## index.js

//index.js

var Bmob = require('../../utils/bmob.js')

var util = require('../../utils/util.js')

var common = require('../../utils/common.js')

var that

//校区选择

Page({

data: {

campus: [ "南校区", "北校区"],

campusIndex: 3,

limit: 2,

skip: 0,

postList: []

},

//图书信息显示

onLoad: function () {

that = this;

var Post = Bmob.Object.extend("post");

var query = new Bmob.Query(Post);

query.equalTo("campus", this.data.campus[this.data.campusIndex]);

query.descending('updatedAt');

query.limit(this.data.limit);

query.find({

success: function (results) {

that.setData({

postList: results,

skip: results.length

})

},

error: function (error) {

console.log("onLoad查询post失败: " + error.code + " " + error.message);

}

})

},

onPullDownRefresh: function () {

var Post = Bmob.Object.extend("post");

var query = new Bmob.Query(Post);

query.equalTo("campus", that.data.campus[this.data.campusIndex]);

query.descending('updatedAt');

query.limit(this.data.limit);

query.find({

success: function (results) {

that.setData({

postList: results,

skip: results.length

})

},

error: function (error) {

console.log("onLoad查询post失败: " + error.code + " " + error.message);

}

})

},

//图书加载是否完成

onReachBottom: function () {

var Post = Bmob.Object.extend("post");

var query = new Bmob.Query(Post);

query.equalTo("campus", that.data.campus[this.data.campusIndex]);

query.descending('updatedAt');

query.skip(this.data.skip);

query.limit(this.data.limit);

query.find({

success: function (results) {

if (results.length > 0) {

var nl = that.data.postList.concat(results);

that.setData({

skip: that.data.skip + results.length,

postList: nl

})

}

else {

wx.showToast({

title: '已全部加载',

icon: 'success',

duration: 3000

})

}

wx.stopPullDownRefresh()

},

error: function (error) {

console.log("onReachBottom查询post失败: " + error.code + " " + error.message);

}

})

},

//校区选择按钮

bindCampusChange: function (e) {

this.setData({

campusIndex: e.detail.value

})

},

//搜索按钮显示

showInput: function () {

this.setData({

inputShowed: true

});

},

//搜索输入栏

hideInput: function () {

this.setData({

inputVal: "",

inputShowed: false

});

var Post = Bmob.Object.extend("post");

var query = new Bmob.Query(Post);

query.equalTo("campus", this.data.campus[this.data.campusIndex]);

query.descending('updatedAt');

query.limit(this.data.limit);

query.find({

success: function (results) {

that.setData({

postList: results,

skip: results.length

})

},

error: function (error) {

console.log("hideInput查询post失败: " + error.code + " " + error.message);

}

})

},

//刷新

clearInput: function () {

this.setData({

inputVal: ""

});

var Post = Bmob.Object.extend("post");

var query = new Bmob.Query(Post);

query.equalTo("campus", this.data.campus[this.data.campusIndex]);

query.descending('updatedAt');

query.limit(this.data.limit);

query.find({

success: function (results) {

that.setData({

postList: results,

skip: results.length

})

},

error: function (error) {

console.log("clearInput查询post失败: " + error.code + " " + error.message);

}

})

},

//图书搜索按钮

bindSearch: function (e) {

this.getList(e.detail.value);

console.log(e.detail.value)

this.setData({

inputVal: e.detail.value

});

},

//图书名称匹配

getList: function (k) {

var Post = Bmob.Object.extend("post");

var queryBookName = new Bmob.Query(Post);

var queryISBN = new Bmob.Query(Post);

var queryCourseName = new Bmob.Query(Post);

queryBookName.equalTo("bookName", k);

queryISBN.equalTo("bookISBN",k)

queryCourseName.equalTo("courseName", k)

var mainQuery = Bmob.Query.or(queryBookName, queryISBN,queryCourseName);

mainQuery.equalTo("campus", that.data.campus[this.data.campusIndex]);

mainQuery.descending('updatedAt');

//mainQuery.limit(this.data.limit);

mainQuery.find({

success: function (results) {

console.log(results)

that.setData({

postList: results

})

},

error: function (error) {

console.log("getList查询pos失败: " + error.code + " " + error.message);

}

});

}

})

## 评价：

**代码格式和命名规范符合要求，规模适中，业务逻辑比较清晰，注释可以再多写一点，增强可读性，尽量避免重复的代码，提高程序的可维护性和执行效率。**

## post.js

// post.js

var Bmob = require('../../utils/bmob.js');

var basicURL = 'http://localhost/v2/book/isbn/:';

var app = getApp()

Page({

/\*\*

\* 页面的初始数据

\*/

data: {

ownerId: '',

ownerNickname: '',

ownerGender: '',

bookFound: false,

bookISBN: '',

bookImg: '',

bookName: '',

bookAuthor: '',

bookPress: '',

bookPrice: '',

isTextbook: false,

courseName: '',

conditions: ["全新","几乎全新", "少量笔记", "较多笔记", "不影响阅读"],

conditionIndex: 2,

campus: ["浙江大学城市学院南校区","浙江大学城市学院北校区", "浙江大学工程师学院",],

campusIndex: 3,

currentPrice: '',

postRemark: '',

buttonLoading: false

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function () {

var that = this

app.getUserInfo(function (userInfo) {

that.setData({

ownerNickname: userInfo.nickName,

ownerGender: userInfo.gender

}),

console.log('ownerNickname:', that.data.ownerNickname)

console.log('ownerGender:', that.data.ownerGender)

