


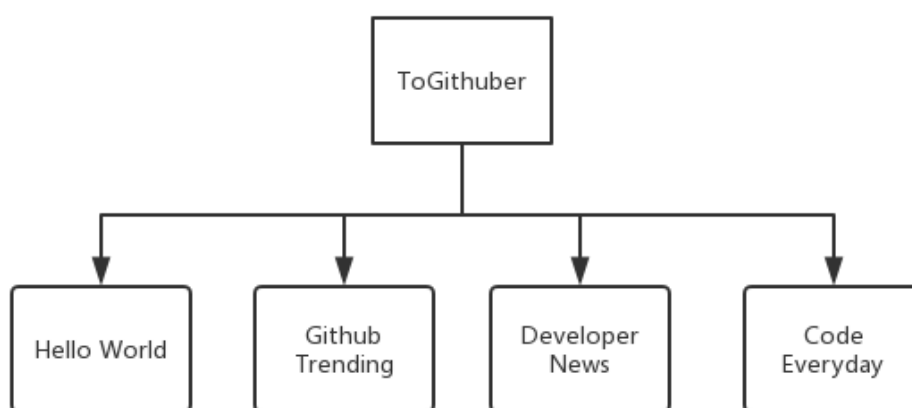
ToGithuber

小程序云开发作品介绍

一、 小程序由来：

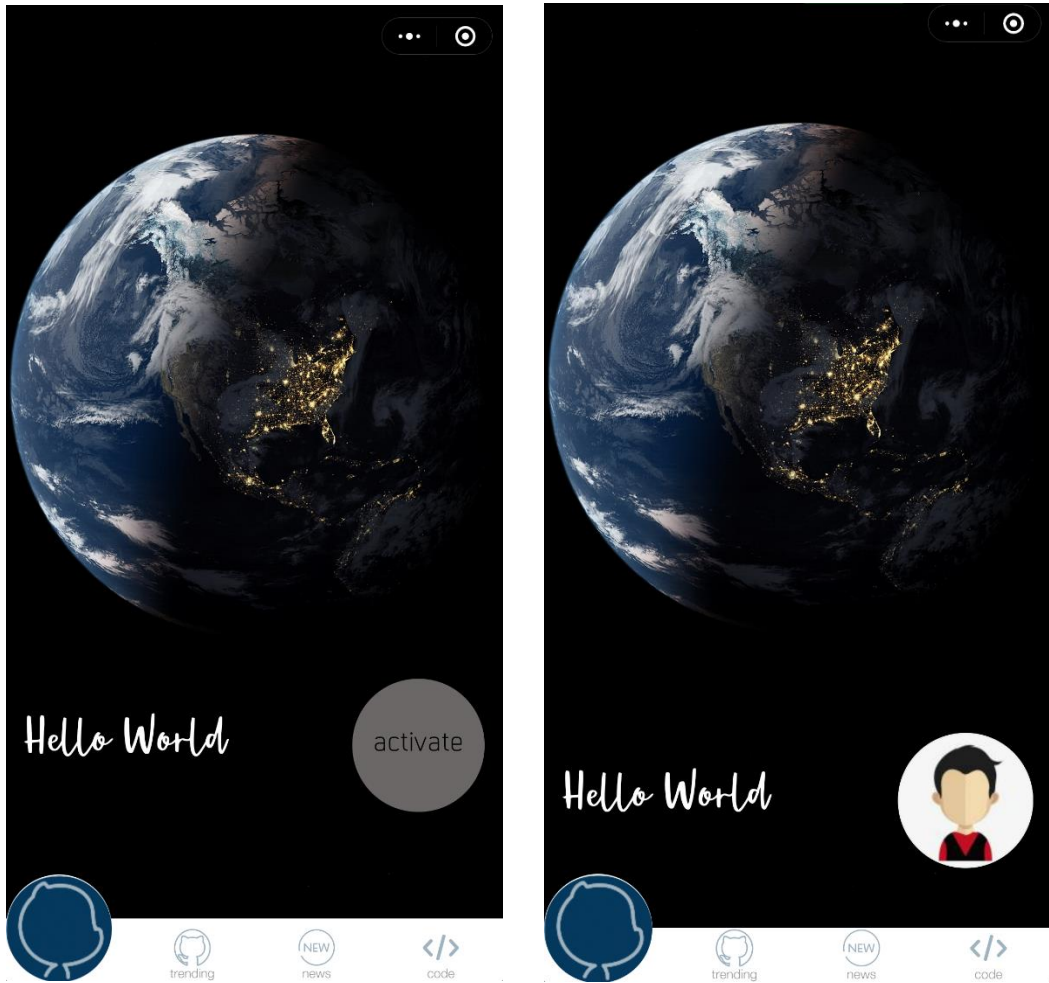
ToGithuber，诞生于 2019 年 5 月份，小程序是结合本人作为一名高校计算机专业的学生日常生活，平时希望能够在空闲的时候利用好零碎的时间，不仅能够实时跟进 Github 社区，了解互联网新鲜的新闻资讯以及刷刷 leetcode，对于各大客户端 app 而言只是在零碎时间使用的需求而显得繁琐，因此利用小程序的即时即用的特点，能最大限度的为程序员提供方便，充分利用碎片化时间。通过近一个月的个人独立开发，使用到小程序云开发，快速方便的实现了这个想法，“From Github, To Githuber”，也用来致敬 Github 社区，小程序的源码地址：<https://github.com/10ngui/ToGithuber>，小程序已于 20190531 上线，

