评论分享页面

* 建立commentShare组件，写好布局
* 引入login组件，通过loginshow拿到modalshow属性，默认是false。给评论区域绑定事件，在此事件中，判断用户是否已授权，如果授权了，那就拿到用户的信息。如果没有授权，把loginshow设为true
* 拿到从login组件中抛出的loginsuccess和loginfail方法，并在方法内做相应处理
* 再从bottom-modal中拿到modalshow属性，登录成功后显示
* 再在bottom-modal标签中写个评论的插槽

云数据库一对多

* 数据量较大的时候(评论上万条)，最好在n的一方（评论）存储1的一方的信息（这个blog的id）

评论功能

* 给input绑定事件拿到评论输入的内容
* 给发送绑定插入评论内容到数据库的事件。 如果内容为空弹窗提示，之后loading显示正在评论中。并传入输入内容，发布时间，blogid, nickname和头像
* 在成功的方法中给userInfo也赋值为event.detail
* 评论成功后，关闭底部评论区域，并清空评论内容

！！\*模版消息推送

* 把slot变成form, 去除oninput事件绑定，report-submit设为true， 把onSubmit事件绑定到form上
* 在onsubmit中通过event拿到content和formid（formid作为参数传入模版推送的云函数中，有效期为7天，每次提交一个表单会生成一个formid）
* 去到微信公众平台，进入模版消息，找到评价完成通知，在这里选评价结果和评价内容。提及后可拿到模版id
* 创建一个云函数sendMessage, 在里面创建一个config.json文件，固定写入：

{

“permissions”:{

“openapi”:[

“templateMessage.send”

]

}

}

* 云函数js文件中，通过cloud的getWXContext()拿到{OPENID}
* 之后cloud里面一直拿到templatemessage.send这里，传入tourser为OPENID, 然后page为当前这个blogid的人评论的url， data里是模版中写好的传递的两个参数，传入模版id， 再传入formid。此推送消息为异步，把它赋予一个值并return
* 回到前端，在插入评论数据之后调用推送模版，调用云函数，传入评论内容，formid以及blogid

博客详情（云数据库多集合查询）

* 点击博客，获取当前博客的博客blogid，再通过blogid去博客集合里面查询。之后返回博客的内容以及评论列表
* 进行详情查询和评论查询

详情查询:

在blog集合中查询blogid是否等同于\_id

* 评论查询:

1. 云函数端每次最多可查询100条，所以要分次查询
2. 在blog-comment集合里查，拿到获取的总次数，遍历次数
3. 建个commentlist对象里面data是个空数组
4. 用skip和limit方法以及当前blogid是否等同于集合里的blogid过滤选项，再由倒叙创建时间排列。整个结果是一个promise。最后把这个promise推进数组
5. 执行完所有promise以后执行: 用一个累加器把每次的结果累加到当前data上面，并把它赋值给commentList
6. 之后来到blog-comment前端，调用云函数
7. 前端拿到blog 和comment
8. 把commentlist里的时间通过之前写的format方法转换格式
9. 前端样式渲染
10. 评论发送成功后，在blog-ctrl组件中抛出一个事件，再在blog-comment中接受并定义，让这个事件为重新获取整个评论的数据

分享博客

* 在commentShare的wxml中把分享用button包起来， 加入open-type=”share”, 再把blogid传参过去
* 在find page中把整个blog传给blog-ctrl组件
* 修改分享的基础按钮的基础样式
* 分别到blog和blog-comment中的onShareAppMessage()生命周期函数中，拿到当前的blog, 并返回一个包含title和点开始之后跳到的路径(要传入blogId)