, series, x, y, radius, count, line, process, flag) {  
 var pieSeries = this.cailPieAngle(series, count, process)  
 this.setData({  
 angleList: pieSeries  
 })  
 var that = this  
 pieSeries.forEach((item, index) => {  
 ctx.beginPath()  
 ctx.lineWidth = line  
 ctx.arc(x, y, radius, item.startAngle, item.startAngle + 2 \* Math.PI \* item.proportion)  
 if (pieSeries.length == 1 && this.data.total == 0) {  
 ctx.strokeStyle = "#a4a4a4"  
 } else {  
 if (flag == 0) {  
 if (index == pieSeries.length - 1 && index % 5 == 0) {  
 ctx.strokeStyle = that.data.exp\_color[1]  
 } else {  
 ctx.strokeStyle = that.data.exp\_color[index % that.data.exp\_color.length]  
 }  
 } else {  
 if (index == pieSeries.length - 1 && index % 5 == 0) {  
 ctx.strokeStyle = that.data.in\_color[1]  
 } else {  
 ctx.strokeStyle = that.data.in\_color[index % that.data.in\_color.length]  
 }  
 }  
 }  
 ctx.stroke()  
 })  
 },  
 showRing: function (itemlist, total, flag) {  
 const query = wx.createSelectorQuery().in(this)  
 query.select('#myCanvas')  
 .fields({  
 node: true,  
 size: true  
 })  
 .exec((res) => {  
 const canvas = res[0].node  
 const ctx = canvas.getContext('2d')  
 const dpr = wx.getSystemInfoSync().pixelRatio  
 var ori\_radius = this.data.canvasWidth / 2  
 canvas.width = res[0].width \* dpr  
 canvas.height = res[0].height \* dpr  
 const x = this.data.x\_position \* dpr  
 const y = this.data.height / 2 \* dpr  
 const radius = (ori\_radius - 10) \* dpr  
 const line = radius \* 0.5  
 this.setData({  
 radius: ori\_radius - 10  
 })  
 if (itemlist.length == 0) {  
 this.setData({  
 total: 0  
 })  
 }  
 let centerPosition = {  
 x: this.data.x\_position,  
 y: this.data.height / 2 + res[0].node.\_top  
 }  
 Animation({  
 duration: 1000,  
 onProcess: (process) => {  
 this.drawPie(ctx, itemlist, x, y, radius, this.data.total, line, process, flag)  
 this.drawLegend(ctx, itemlist, this.data.total, x, y, radius)  
 }  
 })  
 })  
 },  
 switchItem: function () {  
 this.data.type = !this.data.type  
 this.setData({  
 type: this.data.type,  
 index: 0  
 })  
 this.triggerEvent('figureSwitch', {  
 type: this.data.type,  
 index: this.data.index  
 })  
 this.showRing(this.data.ringList, this.data.total, this.data.type)  
 }  
 }  
})var windWidth = wx.getSystemInfoSync().windowWidth  
import Animation from './../../../utils/animation'  
Component({  
 properties: {  
 },  
 data: {  
 width: windWidth,  
 height: windWidth \* 0.5 + 40,  
 canvasWidth: windWidth \* 0.5,  
 x\_position: windWidth / 2,  
 y\_position: 0,  
 do: 5,  
 count: 10,  
 com: 50  
 },  
 ready: function () {  
 var that = this  
 setTimeout(function () {  
 that.showRing()  
 }, 50)  
 },  
 methods: {  
 draw: function (ctx, line, x, y, radius, process) {  
 ctx.beginPath()  
 ctx.lineWidth = line  
 console.log("\*\*\*\*\*\*\*\*\*\*\*", x, y, radius)  
 ctx.arc(x, y, radius, 0, 0 + 2 \* Math.PI \* 0.5 \* process)  
 ctx.strokeStyle = "#B39CD0"  
 ctx.stroke()  
 },  
 showRing: function () {  
 const query = wx.createSelectorQuery().in(this)  
 query.select('#myCanvas')  
 .fields({  
 node: true,  
 size: true  
 })  
 .exec((res) => {  
 const canvas = res[0].node  
 const ctx = canvas.getContext('2d')  
 const dpr = wx.getSystemInfoSync().pixelRatio  
 var ori\_radius = this.data.canvasWidth / 2  
 canvas.width = res[0].width \* dpr  
 canvas.height = res[0].height \* dpr  
 const x = this.data.x\_position \* dpr  
 const y = this.data.height / 2 \* dpr  
 const radius = (ori\_radius - 10) \* dpr  
 const line = radius \* 0.5  
 this.setData({  
 radius: ori\_radius - 10  
 })  
 ctx.beginPath()  
 ctx.lineWidth = line  
 console.log("\*\*\*\*\*\*\*\*\*\*\*", x, y, radius)  
 ctx.arc(x, y, radius, 0, 0 + 2 \* Math.PI)  
 ctx.strokeStyle = "#FBEAFF"  
 ctx.stroke()  
 Animation({  
 duration: 1000,  
 onProcess: (process) => {  
 this.draw(ctx, line, x, y, radius, process)  
 }  
 })  
 })  
 },  
 close:function(){  
 this.triggerEvent('close')  
 }  
 }  
})import {addTodo} from '../../service/api'  
Component({  
 properties:{  
 todoList:{  
 type:Array,  
 value:[]  
 },  
 title:{  
 type:String  
 }  
 },  
 data:{  
 time:'',  
 detail:'',  
 tag:'',  
 flag:'',  
 toggle:true  
 },  
 methods:{  
 submits:function(){  
 var addlist={  
 time:this.data.time,  
 detail:this.data.detail,  
 flag:this.data.flag,  
 tag:this.data.tag  
 }  
 addTodo(addlist)  
 },  
 switchflag:function(e){  
 this.data.todoList[e.currentTarget.dataset.index].flag=e.detail.value==true?1:0  
 this.setData({  
 todoList:this.data.todoList  
 })  
 console.log(this.data.todoList);  
 },  
 chToggle:function(){  
 this.data.toggle=!this.data.toggle  
 this.setData({  
 toggle:this.data.toggle  
 })  
 }  
 }  
})Page({})Page({})const app = getApp()  
var util = require('../../utils/util')  
import {  
 getFigure,  
 getBud  
} from './../service/api'  
Page({  
 data: {  
 tabbar: {},  
 in\_list: [],  
 in\_val: 0,  
 exp\_list: [],  
 exp\_val: 0,  
 choiceList: {},  
 ring\_list: [],  
 choiceTotal: 0,  
 choiceType: 0,  
 choiceIndex: 0,  
 budget: 500,  
 subbudget: 400,  
 had:0  
 },  
 onLoad: function (option) {  
 app.editTabbar()  
 this.reqMoney()  
 },  
 reqMoney: async function (e) {  
 var res = util.formatYM(new Date())  
 var obj = {  
 year: e ? e.detail.year : res[0],  
 month: e ? e.detail.month : res[1],  
 usr: app.globalData.userInfo.nickName  
 }  
 var ans = await getFigure(obj)  
 var ans2 = await getBud(app.globalData.userInfo.nickName)  
 var bud = ans2.usr[0].bud  
 var title = ''  
 var list = []  
 if (ans.expList.length > 0) {  
 list = ans.expList[0]  
 title = app.globalData.iconlist[ans.expList[0].tag].name  
 }  
 var sub = bud - ans.exp\_money  
 this.setData({  
 in\_val: ans.in\_money,  
 exp\_val: ans.exp\_money,  
 in\_list: ans.incomeList,  
 exp\_list: ans.expList,  
 choiceList: list,  
 choiceTotal: ans.exp\_money,  
 choiceType: 0,  
 typeTitle: title,  
 ring\_list: ans.expList,  
 budget: bud,  
 subbudget: sub  
 })  
 let ring = this.selectComponent('#ring\_id')  
 ring.showRing(this.data.ring\_list, this.data.choiceTotal, 0)  
 },  
 figureSwitch: function (e) {  
 var type = e.detail.type  
 var index = e.detail.index  
 var ring\_list = type ? this.data.in\_list : this.data.exp\_list  
 var total = type ? this.data.in\_val : this.data.exp\_val  
 var list = []  
 var typeTitle = ''  
 if (ring\_list.length > 0) {  
 list = type ? this.data.in\_list[index] : this.data.exp\_list[index]  
 typeTitle = app.globalData.iconlist[list.tag].name  
 }  
 this.setData({  
 choiceType: type,  
 choiceTotal: total,  
 choiceList: list,  
 typeTitle: typeTitle,  
 ring\_list: ring\_list  
 })  
 }  
})const app = getApp()  
Page({  
 data: {  
 tabbar: {},  
 userInfo: {},  
 hasUserInfo: false,  
 fontFamily: 'shouxieti',  
 canIUse: wx.canIUse('button.open-type.getUserInfo')  
 },  
 bindViewTap: function () {  
 wx.navigateTo({  
 url: '../logs/logs'  
 })  
 },  
 onLoad: function () {  
 app.editTabbar()  
 if (app.globalData.userInfo) {  
 this.setData({  
 userInfo: app.globalData.userInfo,  
 hasUserInfo: true  
 })  
 } else if (this.data.canIUse) {  
 app.userInfoReadyCallback = res => {  
 this.setData({  
 userInfo: res.userInfo,  
 hasUserInfo: true  
 })  
 }  
 } else {  
 wx.getUserInfo({  
 success: res => {  
 app.globalData.userInfo = res.userInfo  
 this.setData({  
 userInfo: res.userInfo,  
 hasUserInfo: true  
 })  
 }  
 })  
 }  
 },  
 getUserInfo: function (e) {  
 app.globalData.userInfo = e.detail.userInfo  
 this.setData({  
 userInfo: e.detail.userInfo,  
 hasUserInfo: true  
 })  
 }  
})const util = require('../../utils/util.js')  
Page({  
 data: {  
 logs: []  
 },  
 onLoad: function () {  
 this.setData({  