二、 小程序功能需求：



● Hello World 首页设计：

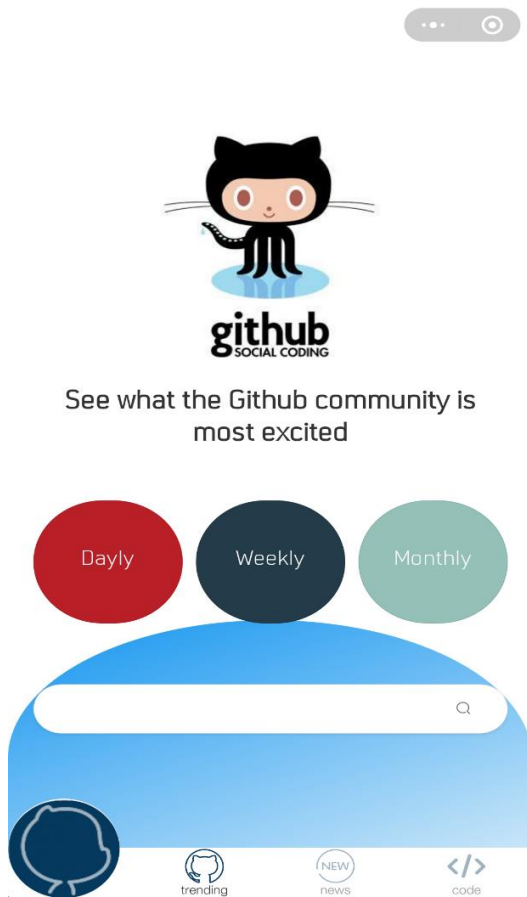
首页展示了宏伟的地球全景图，并以 Hello World 向地球致敬，同样的，Hello World 也是广大程序员一句经典引用与文化，点击右边的圆形按钮则会引导用户授权信息。



- Github trending

Github 作为全世界的程序员交流与方向的社区也作为开源设计的代表，其优质的项目不仅引领着世界前沿技术的潮流，也凝聚了广大程序员自由分享合作的精神。

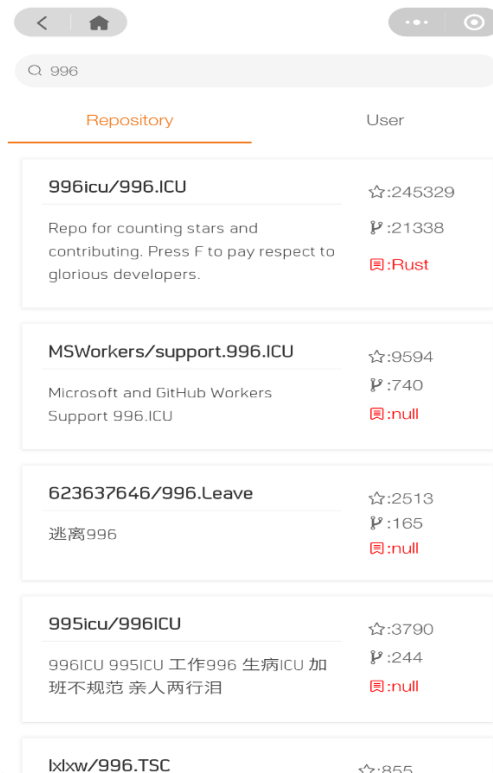
借用 Github API，小程序集成了对接 Github 的功能模块，在这个模块中，用户可以实时获取日，周，月的 github trending 优质项目，选择不同的语言类型，用户可以进入项目详情页面查阅该项目的 README，并且了解更多项目拥有者相关的个人信息以及关注的项目，并且用户可以在此模块中搜索项目。



