

# 云开发公开课

一天打造论坛小程序



# 目录

Part1:快速入门小程序	错误!未定义书签。
1.1 前置知识	错误!未定义书签。
1.1.1 技术栈	
1.1.2 微信开发文档	
1.2 准备工作	错误!未定义书签。
1.2.1 注册小程序	错误!未定义书签。
1.2.2 下载 IDE	
Part 2:云开发入门实战	错误!未定义书签。
2.1 初探云开发	
2.1.1 云开发简介	
2.1.2 云开发能力	错误!未定义书签。
2.1.3 云开发优势	
2.2 Hello World	错误!未定义书签。
2.2.1 初始化云开发小程序	错误!未定义书签。
2.2.2 开通小程序·云开发	错误!未定义书签。
2.2.3 熟悉云开发控制台	错误!未定义书签。
2.3 云数据库	错误!未定义书签。
2.3.1 插入数据	错误!未定义书签。
2.3.2 删除数据	错误!未定义书签。
2.3.3 查询数据	
2.3.4 更新数据	
2.3.5 索引管理	
2.3.6 权限管理	错误!未定义书签。
2.4 云存储	错误!未定义书签。
2.4.1 上传文件	
2.4.2 获取临时链接	
2.4.3 权限管理	
2.5 云函数	

		2.5.1 创建、安装依赖与部署	错误!未定义书签。
		2.5.2 调用云函数	错误!未定义书签。
		2.5.3 获取用户登录态	错误!未定义书签。
		2.5.4 云调用	错误!未定义书签。
	2.6	更多云开发能力	错误!未定义书签。
Part	t 3 :	云开发答疑环节	错误!未定义书签。
Part	t <b>4</b> :	云开发项目实战	3
	4.1	项目概述	3
	4.2	任务目标	3
	4.3	准备工作	3
	4.4	实战架构	3
	4.5	实战任务	4
		4.5.1 创建小程序·云开发环境	4
		4.5.2 初始化云开发能力	6
		4.5.3 开发发帖功能	6
		4.5.4 开发看帖功能	8
		4.5.5 开发评论功能	10
	4.6	完成标准	11
	4.7	附加任务	11

# Part 1:云开发项目实战

# 4.1 项目概述

社区,是所有应用开发里最常见的场景之一。

本次「云开发」公开课,将通过实战"论坛小程序"开发,教你如何借助小程序·云开发能力,提升功能开发效率。

# 4.2 任务目标

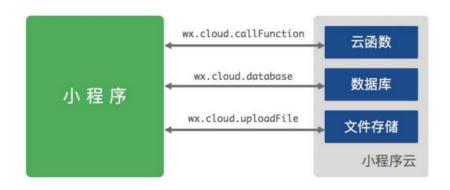
根据项目初始框架,完善代码令发帖、看帖、评论等功能能够正常使用,完成论坛小程序的开发,并上传设为体验版。并将助教微信号添加为体验者以便助教同学体验。

# 4.3 准备工作

在 PC 中下载论坛小程序项目脚手架,下载地址:

http://url.yellowsun.cn/hole

# 4.4 实战架构



# 4.5 实战任务

\*有能力的同学推荐尝试自行编写代码,不要偷看参考答案噢!

### 4.5.1 创建小程序•云开发环境

1、打开微信 IDE,创建一个新的小程序项目,并选择导入树洞 Demo 的目录,AppID 修改为你已经注册好的小程序 AppID。



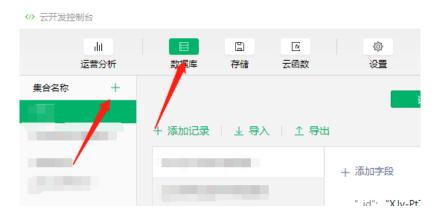
2、打开项目后,点击顶部工具栏上的"云开发"



3、自定义填入环境名称和环境 ID,点击"确定"创建云开发环境,即可进入云开发控制台。



4、树洞小程序后续需要使用云开发提供的云数据库能力,数据库基于 MongoDB,需要先创建一个集合。进入云开发控制台,找到"数据库",点击左侧边栏的"+"按钮创建集合。