 logs: (wx.getStorageSync('logs') || []).map(log => {  
 return util.formatTime(new Date(log))  
 })  
 })  
 }  
})  
const app = getApp()  
import {  
 getMoney  
} from './../service/api'  
var util = require('../../utils/util')  
Page({  
 data: {  
 moneyList: [],  
 tabbar: {},  
 income: 0,  
 expenditure: 0  
 },  
 onLoad: function () {  
 this.reqMoney()  
 app.editTabbar()  
 },  
 reqMoney: async function (e) {  
 var res = util.formatYM(new Date(), 0)  
 var obj = {  
 year: e ? e.detail.year : res[0],  
 month: e ? e.detail.month : res[1],  
 usr: app.globalData.userInfo.nickName  
 }  
 var res = await getMoney(obj)  
 this.setData({  
 moneyList: res.newMoneyList,  
 income: res.all\_income,  
 expenditure: res.all\_exp  
 })  
 }  
})import request from './index'  
export function getTodo(params) {  
 return request('/getTodo', 'get', params)  
}  
export function addTodo(list) {  
 return request('/addTodo', 'post', list)  
}  
export function getMoney(params) {  
 return request('/getMoney', 'get', params)  
}  
export function getFigure(params) {  
 return request('/getFigure', 'get', params)  
}  
export function addMoney(list) {  
 return request('/addMoney', 'post', list)  
}  
export function getBud(params) {  
 return request(`/getBud?usr=${params}`, 'get')  
}  
export function postBud(usr, bud) {  
 return request(`/postBud?usr=${usr}&bud=${bud}`, 'post')  
}const baseURL = 'http://localhost:8080/test'  
const timeout = 5000  
function request(url, method, params) {  
 console.log(params);  
 wx.showLoading({  
 title: '数据加载中ing',  
 })  
 return new Promise((resolve, reject) => {  
 wx.request({  
 url: baseURL + url,  
 timeout: timeout,  
 method: method == 'post' ? 'post' : 'get',  
 data: params,  
 success: function (res) {  
 resolve(res.data.data)  
 },  
 fail: reject,  
 complete: res => {  
 wx.hideLoading()  
 }  
 })  
 })  
}  
export default request;const app = getApp()  
import {  
 getTodo  
} from '../service/api'  
var util = require('../../utils/util')  
Page({  
 data: {  
 p2List: [],  
 p1List: [],  
 p0List: [],  
 add: false,  
 com: false  
 },  
 onLoad: async function (options) {  
 var getdata = {  
 time: util.formatYMD(new Date(), 1),  
 name: app.globalData.userInfo.nickName  
 }  
 var res = await getTodo(getdata)  
 console.log("$$$$$$$$$$$$$$$", res)  
 this.setData({  
 p2List: res.p2List,  
 p1List: res.p1List,  
 p0List: res.p0List  
 })  
 },  
 changeImg: function () {  
 this.setData({  
 add: true  
 })  
 },  
 changeCom: function () {  
 this.setData({  
 com: true  
 })  
 },  
 comClose: function () {  
 this.setData({  
 com: false  
 })  
 },  
 addClose: function (e) {  
 this.data.add = e.detail.add  
 this.setData({  
 add: this.data.add  
 })  
 },  
 addSuc: function (e) {  
 const res = 'p' + e.detail.tag + 'List'  
 this.data[res].push(e.detail.obj)  
 this.setData({  
 [res]: this.data[res]  
 })  
 }  
})const app = getApp()  
import {  
 getTodo  
} from '../../service/api'  
var util = require('../../../utils/util')  
Page({  
 data: {  
 item: '今天',  
 p2List: [],  
 p1List: [],  
 p0List: [],  
 add: false  
 },  
 onLoad: async function (options) {  
 var str = ''  
 str = options.item == 'today' ? '今天' : '明天'  
 var getdata = {  
 time: util.formatYMD(new Date()),  
 name: app.globalData.userInfo.nickName  
 }  
 var res = await getTodo(getdata)  
 this.setData({  
 item: str,  
 p2List: res.p2List,  
 p1List: res.p1List,  
 p0List: res.p0List  
 })  
 },  
 changeImg: function () {  
 this.setData({  
 add: true  
 })  
 },  
 addClose: function (e) {  
 this.data.add = e.detail.add  
 this.setData({  
 add: this.data.add  
 })  
 },  
 addSuc: function (e) {  
 const res = 'p' + e.detail.tag + 'List'  
 this.data[res].push(e.detail.obj)  
 this.setData({  
 [res]: this.data[res]  
 })  
 }  
})Component({  
 properties: {  
 tabbar: {  
 type: Object,  
 value: {  
 "backgroundColor": "#ffffff",  
 "color": "#979795",  
 "selectedColor": "#1c1c1b",  
 "list": [{  
 "pagePath": "pages/money/money",  
 "text": "明细",  
 "iconPath": "icon/detail.png",  
 "selectedIconPath": "icon/detail1.png"  
 },  
 {  
 "pagePath": "pages/addmoney/addmoney",  
 "text": "添加",  
 "isSpecial": true,  
 "iconPath": "icon/add1.png",  
 "selectedIconPath": "icon/add1.png"  
 },  
 {  
 "pagePath": "pages/figure/figure",  
 "text": "图表",  
 "iconPath": "icon/figure.png",  
 "selectedIconPath": "icon/figure1.png"  
 }  
 ]  
 }  
 }  
 },  
 data: {  
 },  
 methods: {  
 }  
})import Timing from './timing'  
export default function Animation (opts) {  
 opts.duration = typeof opts.duration === 'undefined' ? 1000 : opts.duration;  
 opts.timing = opts.timing || 'linear';  
 let delay = 17;  
 let createAnimationFrame = function () {  
 if (typeof requestAnimationFrame !== 'undefined') {  
 return requestAnimationFrame;  
 } else if (typeof setTimeout !== 'undefined') {  
 return function (step, delay) {  
 setTimeout(function () {  
 let timeStamp = +new Date();  
 step(timeStamp);  
 }, delay);  
 }  
 } else {  
 return function (step) {  
 step(null);  
 }  
 }  
 }  
 let animationFrame = createAnimationFrame();  
 let startTimeStamp = null;  
 let step = function (timestamp) {  
 if (timestamp === null) {  
 opts.onProcess && opts.onProcess(1);  
 opts.onAnimationFinish && opts.onAnimationFinish();  
 return;  
 }  
 if (startTimeStamp === null) {  
 startTimeStamp = timestamp;  
 }   
 if (timestamp - startTimeStamp < opts.duration) {  
 let process = (timestamp - startTimeStamp) / opts.duration;  
 opts.onProcess && opts.onProcess(process);  
 animationFrame(step, delay);  
 } else {  
 opts.onProcess && opts.onProcess(1);  
 opts.onAnimationFinish && opts.onAnimationFinish();  
 }  
 };  
 step = step.bind(this);  
 animationFrame(step, delay);  
}export default {  
 easeIn: function(pos){  
 return Math.pow(pos, 3);  
 },  
 easeOut: function(pos){  
 return (Math.pow((pos - 1), 3) + 1);  
 },  
 easeInOut: function(pos){  
 if ( (pos /= 0.5) < 1 ) {  
 return 0.5 \* Math.pow(pos, 3);  
 } else {   
 return 0.5 \* (Math.pow((pos - 2), 3) + 2);  
 }  
 },  
 linear: function(pos) {  
 return pos;  
 },  
}const formatTime = date => {  
 const year = date.getFullYear()  
 const month = date.getMonth() + 1  
 const day = date.getDate()  
 const hour = date.getHours()  
 const minute = date.getMinutes()  
 const second = date.getSeconds()  
 return [year, month, day].map(formatNumber).join('/') + ' ' + [hour, minute, second].map(formatNumber).join(':')  
}  
const formatYMD = (date, flag) => {  
 const year = date.getFullYear()  
 const month = date.getMonth() + 1  
 const day = date.getDate()  
 if (flag) return [year, month, day].map(formatNumber).join('-')  
 else return [year, month, day].map(formatNumber)  
}  
const formatYM = date => {  
 const year = date.getFullYear()  
 const month = date.getMonth() + 1  
 return [year, month].map(formatNumber)  
}  
const formatNumber = n => {  
 n = n.toString()  
 return n[1] ? n : '0' + n  
}  
function measureText (text, fontSize=50) {  
 text = String(text);  
 var text = text.split('');  
 var width = 0;  
 text.forEach(function(item) {  
 if (/[a-zA-Z]/.test(item)) {  
 width += 7;  
 } else if (/[0-9]/.test(item)) {  
 width += 5.5;  
 } else if (/\./.test(item)) {  
 width += 2.7;  
 } else if (/-/.test(item)) {  
 width += 3.25;  
 } else if (/[\u4e00-\u9fa5]/.test(item)) {  
 width += 10;  
 } else if (/\(|\)/.test(item)) {  
 width += 3.73;  
 } else if (/\s/.test(item)) {  
 width += 2.5;  
 } else if (/%/.test(item)) {  
 width += 8;  
 } else {  
 width += 10;  
 }  
 });  
 return width \* fontSize / 10;  
}  
module.exports = {  
 formatTime: formatTime,  
 formatYMD: formatYMD,  
 formatYM: formatYM,  
 measureText:measureText  
}