Python - 100天从新手到大师

作者：骆昊

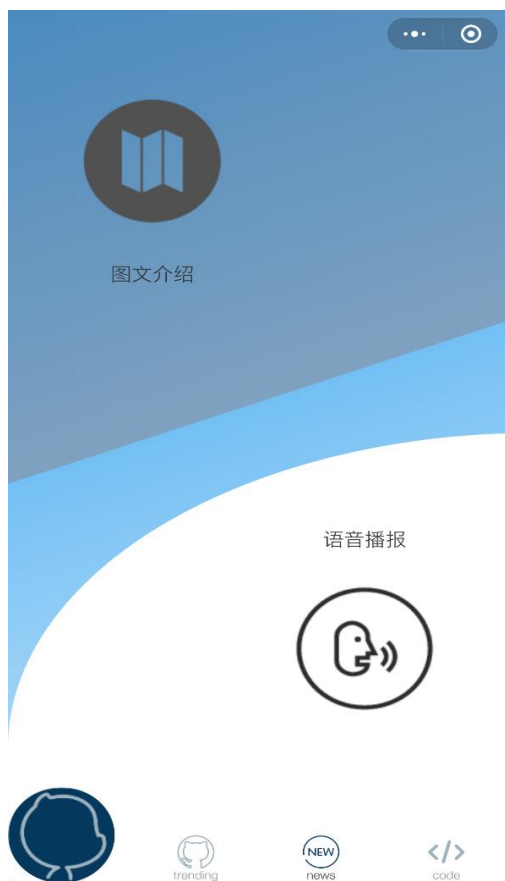
说明：最近有很多想学习Python的小伙伴申请单独加我微信和QQ，因为我自己平时也很忙，没办法一一解答大家的问题，我创建了 **Python100天学习交流1群** 和 **Python100天学习交流2群** 两个交流群，群号分别为 680701107 和 213132758，二维码在下方。我的同事和朋友很多也在这个群里，他们都是优秀的Python开发者，有丰富的商业项目经验，我们在时间充足的时候会及时解答大家的问题，而且从Python语言入门到Web应用开发，从数据分析到机器学习，每个领域都有技术大咖为大家解惑答疑。以后我们争取每周做一次视频直播，以专题的形式分享Python开发的点点滴滴，同时还会不定期的举办线上和线下的技术交流和分享活动，小伙伴们可以加群进行交流。感谢 千锋教育



● Developer News

作为一名开发者，希望在平时空闲时间了解互联网相关的科技动态，但如今各大新闻客户端对于开发者所关注的新闻少之又少，因此希望能够利用小程序即时即用的特点，能够快速获取互联网前沿动态。

