数据库操作基本可以看文档解决，（主要是按加号新增数据等操作要了解）

连接数据库（js中的最外层，page外）

const db = wx.cloud.database()

指定数据库（在js，方法中）

db.collection("数据库名字")

查询

.get()方法就是查询

db.collection("demolist").get({

      success : res=>{

        console.log(res)

      }

    })

若在.get方法前加入一个.doc(“\_id”)，则可以筛选数据（只能放id）

db.collection("demolist").get().then(res=>{

      console.log(res)

      })

这种方法也是可以的，这里then就是相当于success的用法。返回结果完全一致。

如果有调用嵌套，继续在后面加then即可。

    db.collection("demolist").get().then(res=>{

      console.log(res)

      }).catch(err=>{

      })

但通常我们还会加入catch来捕获错误。

若知道其他数据内容（如知道名字，但不知道id），就可以用where

    db.collection("demolist").where({

      author:"作者"

    }).get()

    .then(res=>{

      console.log(res)

    })

（大于小于之类的操作，等sdk再详细讲解）

添加

.add()添加数据

    db.collection("demolist").add({

      data:{

        title : "测试标题2",

        author : "李四",

        content : "测试内容2"

      }

    }).then(res=>{

      console.log(res)

    })

这里的res（then）可以不写，不过最好写一下，查询有没有错误

添加记录的时候，会自动写入openid（每一个人的openid是一样的）

    wx.showLoading({

      title: '数据加载中',

      mask:true

    })

    //这一步用于放止用户一直点击

    db.collection("demolist").add({

      data:{

        title : "测试标题2",

        author : "李四",

        content : "测试内容2"

      }

    }).then(res=>{

      console.log(res)

      wx.hideLoading()

    })

在最前面加一个showloading和后面用hideloading，放止用户一直点击。

表单提交（用输入框提交到数据库里）

  btnSub(res){

    var title = res.detail.value.title;

    var author = res.detail.value.author;

    var content = res.detail.value.content;

    db.collection("demolist").add({

      data : {

        title : title,

        author :author,

        content : content

      }

    }).then(res=>{

      console.log(res)

    })

  }

安全写法。

var resVlu = res.detail.value;

    db.collection("demolist").add({

      data:resVlu

    }).then(res=>{

      console.log(res)

    })

简洁写法。

data:resVlu

就可以自动的做类似于上面content:content的操作。

系统会自动写入openid，我们可以通过用户的openid来查询数据

更新数据

    db.collection("demolist").doc("f2a60d815edf897a00727b77257d4935").update({

      data : {

        author:"作者"

      }

    }).then(res=>{

      console.log(res)

    })

这里也可以用where，不用doc。Doc就只能改一条数据，where可以改多个。

（where要用云函数）

更新操作不仅可以修改数据，还能增加数据，

如果用的不是update而是用set方法，就会把其他字段覆盖。比如只写了author，其他数据都会消失。

云开发p9

C:\Users\asus\WeChatProjects\miniprogram-1