<!--index.wxml-->

<view class="container">

  <view class="userinfo">

    <button wx:if="{{!hasUserInfo && canIUse}}" open-type="getUserInfo" bindgetuserinfo="getUserInfo"> 获取头像昵称 </button>

    <block wx:else>

      <image bindtap="bindViewTap" class="userinfo-avatar" src="{{userInfo.avatarUrl}}" mode="cover"></image>

      <text class="userinfo-nickname">{{userInfo.nickName}}</text>

    </block>

  </view>

  <view class="block-all">

    <navigator class="block-inside" url="../todo/todo">

      <image src="./image/todo.png"></image>

      <text class="text-info">Todo</text>

    </navigator>

    <navigator class="block-inside" url="../money/money" open-type="switchTab">

      <image src="./image/jizhang.png"></image>

      <text class="text-info">记账</text>

    </navigator>

  </view>

  <view class="logo">

    <text class="logo-info">记录你的生活</text>

  </view>

</view>

/\*\*index.wxss\*\*/

page {

  height: 100%;

}

.container {

  height: 100%;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: space-between;

  /\* padding: 95rpx 0; \*/

  box-sizing: border-box;

}

.container::before {

  content: '';

  position: absolute;

  top: 0;

  left: 0;

  width: 100%;

  height: 100%;

  background-image: url(https://hbimg.huabanimg.com/a786fa0590bdbd216d8ec1f3f3a351e9b1515a516c2a8-mmcxHX\_fw658/format/webp);

  background-repeat: no-repeat;

  background-size: 100% 100%;

  /\* filter: blur(2px); \*/

  z-index: -1;

}

.userinfo {

  display: flex;

  flex-direction: column;

  align-items: center;

}

.userinfo-avatar {

  width: 128rpx;

  height: 128rpx;

  margin: 20rpx;

  border-radius: 50%;

}

.userinfo-nickname {

  color: rgb(97, 93, 93);

}

.usermotto {

  margin-top: 200px;

}

.block-all {

  margin-top: 40%;

  width: 85%;

  display: flex;

  justify-content: space-around;

}

.block-inside {

  display: flex;

  flex-direction: column;

  align-items: center;

}

.block-inside image {

  width: 50px;

  height: 50px;

}

.text-info {

  font-family: "shouxieti";

  color: dimgray;

  font-size: 50rpx;

}

.logo-info {

  font-family: "shouxieti";

  color: dimgray;

  font-size: 80rpx;

}

.page {

  height: 100%;

}

<!--logs.wxml-->

<view class="container log-list">

  <block wx:for="{{logs}}" wx:for-item="log">

    <text class="log-item">{{index + 1}}. {{log}}</text>

  </block>

</view>

.log-list {

  display: flex;

  flex-direction: column;

  padding: 40rpx;

}

.log-item {

  margin: 10rpx;

}

<view>

  <banner bind:reqMoney="reqMoney" income="{{income}}" expenditure="{{expenditure}}" type="1"></banner>

  <view wx:if="{{moneyList.length>0}}" class="money-container">

    <view wx:for="{{moneyList}}" wx:for-item="moneylist" wx:key="date">

      <money moneylist="{{moneylist}}"></money>

    </view>

  </view>

  <view wx:else class="money-no">

    <text>您还没有任何消费记录</text>

    <text>赶快去记录一笔吧~</text>

  </view>

  <tabbar tabbar="{{tabbar}}"></tabbar>

</view>

.money-no{

  color: #fe947e;

  display: flex;

  flex-direction: column;

  justify-content: center;

  height: 200px;

  align-items: center;

}

.money-container{

  margin-bottom: 11.5%;

}

<view>

  <view class="todo-container">

    <todocard title="紧急" todoList="{{p2List}}"></todocard>

    <todocard title="一般" todoList="{{p1List}}"></todocard>

    <todocard title="日常" todoList="{{p0List}}"></todocard>

    <view class="todo-add">

      <image wx:if="{{!add}}" src="./image/unadd.png" bindtap="changeImg"></image>

      <image src="./image/com.png" bindtap="changeCom"></image>

    </view>

  </view>

  <view wx:if="{{add}}" class="todo-shadow">

    <addcard isClose="{{add}}" class="add-style" bindaddClose="addClose" bindaddSuc="addSuc"></addcard>

  </view>

  <view wx:if="{{com}}" class="todo-shadow">

    <complete bindclose="comClose"></complete>

  </view>

</view>

.todo-container{

  min-height: 100vh;

  background-color: #f7f8fa;

}

.todo-add{

  margin-top: 20%;

  display: flex;

  justify-content: space-around;

  padding: 0 10%;

background-color: #ffff;

}

.todo-add image{

  width:48px;

  height: 48px;

}

.add-style{

  width:80%;

  border:1px solid whitesmoke;

  border-radius: 10px;

  background-color: white;

}

.todo-shadow{

  position: fixed;

  width: 100%;

  height: 100%;

  top: 0;

  background: rgba(0, 0, 0, 0.4);

  display: flex;

  justify-content: center;

  align-items: center;

  flex-direction: column;

}

.time-img {

  width: 20px;

  height: 20px;

  margin-right: 4px;

}

.record-detail {

  font-size: small;

  flex: 0 1 60%;

}

.picker {

  color: #76777b;

}

.del-icon {

  position: absolute;

  left:10px;

  width: 20px;

  height: 20px;

}

<!--pages/components/budget/budget.wxml-->

<view class="budget-container">

  <view class="budget-box">

    <view class="bugget-info">

      <text class="text-info">可根据需要调整通用额度</text>

      <text class="text-info text-info2">当前已用额度{{had}}元</text>

    </view>

    <view class="bugget-info">

      <!-- <input class="input-style" placeholder-class="placeholder-style" placeholder="500"></input> -->