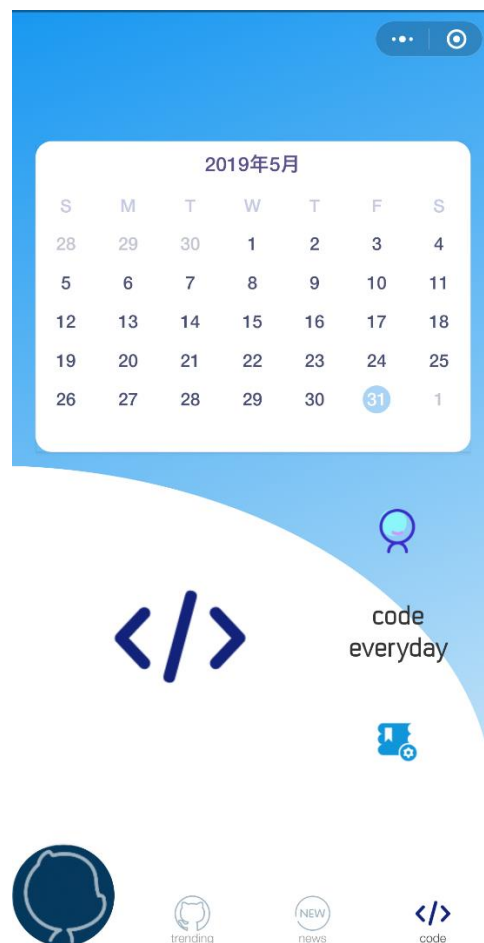
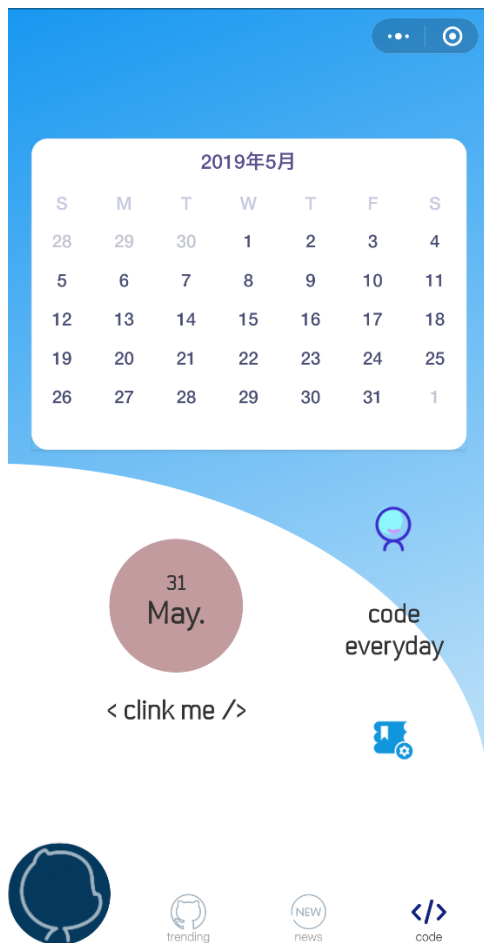
使用网络爬虫爬取开发者常关注的新闻专栏并对接小程序的功能模块，用户可以实时获取到优质的互联网科技动态，除此，更具特色的一点是不同于常规的图文浏览新闻内容，语音播报或许可以在某些时候更方便的让用户获取新闻资讯。





● Code Everyday

相信每个程序员对 leetcode 都不陌生，算法刷题对于我们无论是高校在读生或是求职工作者，坚持每天刷几道算法题是个非常不错的习惯，那么，同样使用小程序可以充分利用零碎时间，能够让用户在不同地点不同时间的空闲了解几道 code 题，并且在此小程序中所使用的题都是 leetcode 官方原汁原味的英文题，这让同本人的高校在读生来说更能提升自身的能力和体会算法题的乐趣，同时，在刷题的同时，你可以记录你的笔记以及对收藏有意思的题。



Number of Squareful Arrays

Math / Backtracking / Graph / Hard

Given an array `A` of non-negative integers, the array is *squareful* if for every pair of adjacent elements, their sum is a perfect square. Return the number of permutations of `A` that are squareful. Two permutations `A1` and `A2` differ if and only if there is some index `i` such that `A1[i] != A2[i]`.

Example 1:

Input: [1,17,8]

Output: 2

Explanation: [1,8,17] and [17,8,1] are the valid permutations.

Example 2:

Input: [2,2,2]

Output: 1

Note:

- 1. `1 <= A.length <= 12`
- 2. `0 <= A[i] <= 1e9`

🤍 收藏 📖 帮助

我的收藏	我的笔记
1. 🔥 Search a 2D Matrix	Medium >
2. 🔥 Permutations	Medium >
3. 🔥 Combination Sum II	Medium >
4. 🔥 Rotate List	Medium >