5、我们需要创建两个集合,分别命名为"post\_collection"和"comment\_collection",用于存储帖子数据和评论数据。



### 4.5.2 初始化云开发能力

打开 miniprogram 目录下的 app.js 文件 在 onLaunch 事件最后初始化云开发能力,env 参数为上一任务中创建的环境 ID, traceUser 参数值为 true,并保存。

```
+
     Q
                              •=
                                    ₽←
                                                       ×
                                           app.js
                                            1
                                                 //app.js
▶ 🗟 cloudfunctions | demo
                                                 const util = require('./utils/util.js');
                                            2
▼ Imminiprogram
                                            3
                                                 App({
                                            4
                                                   onLaunch: function () {
 ▶ ☐ images
                                                     var that = this
                                            5
 ▶ □ pages
                                            6
                                                     wx.clearStorage()
 JS util.js
                                            8
                                                     //初始化云开发能力
                                            9
                                                     wx.cloud.init({
   JS app.js
                                                       env: '环境ID',
                                           10
   () app.json
                                           11
                                                       traceUser: true
   wxss app.wxss
                                           12
                                           13
   () min.config.json
                                           14
   () sitemap.json
                                                   globalData: {
                                           15
 gitignore .
                                                     userInfo: "StorageUserInfo",
                                           16
 README.md
                                                     wechatNickName: '
                                           17
                                                     wechatAvatarUrl: ''
                                           18
  () package-lock.json
                                           19
  (o) project.config.json
                                           20
                                                 })
```

### 4.5.3 开发发帖功能

#### 上传图片到云存储

打开 miniprogram/pages/publish 目录下的 publish.js 文件,在 send 事件依次上传多张图片,所有图片上传成功后调用 publish 方法并传入上传成功的 fileID 列表。

```
+ Q
                        -≡ 8←
                                    publish.js
                                   126
                                              for (let i = 0; i < img_url.length; i++) {
127
                                                var str = img_url[i];
▼ P miniprogram
                                                var obj = str.lastIndexOf("/");
                                   128
 ▶ ☐ images
                                                var fileName = str.substr(obj + 1)
                                   129
                                                console.log(fileName)
                                   130
 ▼ Pages
                                   131
  ▶ □ authorize
                                                //上传图片
                                   132
  ▶ [ ] logs
                                                wx.cloud.uploadFile({
                                   133
                                                  cloudPath: 'post_images/' + fileName,
  ▶ □ postdetail
                                   134
                                   135
                                                  filePath: img_url[i],
  ▶ □ postlist
                                   136
                                                  success: res => {
  ▼ D publish
                                   137
                                                    img url ok.push(res.fileID)
     JS publish.js
                                   138
                                                    if (img_url_ok.length == img_url.length) {
                                                      that.publish(img_url_ok)
                                   139
     () publish.json
                                   140
     <> publish.wxml
                                   141
     wxss publish.wxss
                                   142
                                                  fail: err => {
 ▶ □ utils
                                                    that.publishFail('图片上传失败')
                                   143
                                   144
                                                    console.log('fail: ' + err.errMsg)
   JS app.js
                                   145
   () app.json
                                   146
                                                1)
   wxss app.wxss
                                   147
                                   148
  () min.config.ison
```

#### 调用发帖云函数

写入数据库的操作我们让云函数来执行,小程序端只需要将需要调用相应云函数并将需要写入的内容传给它。打开 miniprogram/pages/publish 目录下的 publish.js 文件,调用名为 publish\_post 的云函数,传递作者昵称、作者头像 url、正文内容、配图 fileID 列表。

```
+ Q
                             ₽←
                                    publish.js
                        •=
                                                 ×
                                              * 执行发布
                                     73
▶ 🙆 cloudfunctions | demo
▼ Iminiprogram
                                     75
                                            publish: function(img_url_ok) {
                                     76
                                               var that = this
 ▶ [☐ images
                                     77
                                               wx.cloud.callFunction({
 ▼ 🗁 pages
                                                 name: 'publish_post',
                                     78
  ▶ □ authorize
                                                 //传递数据
                                     79
  ▶ [☐ loas
                                                 data: {
                                     80
                                     81
                                                   author_name: app.globalData.wechatNickName,
  Dostdetail
                                     82
                                                   author avatar url: app.globalData.wechatAvatarUrl,
  ▶ □ postlist
                                     23
                                                   content: that.data.content,
  ▼ Dublish
                                     84
                                                   image_url: img_url_ok,
                                     85
      JS publish.js
                                     86
                                                  success: function (res)
      () publish.json
                                     87
                                                   var pages = getCurrentPages();
      <> publish.wxml
                                                                                                 // 上一个员
                                                   var prevPage = pages[pages.length - 2];
                                     88
      wxss publish.wxss
                                                   prevPage.setData({
```