      <input class="input-style" model:value="{{budget}}" bindfocus="changeShow"></input>

      <text class="text-info2">(调整以10为单位)</text>

    </view>

  </view>

  <button class="button-style {{btnblur?'btn-ano':''}}" bindtap="setSuc">确定</button>

</view>

/\* pages/components/budget/budget.wxss \*/

.budget-container {

  display: flex;

  flex-direction: column;

  align-items: center;

  height: 100vh;

  background-color: #f5f5f7;

}

.budget-box {

  border-radius: 8px;

  display: flex;

  flex-direction: column;

  align-items: center;

  width: 80%;

  background-color: #ffff;

  height: 22%;

  margin: 5% 0;

  padding: 5%;

  justify-content: space-between;

}

.bugget-info {

  display: flex;

  flex-direction: column;

  align-items: center;

}

.text-info {

  margin-bottom: 5%;

}

.text-info2 {

  color: rgba(44, 42, 42, 0.5);

}

.input-style {

  margin-bottom: 2%;

  border-bottom: 2rpx solid rgba(44, 42, 42, 0.1);

  padding-bottom: 8px;

  font-size: 30px;

  font-weight: 400;

  height: 2rem;

  text-align: center;

  width: 70%;

}

.placeholder-style {

  color: black;

  font-size: 30px;

  font-weight: 400;

  height: 2rem;

  text-align: center;

  cursor: ;

}

.button-style {

  background-color: #9bcaf4;

  color: #ffff;

}

.key-board {

  position: absolute;

  bottom: 0;

  width: 100%;

  height: 30%;

}

.btn-ano{

  background-color: #118af5;

}

<!--pages/money/figure/figure.wxml-->

<view>

  <banner bind:reqMoney="reqMoney" type="0" budget="{{budget}}" subbudget="{{subbudget}}" expval="{{exp\_val}}"></banner>

  <view class="figure-inside">

    <tabbar tabbar="{{tabbar}}"></tabbar>

    <ring id="ring\_id" ringList="{{ring\_list}}" total="{{choiceTotal}}" type="{{choiceType}}" bind:figureSwitch="figureSwitch"></ring>

    <figureitem itemList="{{choiceList}}" total="{{choiceTotal}}" type="{{choiceType}}" typeTitle="{{typeTitle}}"></figureitem>

  </view>

</view>

<!--pages/components/addcard/addcard.js.wxml-->

<view class="add-outline" wx:if="{{isClose}}">

  <view class="add-item add-container">

    <view class="del-image">

      <view class="del-out">

        <image src="./image/del.png" bindtap="changeDel"></image>

      </view>

    </view>

    <view class="add-inside">

      <view class="add-item">

        <input model:value="{{isDetail}}" placeholder="请输入你要做的事情"></input>

      </view>

      <view class="add-item item-tags">

        <image class="tag-image" src="./image/tag.png"></image>

        <view class="add-tag">

          <view class="tag {{isTag==2?'tag2':''}}" data-tags="2" bindtap="changeTag">紧急</view>

          <view class="tag {{isTag==1?'tag1':''}}" data-tags="1" bindtap="changeTag">一般</view>

          <view class="tag {{isTag==0?'tag0':''}}" data-tags="0" bindtap="changeTag">日常</view>

        </view>

      </view>

      <view class="add-radio">

        <button bindtap="addSubmit">提交</button>

      </view>

    </view>

  </view>

</view>

/\* pages/components/addcard/addcard.js.wxss \*/

.add-outline {

  padding: 20px;

}

.del-image{

  display: flex;

  flex-direction: row-reverse;

}

.del-out{

  width:40px;

}

.del-out image {

  width: 20px;

  height: 20px;

}

.add-item {

  margin-bottom: 10px;

}

.item-tags {

  display: flex;

  flex-direction: row;

}

.add-tag {

  width: 100%;

  display: flex;

  flex-direction: row;

  justify-content: space-around;

  align-items: center;

}

.add-radio {

  display: flex;

  justify-content: center;

}

.add-radio button{

  width: 120px;

  font-family: 'shouxieti';

  font-size: 30px;

  line-height: 1;

  font-weight: 350;

}

.tag-image {

  width: 20px;

  height: 20px;

}

.tag {

  padding: 4px;

  border-radius: 6px;

  color: white;

  font-size: 22px;

  background-color: #52616b;

  font-family: 'shouxieti';

  width: 20%;

  display: flex;

  justify-content: center;

}

.tag0 {

  background-color: teal;

}

.tag1 {

  background-color: #ffaa64;

}

.tag2 {

  background-color: tomato;

}

<!--pages/components/addicon/addicon.wxml-->

<view class="icon-out" wx:if="{{iconShow}}">

  <view class="icon-shadow">

  </view>

  <view class="shadow-inside">

    <view class="inside-top">

      <view class="item-title">请输入类别名称</view>

      <input model:value="{{newTag}}" placeholder="最长输入4个字"></input>

    </view>

    <view class="inside-bottom">

      <view class="bottom-btn" data-flag="0" bindtap="addSuc">取消</view>

      <view class="bottom-btn" data-flag="1" bindtap="addSuc">确定</view>

    </view>

  </view>

</view>

/\* pages/components/addicon/addicon.wxss \*/

.icon-out{

  display: flex;

  align-content: center;

  justify-content: center;

}

.icon-shadow{

  background-color: rgba(0,0,0,0.5);

  position: absolute;

  top:0;

  left:0;

  width: 100%;

  height: 100%;

  z-index: 100;

}

.shadow-inside{

  background-color: white;

  border: 1px solid white;

  border-radius: 5px;

  width: 55%;

  z-index: 200;

}

.inside-top{

  padding: 10px;

}

.inside-bottom{

  display: flex;

  flex-direction: row;

  justify-content: space-between;

  border-top: 1px solid  rgba(0,0,0,0.5);

}

.item-title{

  margin-bottom: 5px;

  color: #303841;

}

.bottom-btn{

  width: 100%;

  text-align: center;

  font-size: 14px;

  padding: 8px;

  color: #3a4750;

}

.bottom-btn:nth-child(1){

  border-right: 1px solid  rgba(0,0,0,0.5);

}

<!--pages/components/moneytop/moneytop.wxml-->

<wxs module="format" src="../../../utils/format.wxs"></wxs>

<view class="banner-top">

  <view>

    <text class="banner-title" wx:if="{{type}}">明细</text>

    <text class="banner-title" wx:else>图表</text>

  </view>

  <view class="banner-detail">

    <view class="banner-item">

      <picker mode="date" fields="month" bindchange="bindDateChange">

        <view>{{year}}</view>

        <view class="item-bold item-more">

          <text>{{month}}</text>

          <text class="text-small">月</text>

          <view class="triangle" bindtap="monthSwitch"></view>

        </view>

      </picker>

    </view>

    <view wx:if="{{type}}" class="banner-info">

      <view class="banner-item">

        <view>收入</view>

        <view>{{format.numberFixed(income)}}</view>

      </view>

      <view class="banner-item">

        <view>支出</view>

        <view>{{format.numberFixed(expenditure)}}</view>

      </view>

    </view>

    <view wx:else class="banner-info">

      <view class="banner-item" bindtap="setBudget">

        <view>预算</view>

        <view>{{format.numberFixed(budget)}}</view>

      </view>

      <view class="banner-item">

        <view>剩余预算</view>

        <view>{{format.numberFixed(subbudget)}}</view>

      </view>

    </view>

  </view>

</view>

/\* pages/components/moneytop/moneytop.wxss \*/

.banner-top {

  display: flex;

  flex-direction: column;

  padding: 10px;

  background-color: #fe947e;

  color: white;

}

.banner-title {

  display: flex;

  justify-content: center;

}

.banner-detail {

  display: flex;

  flex-direction: row;

}

.banner-item {

  display: flex;

  flex-direction: column;

  justify-content: space-between;

}

.item-bold {

  margin-top: 5px;

  font-size: 23px;

  font-weight: 530;

}

.text-small {

  font-weight: normal;

  font-size: 15px;

  margin: 0 4px;

}

.item-more {

  display: flex;

  flex-direction: row;

  align-items: baseline;