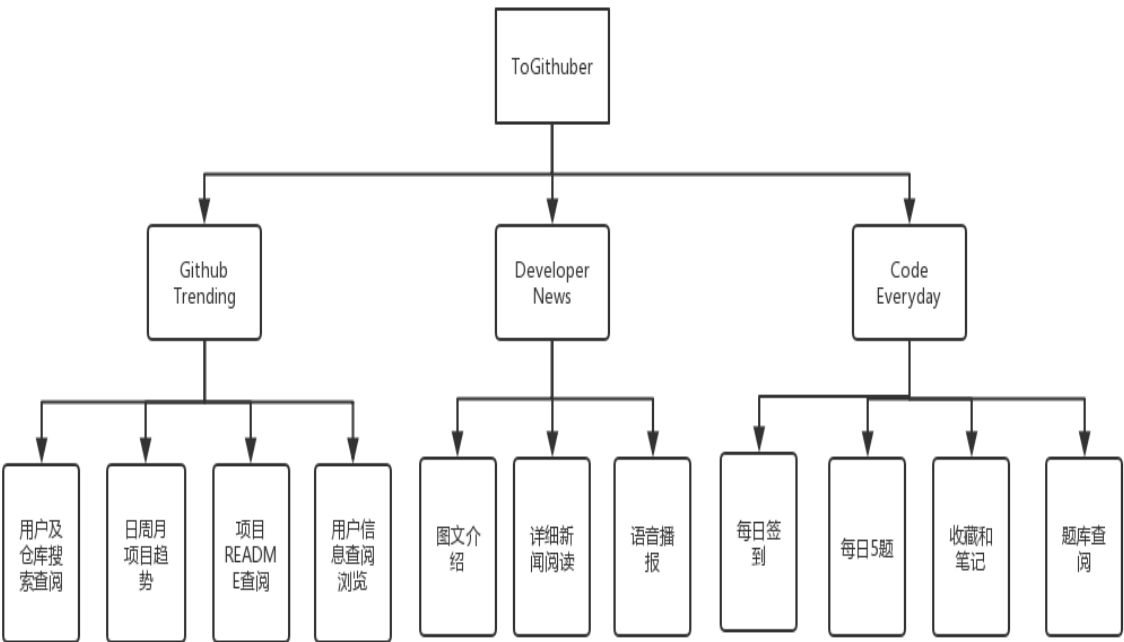
题库分类	
Easy	>
Medium	>
Hard	>
Array	168 >
Dynamic Programming	139 >
Math	129 >
String	123 >
Tree	114 >
Hash Table	104 >
Depth - first Search	99 >
Binary Search	67 >
Two Pointers	56 >
Greedy	52 >
Breadth - first Search	49 >

Medium	
🔥 Video Stitching	Medium >
🔥 Longest String Chain	Medium >
🔥 Flip Equivalent Binary Trees	Medium >
🔥 Broken Calculator	Medium >
🔥 Binary String With Substrings Representing 1 To N	Medium >
🔥 Number of Enclaves	Medium >
🔥 Maximum Sum of Two Non-Overlapping Subarrays	Medium >
🔥 Delete Columns to Make Sorted II	Medium >
🔥 Minimum Increment to Make Array Unique	Medium >
🔥 Vertical Order Traversal of a Binary Tree	Medium >
🔥 Most Stones Removed with Same Row or Column	Medium >
🔥 Numbers With Same Consecutive Differences	Medium >
🔥 Best Sightseeing Pair	Medium >
🔥 Construct Binary Search Tree from	Medium >