}),

this.setData({

ownerId: Bmob.User.current().id

})

console.log('ownerId:', this.data.ownerId)

},

/\*\*

\* 生命周期函数--监听页面初次渲染完成

\*/

onReady: function () {

},

/\*\*

\* 生命周期函数--监听页面显示

\*/

onShow: function () {

},

/\*\*

\* 生命周期函数--监听页面隐藏

\*/

onHide: function () {

},

/\*\*

\* 生命周期函数--监听页面卸载

\*/

onUnload: function () {

},

/\*\*

\* 页面相关事件处理函数--监听用户下拉动作

\*/

onPullDownRefresh: function () {

},

/\*\*

\* 页面上拉触底事件的处理函数

\*/

onReachBottom: function () {

},

//响应事件

//扫码

scanBookISBN: function () {

var that = this;

wx.scanCode({

success: (res) => {

that.setData({

bookISBN: res.result

})

that.searchBookISBN()

}

})

},

//输入

inputBookISBN: function (e) {

var that = this;

if (e.detail.value != '') {

that.setData({

bookISBN: e.detail.value

})

that.searchBookISBN();

}

},

searchBookISBN: function () {

var that = this;

// req(this.data.bookISBN);

wx.request({

url: basicURL + that.data.bookISBN,

// url: basicURL + this.data.bookISBN,

// data: data,

method: 'post',

header: { 'Content-Type': 'json' },

success: function (res) {

if (res.statusCode != '404') {

console.log(res)

that.setData({

bookFound: true,

bookName: res.data.title,

bookAuthor: res.data.author[0],

bookPress: res.data.publisher,

bookPrice: res.data.price,

bookImg: res.data.image,

})

}

else {

var book = Bmob.Object.extend("book");

var query = new Bmob.Query(book);

query.equalTo("bookISBN", that.data.bookISBN);

// 查询所有数据

query.find({

success: function (results) {

console.log('查找book表成功')

if (results.length != 0) {

console.log('book表中有该书')

var object = results[0];

that.setData({

bookFound: true,

bookName: res.data.title,

bookAuthor: res.data.author[0],

bookPress: res.data.publisher,

bookPrice: res.data.price,

bookImg: res.data.image,

})

}

else {

console.log('book表中无该书')

wx.showToast({

title: '抱歉未找到该书',

icon: 'success',

duration: 3000

})

}

},

error: function (error) {

console.log("查询book表失败: " + error.code + " " + error.message);

}

});

}

}

})

},

bindBookNameInput: function (e) {

this.setData({

bookName: e.detail.value

})

},

bindBookAuthorInput: function (e) {

this.setData({

bookAuthor: e.detail.value

})

},

bindBookPressInput: function (e) {

this.setData({

bookPress: e.detail.value

})

},

bindBookPriceInput: function (e) {

this.setData({

bookPrice: e.detail.value

})

},

bindNeedCourse: function (e) {

this.setData({

isTextbook: e.detail.value

})

if (!this.data.isTextbook) {

this.setData({

courseName: ''

})

}

},

bindCourseInput: function (e) {

this.setData({

courseName: e.detail.value

})

},

bindConditionChange: function (e) {

this.setData({

conditionIndex: e.detail.value

})

},

bindCampusChange: function (e) {

this.setData({

campusIndex: e.detail.value

})

},

bindCurrentPriceInput: function (e) {

this.setData({

currentPrice: e.detail.value

})

},

bindPostRemarkInput: function (e) {

this.setData({

postRemark: e.detail.value

})

},

bindSubmit: function () {

var that = this;

this.setData({

buttonLoading: true

})

var Post = Bmob.Object.extend("post");

var post = new Post();

post.set("ownerId", this.data.ownerId);

post.set("ownerNickname", this.data.ownerNickname);

post.set("ownerGender", this.data.ownerGender);

post.set("bookISBN", this.data.bookISBN);

post.set("bookImg", this.data.bookImg);

post.set("bookName", this.data.bookName);

post.set("bookAuthor", this.data.bookAuthor);

post.set("bookPress", this.data.bookPress);

post.set("bookPrice", this.data.bookPrice);

post.set("isTextbook", this.data.isTextbook);

post.set("courseName", this.data.courseName);

post.set("condition", this.data.conditions[this.data.conditionIndex]);

post.set("campus", this.data.campus[this.data.campusIndex]);

post.set("currentPrice", parseInt(this.data.currentPrice));

post.set("postRemark", this.data.postRemark);

//添加数据，第一个入口参数是null

post.save(null, {

success: function (result) {

// 添加成功，返回成功之后的objectId（注意：返回的属性名字是id，不是objectId），你还可以在Bmob的Web管理后台看到对应的数据

console.log("创建post成功, objectId:" + result.id);

that.setData({

buttonLoading: false

});

wx.showToast({

title: '发布成功',

icon: 'success',

duration: 3000

})

},

error: function (result, error) {

// 添加失败

console.log('创建post失败',result,error);

that.setData({

buttonLoading: false

})

}

});

}

})

## 评价：

**代码格式和命名规范符合要求，规模适中，逻辑比较清晰，可以更加简洁，多一些注释，同时也要注意注释的质量，在代码可读性合格的基础上，注释应该提供比代码更多的信息，可以适当优化，提高执行效率，降低复杂度。**