}

/\*实心三角\*/

.triangle {

  width: 0;

  height: 0;

  border-left: 7px solid transparent;

  border-right: 7px solid transparent;

  border-top: 8px solid #fef9fa;

}

.banner-info{

  display: flex;

  flex-direction: row;

  justify-content: space-between;

  flex: 1 1 auto;

  margin-left: 20%;

}

<!--pages/components/complete/complete.wxml-->

<view class="com-container">

  <view>

    <ring id="ring\_id" bind:close="close"></ring>

  </view>

</view>

/\* pages/components/complete/complete.wxss \*/

.com-container{

  border-radius: 5px;

  background-color: #ffff;

  display: flex;

  padding: 2%;

}

<!--pages/components/figureitem/figureitem.wxml-->

<wxs module="format" src="../../../utils/format.wxs"></wxs>

<view class="figure-all" wx:if="{{itemList.length!=0}}">

  <view class="figure-banner">

    <view class="banner-icon" wx:if="{{itemList.tag>=0}}">

      <image src="{{format.pathFixed(itemList.tag)}}"></image>

    </view>

    <view class="banner-name">

      <text>{{typeTitle}}</text>

      <text class="banner-percent">{{format.numberFixed(itemList.val / total \* 100)}}%</text>

    </view>

    <view class="banner-val">

      <view wx:if="{{type==0}}">-</view>

      <view wx:else>+</view>

      <view>{{itemList.val}}

      </view>

    </view>

  </view>

  <view class="figure-title">

    <text>{{typeTitle}}</text>

    <text wx:if="{{type==0}}">支出</text>

    <text wx:else>收入</text>

    <text>排行榜</text>

  </view>

  <view class="figure-list">

    <view class="figure-item" wx:for="{{itemList.details}}" wx:key="id">

      <view class="item-id {{index<3?'item-red':''}}">{{index+1}}</view>

      <view class="item-detail" wx:if="{{item.remark==''}}">{{item.name}}</view>

      <view class="item-detail" wx:else>{{item.remark}}</view>

      <view class="item-val">

        <view wx:if="{{item.flag==0}}">-</view>

        <view wx:else>+</view>

        <text>{{format.numberFixed(item.money)}}</text>

      </view>

    </view>

  </view>

</view>

/\* pages/components/figureitem/figureitem.wxss \*/

.figure-all {

  padding: 10px;

  margin-bottom: 8%;

}

.figure-banner {

  display: flex;

  flex-direction: row;

  align-items: center;

  border-bottom: 1px solid whitesmoke;

  padding: 2% 0;

  margin-bottom: 7%;

}

.figure-item {

  display: flex;

  flex-direction: row;

  border-bottom: 1px solid whitesmoke;

  padding: 3% 0;

}

.banner-icon image {

  width: 40px;

  height: 40px;

}

.banner-name {

  flex: 1 0 auto;

  color: #666666;

}

.banner-percent {

  margin-left: 2%;

}

.banner-val {

  display: flex;

  flex-direction: row;

  color: #999999;

}

.item-detail {

  margin-left: 5%;

  flex: 1 0 auto;

}

.item-val {

  display: flex;

  flex-direction: row;

  color: #999999;

}

.figure-title {

  color: #999999;

}

.item-id {

  color: #999999;

}

.item-red {

  color: #ff6600;

}

<!--pages/components/keyboard/keyboard.wxml-->

<view class="key-all">

  <view class="key-left">

    <view class="key-row">

      <view class="key-item" hover-class="active" hover-stay-time="100" data-key="1" bindtap="tapkey">1</view>

      <view class="key-item" hover-class="active" hover-stay-time="100" data-key="2" bindtap="tapkey">2</view>

      <view class="key-item" hover-class="active" hover-stay-time="100" data-key="3" bindtap="tapkey">3</view>

    </view>

    <view class="key-row">

      <view class="key-item" hover-class="active" hover-stay-time="100" data-key="4" bindtap="tapkey">4</view>

      <view class="key-item" hover-class="active" hover-stay-time="100" data-key="5" bindtap="tapkey">5</view>

      <view class="key-item" hover-class="active" hover-stay-time="100" data-key="6" bindtap="tapkey">6</view>

    </view>

    <view class="key-row">

      <view class="key-item" hover-class="active" hover-stay-time="100" data-key="7" bindtap="tapkey">7</view>

      <view class="key-item" hover-class="active" hover-stay-time="100" data-key="8" bindtap="tapkey">8</view>

      <view class="key-item" hover-class="active" hover-stay-time="100" data-key="9" bindtap="tapkey">9</view>

    </view>

    <view class="key-row">

      <view class="key-item" hover-class="active" hover-stay-time="100" data-key="." bindtap="tapkey">.</view>

      <view class="key-item key-item-ano" hover-class="active" hover-stay-time="100" data-key="0" bindtap="tapkey">0</view>

    </view>

  </view>

  <view class="key-right">

    <view class="key-col">

      <view class="key-item fun-item" hover-class="active" hover-stay-time="100" data-key="11" bindtap="delkey">退格</view>

      <view class="key-item fun-item-sub" hover-class="active" hover-stay-time="100" data-key="12" bindtap="subkey">确定</view>

    </view>

  </view>

</view>

/\* pages/components/keyboard/keyboard.wxss \*/

.key-all {

  display: flex;

  flex-direction: row;

  height: 100%;

}

.key-left{

  flex:0 0 75%;

  display: flex;

  flex-direction: column;

  justify-content: space-between;

}

.key-right{

  flex:0 0 25%;

}

.key-row{

  flex: 0 0 25%;

  display: flex;

  justify-content: space-between;

  /\* 不要缩小 \*/

}

.key-item {

  /\* 项目空间不足的话要缩小 \*/

  flex: 0 1 33.33%;

  display: flex;

  align-items: center;

  justify-content: center;

  background-color: #3d4454;

  color: white;

  border-right: 1px solid #343b4b;

  border-bottom: 1px solid #343b4b;

}

.active{

  /\* background-color: #343b4bda; \*/

  box-shadow:0px 0px 10px #bbbec4 inset;

}

.key-item-ano{

  flex: 0 1 66.67%;

}

.fun-item{

  flex: 0 1 50%;

}

.fun-item-sub {

  flex: 0 1 50%;

  background-color: #e85862;

}

.hover-item-ano{

  background-color: #e85862c5;

}

.key-col {

  display: flex;

  flex-direction: column;

  justify-content: space-between;

  height: 100%;

}

<!--pages/components/moneyicon/moneyicon.wxml-->

<wxs module="format" src="../../../utils/format.wxs"></wxs>

<view>

  <swiper indicator-dots="true" interval="2000" duration="1000" current="{{curIndex}}">

    <block wx:for="{{newData}}" wx:key="index">

      <swiper-item class="icon-board">

        <view wx:for="{{item}}" wx:key="path" wx:for-item="key" class="icon-tip {{select==key.id?'icon-ano':''}}" bindtap="iconSuc" data-index="{{key.id}}" data-name="{{key.name}}">

          <image class="icon-img" src="{{format.pathFixed(key.id)}}"></image>

          <text class="icon-name {{select==key.id?'text-ano':''}}">{{key.name}}</text>

        </view>

      </swiper-item>

    </block>

  </swiper>

  <addicon iconShow="{{iconShow}}" bind:addIcon="addIcon"></addicon>

</view>

/\* pages/components/moneyicon/moneyicon.wxss \*/

.icon-board {

  display: flex;

  flex-wrap: wrap;

}

.icon-img {

  width: 30px;

  height: 30px;

  margin-bottom: 3px;

}

.icon-tip {

  flex: 0 1 20%;

  display: flex;

  flex-direction: column;

  align-items: center;

  margin-bottom: 3px;

  height: 35%;

}

.icon-name {

  font-size: small;

  color: #76777b;

}

.text-ano {

  text-shadow: 0 0 10px #76777b, 0 0 20px #76777b, 0 0 30px #76777b, 0 0 40px #76777b;

}

.icon-ano {

  box-shadow: 0 0 8px #DDD;