三、技术分析

1. 小程序端

● 功能结构如下：



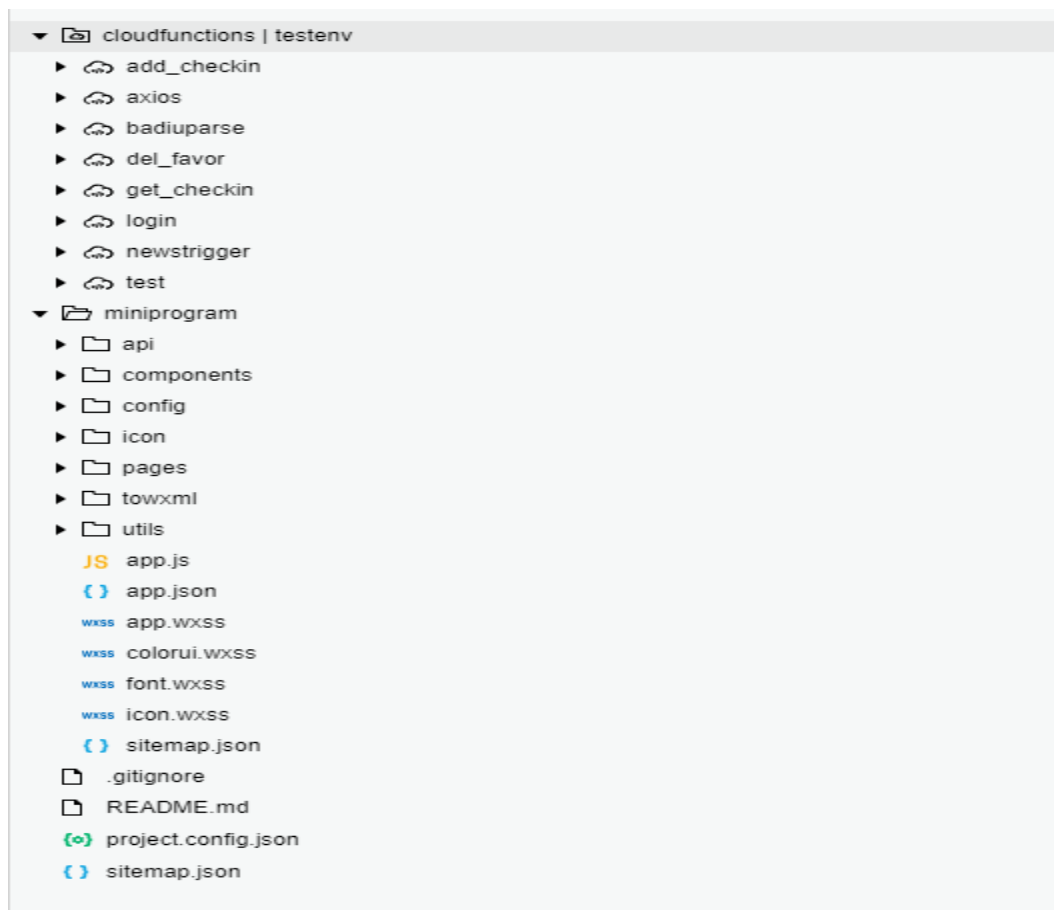
● 功能点说明：

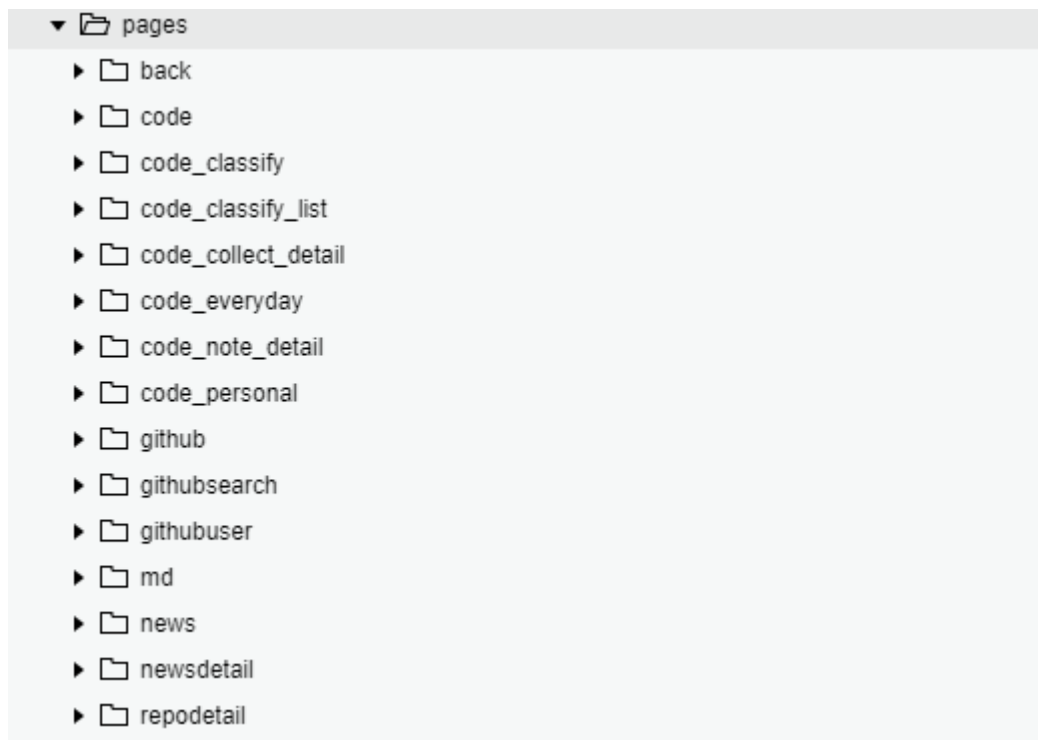
子功能点	子功能描述	所属模块
日周月项目趋势	实时获取 github 社区所关注的火热项目	Github Trending