#### 写入新帖数据

当 publish\_post 云函数被触发,需要将上一步中小程序端传入的数据及微信用户的 openid、发布时间写入到 post\_collection 这个集合中,并初始化最后更新时间和浏览数。

打开 cloudfunctions/publish\_post 目录下的 index.js 文件,在云函数入口函数中键入代码。修改完后记得保存,并上传部署到云端噢~

```
index.js
                                          cloud.init()
▼ 🗟 cloudfunctions | demo
 ▶ ு add_comment
                                          const db = cloud.database()
                                     6
 ▶ ு get_comment_for_post
                                          // 云函数入口函数
                                          exports.main = async (event, context) => {
 ▶ ७ get_post_detail
                                            .
//将新帖写入数据库
                                    9
 ▶ ு get post list
                                    10
 ▼ 🕝 publish post
                                    11
                                              return await db.collection('post_collection').add({
                                                data: {
                                    12
    JS index.js
                                    13
                                                  author_id: event.userInfo.openId,
    () package.json
                                                  author_name: event.author_name,
 ▶ ⋒ update_watch_count
                                    15
                                                  author_avatar_url: event.author_avatar_url,
▶ □ miniprogram
                                                  content: event.content,
                                    16
                                    17
                                                 image_url: event.image_url,
 gitignore .
                                    18
                                                  publish time: Date.now(),

☐ README.md

                                    19
                                                  update_time: Date.now(),
 () package-lock.json
                                    20
                                                  watch_count: 1,//浏览数
                                    21
  (o) project.config.json
                                              })
                                    22
                                    23
                                            } catch (e) {
                                    24
                                              console.error(e)
                                    25
                                    26
```

将上面步骤做完,此时你的论坛小程序就能够正常的发帖了,赶紧发个匿名帖试试吧! 至此,你可以在云开发控制台的数据库中,查看你的帖子内容。

### 4.5.4 开发看帖功能

#### 获取帖子列表

打开 cloudfunctions/get\_post\_list 目录下的 index.js 文件,获取 100 条帖子的 ID、正文内容、浏览数、最新更新时间,并按最后更新时间倒序排列后返回给小程序端。修改完毕后保存,上传并部署到云端。

对小程序进行重新编译,此时的论坛小程序首页就能完整展示帖子列表了,但还无法点看某篇帖子进行查看。

```
index.is
▼ 🗟 cloudfunctions | demo
                                       const db = cloud.database()
 ▶ ு add_comment
                                       // 云函数入口函数
                                   8
 ▶ ७ get_comment_for_post
                                       exports.main = async (event, context) => {
 ▶ ७ get_post_detail
                                  10
                                         return {
                                           // 获取帖子列表
 ▼ ఉ get_post_list
                                  11
                                           postlist: await db.collection('post_collection').field({
   JS index.js
                                             _id: true,
                                  13
    () package.json
                                  14
                                             content: true.
 ▶ ு publish_post
                                             watch_count: true,
                                  15
                                             update_time: true
                                  16
 ▶ ₷ update_watch_count
                                           }).orderBy('update_time', 'desc').get()
                                  17
▶ □ miniprogram
                                  18
  .gitignore
                                  19
  README.md
                                  20
```

#### 获取帖子详情

打开 cloudfunctions/get\_post\_detail 目录下的 index.js 文件,小程序端会传入 postid,我们拿着 postid 去云数据中查询对应的帖子详情,并返回给小程序端。修改完毕 后保存,上传并部署到云端。

```
··· <u>∗≡</u> 8÷
                                     index.js
▼ 🗟 cloudfunctions | demo
                                           // 云函数入口函数
 ▶ ⋒ add_comment
                                           exports.main = async(event, context) => {
 ▶ ு get_comment_for_post
                                     10
 ▼ 
    get_post_detail
                                     11
                                             return {
                                               //获取帖子详情
                                     12
JS index.js
                                               postdetail: await db.collection('post_collection').where({
                                     13
    () package.json
                                                 id: event.postid
                                     14
 ▶ ७ get_post_list
                                     15
                                               }).get(),
                                     16
 ▶ ு publish_post
                                     17
 ▶ ு update_watch_count
```