}

<!--pages/components/moneyitem/moneyitem.wxml-->

<wxs module="format" src="../../../utils/format.wxs"></wxs>

<view>

  <view class="money-title">

    <view>

      <text>{{format.weekFixed(moneylist.date).dateStr}}</text>

      <text>{{format.weekFixed(moneylist.date).dayOfWeek}}</text>

    </view>

    <view class="title-right">

      <view class="title-tip">

        <text>支出:</text>

        <text wx:if="{{moneylist.expenditure>=0}}">{{format.numberFixed(moneylist.expenditure)}}</text>

      </view>

      <view>

        <text>收入:</text>

        <text wx:if="{{moneylist.income>=0}}">{{format.numberFixed(moneylist.income)}}</text>

      </view>

    </view>

  </view>

  <view class="money-item" wx:for="{{moneylist.details}}" wx:key="id">

    <view class="item-icon">

      <image src="{{format.pathFixed(item.tag)}}"></image>

    </view>

    <view class="item-name">

      <text>{{item.name}}</text>

    </view>

    <view class="item-val {{item.flag?'val-earn':''}}">

      <view wx:if="{{item.flag==0}}">-</view>

      <view>{{item.money}}

      </view>

    </view>

  </view>

</view>

/\* pages/components/moneyitem/moneyitem.wxss \*/

.money-title {

  color: #bfbfbf;

  display: flex;

  flex-direction: row;

  align-items: center;

  justify-content: space-between;

  border-bottom: 1px solid #e6e6e6;

  padding: 20px 10px 10px;

  font-size: small;

}

.money-item {

  display: flex;

  flex-direction: row;

  align-items: center;

  padding: 5px 8px;

  border-bottom: 1px solid whitesmoke;

}

.title-right{

  display: flex;

  flex-direction: row;

}

.title-tip{

  margin-right: 4px;

}

.item-icon {

  flex: 1 1 0;

}

.item-icon image {

  width: 40px;

  height: 40px;

}

.item-name {

  flex: 5 1 0;

}

.item-val {

  display: flex;

  flex-direction: row;

  color: red;

  flex: 1 1 0;

  justify-content: flex-end;

}

.val-earn {

  color: green;

}

<!--pages/components/moneyswitch/moneyswitch.wxml-->

<view class="switch-inside">

  <view wx:for="{{switchlist}}" wx:key="index" bindtap="switchSuc" data-tag="{{index}}" class="switch-item {{isTag==index?'item-ano':''}}">

    {{item}}

  </view>

</view>

/\* pages/components/moneyswitch/moneyswitch.wxss \*/

.switch-inside {

  display: flex;

  flex-direction: row;

  justify-content: space-between;

}

.switch-item {

  width: 100%;

  color: black;

  text-align: center;

  padding: 5px 0;

}

.item-ano {

  background-color: #3d4454;

  color: white;

}

.switch-item:nth-child(1) {

  border-top-left-radius: 6px;

  border-bottom-left-radius: 6px;

  border-left: 1px solid #3d4454;

}

.switch-item:nth-last-child(1) {

  border-top-right-radius: 6px;

  border-bottom-right-radius: 6px;

}

.switch-item {

  border-top: 1px solid #3d4454;

  border-bottom: 1px solid #3d4454;

  border-right: 1px solid #3d4454;

}

<!--pages/components/progress/progress.wxml-->

<view class="canvas-out">

  <canvas type="2d" style="height:{{height}}px;width:{{width}}px" id="myCanvas" bindtap="choicePart">

  </canvas>

  <view class="canvas-text" bindtap="switchItem">

    <view class="text-inside">

      <text wx:if="{{!type}}">总支出</text>

      <text wx:else>总收入</text>

      <text class="text-info">{{total}}元</text>

    </view>

  </view>

</view>

/\* pages/components/progress/progress.wxss \*/

.canvas-out{

  display: flex;

  justify-content: center;

  align-items: center;

}

.canvas-text{

  position: absolute;

}

.text-inside{

  display: flex;

  flex-direction: column;

  align-items: center;

  color: #acacac;

}

.text-info{

  font-size: 20px;

}

<!--pages/components/ring/ring.wxml-->

<view class="canvas-out" bindtap="close">

  <canvas type="2d" style="height:{{height}}px;width:{{width}}px" id="myCanvas">

  </canvas>

  <view class="canvas-text" bindtap="switchItem">

    <view class="text-inside">

      <text>用户完成度</text>

      <text class="text-info">{{com}}%</text>

    </view>

  </view>

</view>

/\* pages/components/ring/ring.wxss \*/

.canvas-out{

  display: flex;

  justify-content: center;

  align-items: center;

}

.canvas-text{

  position: absolute;

}

.text-inside{

  display: flex;

  flex-direction: column;

  align-items: center;

  color: #acacac;

}

# 服务器端代码 Java

package com.example.demo;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
@SpringBootApplication  
public class DemoApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(DemoApplication.class, args);  
 }  
}  
package com.example.demo.controller;  
import com.example.demo.Dao.BudDao;  
import com.example.demo.Dao.MoneyDao;  
import com.example.demo.Dao.TodoDao;  
import com.example.demo.Restful.RetCode;  
import com.example.demo.Restful.RetResult;  
import com.example.demo.entity.applets.Bud;  
import com.example.demo.entity.applets.Money;  
import com.example.demo.entity.applets.Todo;  
import com.example.demo.service.MoneyService;  
import org.springframework.format.annotation.DateTimeFormat;  
import org.springframework.web.bind.annotation.\*;  
import java.util.\*;  
@RestController  
@RequestMapping(value = "test")  
@CrossOrigin  
public class TestController {  
 private TodoDao todoDao;  
 private MoneyService moneyService;  
 private MoneyDao moneyDao;  
 private BudDao budDao;  
 public TestController(TodoDao todoDao, MoneyService moneyService, MoneyDao moneyDao, BudDao budDao){  
 this.todoDao=todoDao;  
 this.moneyService=moneyService;  
 this.moneyDao=moneyDao;  
 this.budDao=budDao;  
 }  
 @GetMapping(value = "getTodo")  
 public RetResult getTodo(@DateTimeFormat(pattern = "yyyy-MM-dd") Date time, @RequestParam(name="name") String name)  
 {  
 Map<String, Object> res = new HashMap<>();  
 res.put("p0List",todoDao.findByTimeAndTagAndName(time,0,name));  
 res.put("p1List",todoDao.findByTimeAndTagAndName(time,1,name));  
 res.put("p2List",todoDao.findByTimeAndTagAndName(time,2,name));  
 return new RetResult(RetCode.SUCCESS.getCode(),res);  
 }  
 @PostMapping(value = "addTodo")  
 public RetResult addTodo(@RequestBody Todo body){  
 Map<String, Object> res = new HashMap<>();  
 Todo todo=new Todo();  
 todo.setName(body.getName());  
 todo.setTime(body.getTime());  
 todo.setDetail(body.getDetail());  
 todo.setFlag(body.getFlag());  
 todo.setTag(body.getTag());  
 todoDao.save(todo);  
 res.put("addTodo","ok");  
 return new RetResult(RetCode.SUCCESS.getCode(),res);  
 }  
 @GetMapping(value = "getMoney")  
 public RetResult getMoney(@RequestParam(name="year") String year, @RequestParam(name="month") String month, @RequestParam(name="usr") String usr)  
 {  
 return new RetResult(RetCode.SUCCESS.getCode(),moneyService.getTotal(year, month, usr));  
 }  
 @PostMapping(value = "addMoney")  
 public RetResult addMoney(@RequestBody Money money){  
 Map<String,Object> res=new HashMap<>();  
 moneyDao.save(money);  
 res.put("addMoney","ok");  
 return new RetResult(RetCode.SUCCESS.getCode(),res);  
 }  
 @GetMapping(value = "getFigure")  
 public RetResult getFigure(@RequestParam(name="year") String year, @RequestParam(name="month") String month, @RequestParam(name="usr") String usr){  
 return new RetResult(RetCode.SUCCESS.getCode(),moneyService.getTotalFigure(year, month, usr));  
 }  
 @GetMapping(value = "getBud")  
 public RetResult getBud(@RequestParam(name="usr") String usr){  
 Map<String, Object> res = new HashMap<>();  
 res.put("usr",budDao.findByName(usr));  
 return new RetResult(RetCode.SUCCESS.getCode(),res);  
 }  
 @PostMapping(value = "postBud")  
 public RetResult postBud(@RequestParam(name="usr") String usr, @RequestParam(name="bud") Integer budget){  
 List<Bud> bud=budDao.findByName(usr);  
 Map<String,Object> res=new HashMap<>();  
 Bud bud1=bud.get(0);  
 bud1.setBud(budget);  
 budDao.save(bud1);  
 res.put("change","ok");  
 return new RetResult(RetCode.SUCCESS.getCode(),res);  
 }  
}  
package com.example.demo.Dao;  
import com.example.demo.entity.applets.Bud;  
import org.springframework.data.repository.CrudRepository;  
import java.util.List;  
public interface BudDao extends CrudRepository<Bud,Integer> {  
 List<Bud> findByName(String name);  
}  
package com.example.demo.Dao;  
import com.example.demo.entity.applets.Money;  
import org.springframework.data.repository.CrudRepository;  
import java.util.List;  
public interface MoneyDao extends CrudRepository<Money, Integer> {  
 List<Money> findByYearAndMonthAndUsrOrderByDayDescIdDesc(String year,  
 String month,  
 String usr);  
 List<Money> findByYearAndMonthAndUsrAndFlagOrderByTagAscIdDesc(String year,  
 String month,  
 String usr, int flag);  
}  
package com.example.demo.Dao;  
import com.example.demo.entity.applets.Todo;  
import org.springframework.data.repository.CrudRepository;  
import java.util.Date;  
import java.util.List;  
public interface TodoDao extends CrudRepository<Todo,Integer> {  
 List<Todo> findByTimeAndTagAndName(Date time, Integer tag, String name);  
}  
package com.example.demo.entity.applets;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
@Entity  
public class Bud {  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY) //也要加  
 private Integer id;  
 private String name;  
 private Integer bud;  
 public void setName(String name) {  
 this.name = name;  
 }  
 public String getName() {  
 return name;  
 }  
 public void setId(Integer id) {  
 this.id = id;  
 }  
 public Integer getId() {  
 return id;  
 }  
 public Integer getBud() {  
 return bud;  
 }  
 public void setBud(Integer bud) {  
 this.bud = bud;  
 }  
}  
package com.example.demo.entity.applets;  
import com.fasterxml.jackson.annotation.JsonFormat;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
import java.util.Date;  
@Entity  
public class Monall {  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)   
 private Integer id;  
 private String name;  
 @JsonFormat(pattern = "yyyy-MM",timezone = "GMT+08")  
 private Date time;  
 private Integer pay;  
 private Integer earn;  
 public Integer getId() {  
 return id;  
 }  
 public Date getTime() {  
 return time;  
 }  
 public void setId(Integer id) {  
 this.id = id;  
 }  
 public void setTime(Date time) {  
 this.time = time;  
 }  
 public String getName() {  
 return name;  
 }  
 public void setName(String name) {  
 this.name = name;  
 }  
 public Integer getEarn() {  
 return earn;  
 }  
 public Integer getPay() {  
 return pay;  
 }  
 public void setEarn(Integer earn) {  
 this.earn = earn;  
 }  
 public void setPay(Integer pay) {  
 this.pay = pay;  
 }  
}  
package com.example.demo.entity.applets;  
import com.fasterxml.jackson.annotation.JsonFormat;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
import java.util.Date;  
@Entity  
public class Money {  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY) //也要加  
 private Integer id;  
 private String name;  
 @JsonFormat(pattern = "yyyy-MM-dd",timezone = "GMT+08")  
 private Date time;  
 private float money;  
 private Integer flag;  
 private Integer tag;  
 private String remark;  
 private String usr;  
 private String year;  
 private String month;  
 private String day;  
 public String getDay() {  
 return day;  
 }  
 public void setDay(String day) {  
 this.day = day;  
 }  
 public String getMonth() {  
 return month;  
 }  
 public String getYear() {  
 return year;  
 }  
 public void setMonth(String month) {  
 this.month = month;  
 }  
 public void setYear(String year) {  
 this.year = year;  
 }  
 public String getUsr() {  
 return usr;  
 }  
 public void setUsr(String usr) {  
 this.usr = usr;  
 }  
 public void setName(String name) {  
 this.name = name;  
 }  
 public String getName() {  
 return name;  
 }  
 public void setTag(Integer tag) {  
 this.tag = tag;  
 }  
 public Integer getTag() {  
 return tag;  
 }  
 public void setTime(Date time) {  
 this.time = time;  
 }  
 public void setId(Integer id) {  
 this.id = id;  
 }  
 public void setFlag(Integer flag) {  
 this.flag = flag;  
 }  
 public Integer getId() {  
 return id;  
 }  
 public Integer getFlag() {  
 return flag;  
 }  
 public Date getTime() {  
 return time;  
 }  
 public float getMoney() {  
 return money;  
 }  
 public String getRemark() {  
 return remark;  
 }  
 public void setMoney(float money) {  
 this.money = money;  
 }  
 public void setRemark(String remark) {  
 this.remark = remark;  
 }  
}  
package com.example.demo.entity.applets;  
import com.fasterxml.jackson.annotation.JsonFormat;  
import java.util.Date;  
import java.util.List;  
public class NewMoney {  
 private float income;  
 private float expenditure;  
 @JsonFormat(pattern = "yyyy-MM-dd",timezone = "GMT+08")  
 private Date date;  
 private List<Money> details;  
 public float getExpenditure() {  
 return expenditure;  
 }  
 public float getIncome() {  
 return income;  
 }  
 public void setExpenditure(float expenditure) {  
 this.expenditure = expenditure;  
 }  
 public void setIncome(float income) {  
 this.income = income;  
 }  
 public void setDate(Date date) {  
 this.date = date;  
 }  
 public Date getDate() {  
 return date;  
 }  
 public List<Money> getDetails() {  
 return details;  
 }  
 public void setDetails(List<Money> details) {  
 this.details = details;  
 }  
}  
package com.example.demo.entity.applets;  
import java.util.List;  
public class TagMoney {  
 private int tag;  
 private float val;  
 private List<Money> details;  