#### 更新浏览数

打开 cloudfunctions/update\_watch\_count 目录下的 index.js 文件,小程序端会传入 postid,我们使用 postid 查询相应帖子的记录并使其浏览数+1。修改完毕后保存,上传并 部署到云端。

```
.≡ 8←
+
     Q
                                    index.js
                                      7
                                          const db = cloud.database()
▼ a cloudfunctions I demo
                                     8
 ▶ ு add_comment
                                      9
                                          const _ = db.command
 ▶ ♠ get_comment_for_post
                                          // 云函数入口函数
                                    10
                                          exports.main = async (event, context) => {
                                    11
 ▶ ७ get_post_detail
                                             //再新帖子浏览数
                                    12
 ▶ ₯ get post list
                                             await db.collection('post_collection').where({
                                    13
 ▶ ு publish_post
                                               _id: event.postid
                                    14
 ▼ ⇔ update_watch_count
                                    15
                                            }).update({
                                    16
                                              data: {
     JS index.js
                                    17
                                               watch_count: _.inc(1)
    () package.json
                                    18
                                              },
iminiprogram
                                    19
                                              success: function (res) {
 .gitignore
                                    20
                                                 console.log(res.data)
                                    21
 README.md
                                    22
                                            })
 () package-lock.json
                                    23
(o) proiect.confia.ison
```

### 4.5.5 开发评论功能

#### 更新帖子最新更新时间

打开 cloudfunctions/add\_comment 目录下的 index.js 文件 / 小程序端会传入 postid ,我们使用 postid 查询相应帖子的记录并设置其最后更新时间为当前服务器时间。修改完毕后保存,上传并部署到云端。

```
index.js
                                    11
▼ 🙆 cloudfunctions | demo
                                    12
                                          // 云函数入口函数
                                          exports.main = async (event, context) => {
                                    13
 ▼ 🔈 add_comment
                                    14
    JS index.js
                                    15
                                              const timestamp = Date.now()
    () package.json
                                              // 更新帖子最后更新时间
                                    16
                                    17
                                              await db.collection('post_collection').where({
 ▶ ⋒ get_comment_for_post
                                    18
                                                _id: event.postid
 ▶ ७ get_post_detail
                                    19
 ▶ ு get_post_list
                                    20
                                              .update({
 ▶ ு publish_post
                                    21
                                                data: {
                                    22
                                                  update_time: timestamp
 ▶ ๘৯ update_watch_count
                                    23
▶ ☐ miniprogram
                                    24
 .gitignore
                                    25
 README.md
                                    26
                                              await db.collection('comment_collection').add({
                                    27
                                                data: {
 () package-lock.json
                                    28
                                                  postid: event.postid,//评论对应的post
 (o) project.config.json
                                                  openid: event.userInfo.openId,
                                                  name: event.name,//评论者名字
                                    30
```

如果上面的步骤你都已经完成了,你已经完成了一个简单小程序的搭建,并将最核心的

能力使用小程序·云开发完成,快使用微信开发者工具或者使用手机扫码预览体验吧!

# 4.6 完成标准

- 1、论坛小程序发帖、看帖、评论 3 大功能能够正常使用
- 2、所有帖子、评论数据需要借助云数据库能力存储在云端,并从云端拉取展示
- 3、所有图片类文件需要借助云存储能力存储在云端,并从云端拉取展示
- 4、任何小程序页面没有 Warning 报错

### 4.7 附加任务

如果你提前完成了云开发实战任务,可以思考一下,如何将论坛小程序改为可以显示头 像和昵称的实名社区?如何实现帖子收藏功能?如何实现删帖功能?



关注腾讯云「云开发」 上小程序,用云开发

**M**Tencent UR 特别鸣谢: 腾讯高校合作

**Broadview**