 public float getVal() {  
 return val;  
 }  
 public void setVal(float val) {  
 this.val = val;  
 }  
 public void setDetails(List<Money> details) {  
 this.details = details;  
 }  
 public List<Money> getDetails() {  
 return details;  
 }  
 public void setTag(int tag) {  
 this.tag = tag;  
 }  
 public int getTag() {  
 return tag;  
 }  
}  
package com.example.demo.entity.applets;  
import com.fasterxml.jackson.annotation.JsonFormat;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
import java.util.Date;  
@Entity  
public class Todo {  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY) //也要加  
 private Integer id;  
 private String name;  
 private String detail;  
 private Integer flag;  
 private Integer tag;  
 @JsonFormat(pattern = "yyyy-MM-dd",timezone = "GMT+08")  
 private Date time;  
 public Date getTime() {  
 return time;  
 }  
 public String getName() {  
 return name;  
 }  
 public void setName(String name) {  
 this.name = name;  
 }  
 public Integer getFlag() {  
 return flag;  
 }  
 public String getDetail() {  
 return detail;  
 }  
 public Integer getTag() {  
 return tag;  
 }  
 public void setFlag(Integer flag) {  
 this.flag = flag;  
 }  
 public void setTime(Date time) {  
 this.time = time;  
 }  
 public void setTag(Integer tag) {  
 this.tag = tag;  
 }  
 public void setDetail(String detail) {  
 this.detail = detail;  
 }  
 public void setId(Integer id) {  
 this.id = id;  
 }  
 public Integer getId() {  
 return id;  
 }  
}  
package com.example.demo.entity.applets;  
import java.util.List;  
public class TotalFigure {  
 private float in\_money;  
 private float exp\_money;  
 List<TagMoney> incomeList;  
 List<TagMoney> expList;  
 public float getExp\_money() {  
 return exp\_money;  
 }  
 public float getIn\_money() {  
 return in\_money;  
 }  
 public void setExp\_money(float exp\_money) {  
 this.exp\_money = exp\_money;  
 }  
 public void setIn\_money(float in\_money) {  
 this.in\_money = in\_money;  
 }  
 public List<TagMoney> getExpList() {  
 return expList;  
 }  
 public List<TagMoney> getIncomeList() {  
 return incomeList;  
 }  
 public void setExpList(List<TagMoney> expList) {  
 this.expList = expList;  
 }  
 public void setIncomeList(List<TagMoney> incomeList) {  
 this.incomeList = incomeList;  
 }  
}  
package com.example.demo.entity.applets;  
import java.util.List;  
public class TotalMoney {  
 private float all\_income;  
 private float all\_exp;  
 List<NewMoney> newMoneyList;  
 public float getAll\_exp() {  
 return all\_exp;  
 }  
 public float getAll\_income() {  
 return all\_income;  
 }  
 public List<NewMoney> getNewMoneyList() {  
 return newMoneyList;  
 }  
 public void setAll\_exp(float all\_exp) {  
 this.all\_exp = all\_exp;  
 }  
 public void setAll\_income(float all\_income) {  
 this.all\_income = all\_income;  
 }  
 public void setNewMoneyList(List<NewMoney> newMoneyList) {  
 this.newMoneyList = newMoneyList;  
 }  
}  
package com.example.demo.Restful;  
public enum RetCode {  
 SUCCESS(200),  
 FAIL(400),  
 FALSE(404),  
 ERROR(500);  
 private int code;  
 private RetCode(Integer code){  
 this.code=code;  
 }  
 public int getCode(){  
 return code;  
 }  
 public void setCode(Integer code){  
 this.code=code;  
 }  
}  
package com.example.demo.Restful;  
public class RetResult {  
 private Integer code;  
 private String msg;  
 private Object data;  
 public RetResult(){  
 this.code=Integer.valueOf(0);  
 this.msg="";  
 this.data=null;  
 }  
 public RetResult(Integer code,String msg,Object data){  
 this.code=code;  
 this.msg="";  
 this.data=null;  
 }  
 public RetResult(Integer code,String msg){  
 this.code=code;  
 this.msg=msg;  
 this.data=null;  
 }  
 public RetResult(Integer code,Object data){  
 this.code=code;  
 this.msg="";  
 this.data=data;  
 }  
 public Integer getCode() {  
 return code;  
 }  
 public void setCode(Integer code) {  
 this.code = code;  
 }  
 public String getMsg() {  
 return msg;  
 }  
 public void setMsg(String msg) {  
 this.msg = msg;  
 }  
 public void setData(Object data) {  
 this.data = data;  
 }  
 public Object getData() {  
 return data;  
 }  
}  
package com.example.demo.service;  
import com.example.demo.entity.applets.NewMoney;  
import com.example.demo.entity.applets.TagMoney;  
import com.example.demo.entity.applets.TotalFigure;  
import com.example.demo.entity.applets.TotalMoney;  
import java.util.List;  
public interface MoneyService {  
 List<NewMoney> getAll(String year, String month, String usr);  
 TotalMoney getTotal(String year, String month, String usr);  
 List<TagMoney> getFigure(String year, String month, String usr, int flag);  
 TotalFigure getTotalFigure(String year, String month, String usr);  
}  
package com.example.demo.service.Impl;  
import com.example.demo.Dao.MoneyDao;  
import com.example.demo.entity.applets.\*;  
import com.example.demo.service.MoneyService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import java.util.ArrayList;  
import java.util.List;  
@Service  
public class MoneyServiceImpl implements MoneyService {  
 @Autowired  
 private MoneyDao moneyDao;  
 @Override  
 public List<NewMoney> getAll(String year, String month, String usr) {  
 List<Money> monies;  
 monies = moneyDao.findByYearAndMonthAndUsrOrderByDayDescIdDesc(year, month, usr);  
 List<NewMoney> moneyList = new ArrayList<>();  
 if(monies.isEmpty()) return moneyList;  
 Money mm=monies.get(0);  
 String day\_start = mm.getDay();  
 NewMoney newMoney = new NewMoney();  
 newMoney.setDate(mm.getTime());  
 List<Money> monies1 = new ArrayList<>();  
 monies1.add(mm);  
 newMoney.setDetails(monies1);  
 if(mm.getFlag()==0){  
 newMoney.setExpenditure(mm.getMoney());  
 newMoney.setIncome(0);  
 }else{  
 newMoney.setExpenditure(0);  
 newMoney.setIncome(mm.getMoney());  
 }  
 moneyList.add(newMoney);  
 for (int i = 1; i < monies.size(); i++) {  
 Money money = monies.get(i);  
 String day = money.getDay();  
 if (day.equals(day\_start)) {  
 NewMoney nn=moneyList.get(moneyList.size() - 1);  
 float in=nn.getIncome();  
 float out=nn.getExpenditure();  
 if(money.getFlag()==0){  
 nn.setExpenditure(money.getMoney()+out);  
 }else{  
 nn.setIncome(money.getMoney()+in);  
 }  
 nn.getDetails().add(money);  
 } else {  
 day\_start = day;  
 NewMoney newMoneys = new NewMoney();  
 newMoneys.setDate(money.getTime());  
 List<Money> monies\_ = new ArrayList<>();  
 monies\_.add(money);  
 newMoneys.setDetails(monies\_);  
 if(money.getFlag()==0){  
 newMoneys.setExpenditure(money.getMoney());  
 newMoneys.setIncome(0);  
 }else{  
 newMoneys.setExpenditure(0);  
 newMoneys.setIncome(money.getMoney());  
 }  
 moneyList.add(newMoneys);  
 }  
 }  
 return moneyList;  
 }  
 @Override  
 public TotalMoney getTotal(String year, String month, String usr) {  
 float all\_income=0;  
 float all\_exp=0;  
 TotalMoney totalMoney=new TotalMoney();  
 List<NewMoney> moneyList = new ArrayList<>();  
 moneyList=getAll(year,month,usr);  
 for(int i=0;i<moneyList.size();i++){  
 float in=moneyList.get(i).getIncome();  
 float out=moneyList.get(i).getExpenditure();  
 all\_income+=in;  
 all\_exp+=out;  
 }  
 totalMoney.setAll\_exp(all\_exp);  
 totalMoney.setAll\_income(all\_income);  
 totalMoney.setNewMoneyList(moneyList);  
 return totalMoney;  
 }  
 @Override  
 public List<TagMoney> getFigure(String year, String month, String usr, int flag) {  
 List<Money> monies;  
 monies = moneyDao.findByYearAndMonthAndUsrAndFlagOrderByTagAscIdDesc(year, month, usr,flag);  
 List<TagMoney> moneyList = new ArrayList<>();  
 if(monies.isEmpty()) return moneyList;  
 Money mm=monies.get(0);  
 int tag\_start=mm.getTag();  
 TagMoney tagMoney=new TagMoney();  
 tagMoney.setTag(tag\_start);  
 tagMoney.setVal(mm.getMoney());  
 List<Money> monies1 = new ArrayList<>();  
 monies1.add(mm);  
 tagMoney.setDetails(monies1);  
 moneyList.add(tagMoney);  
 for (int i = 1; i < monies.size(); i++) {  
 Money money = monies.get(i);  
 int tag=money.getTag();  
 if (tag==tag\_start) {  
 TagMoney nn=moneyList.get(moneyList.size() - 1);  
 nn.setVal(nn.getVal()+money.getMoney());  
 nn.getDetails().add(money);  
 } else {  
 tag\_start = tag;  
 TagMoney tagMoney1=new TagMoney();  
 List<Money> monies\_ = new ArrayList<>();  
 monies\_.add(money);  
 tagMoney1.setTag(tag\_start);  
 tagMoney1.setDetails(monies\_);  
 tagMoney1.setVal(money.getMoney());  
 moneyList.add(tagMoney1);  
 }  
 }  
 return moneyList;  
 }  
 @Override  
 public TotalFigure getTotalFigure(String year, String month, String usr) {  
 float in=0;  
 float out=0;  
 TotalFigure totalFigure=new TotalFigure();  
 List<TagMoney> tagMoneyList1=getFigure(year, month, usr, 1);  
 for(int i=0;i<tagMoneyList1.size();i++){  
 in+=tagMoneyList1.get(i).getVal();  
 }  
 List<TagMoney> tagMoneyList2=getFigure(year, month, usr, 0);  
 for(int i=0;i<tagMoneyList2.size();i++){  
 out+=tagMoneyList2.get(i).getVal();  
 }  
 totalFigure.setExp\_money(out);  
 totalFigure.setIn\_money(in);  
 totalFigure.setExpList(tagMoneyList2);  
 totalFigure.setIncomeList(tagMoneyList1);  
 return totalFigure;  
 }  
}