项目 README 查阅	查阅该项目的 README (支持仓库 markdown 文件的跳转浏览)	Github Trending
用户 (组织) 信息浏览	查看项目所属组织或用 户相关的信息及关联的 其他项目	Github Trending
用户及仓库搜索查阅	通过关键词对 github 中的用户及项目进行搜 索	Github Trending
图文介绍	以图文 (封面加标题) 的方式获取新闻资讯列 表	Developer News
语音播报	以语音播报 (标题加语 音) 的方式获取新闻内 容	Developer News
详细内容阅读	进入新闻详细页面,阅读 新闻内容, 作者和出处	Developer News
每日签到	记录自己的打卡签到记 录	Code Everyday
每日 5 题	小程序每天随机为用户 分配 5 道题, 供用户零 碎时间刷题	Code Everyday

收藏和笔记	用户可以对每日分配的题目进行收藏以及做笔记,并在个人中心查阅	Code Everyday
题库分类查询	对题库进行标签以及难易程度分类便于用户针对性的整理和查询	Code Everyday

● 目录结构如下：

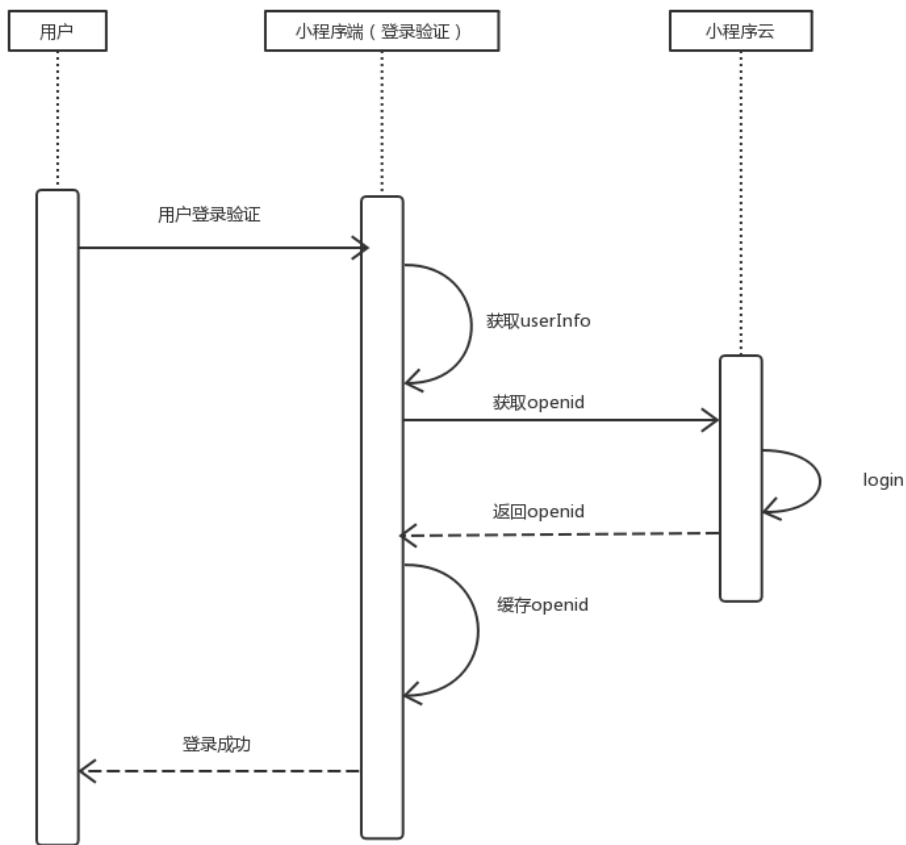




● Hello World 首页（登录验证）

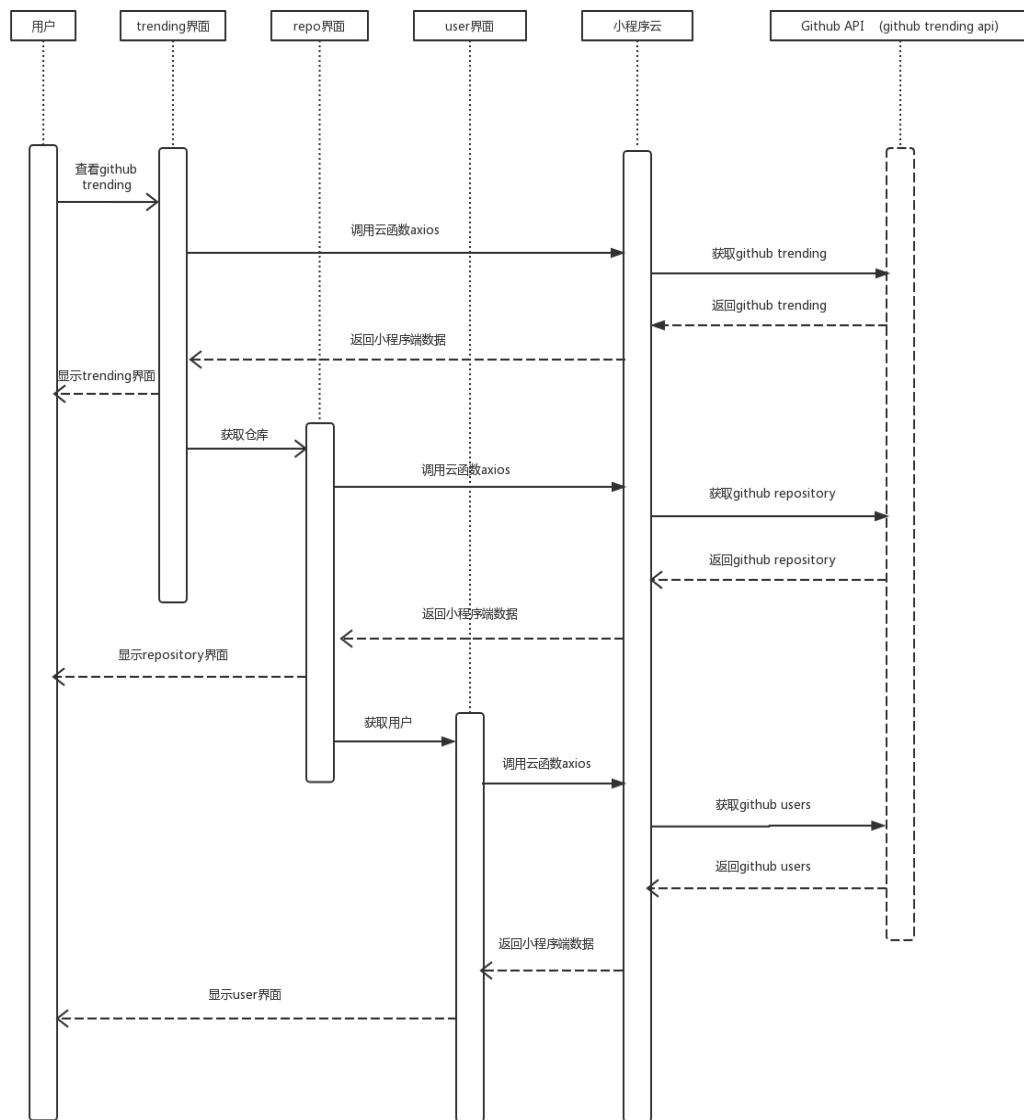
基本功能：

用户点击登录验证按钮触发 `getUserInfo` 函数，其间获取用户的用户信息并调用云函数返回用户的 `openid` 存储本地缓存，时序图如下：



● Github Trending

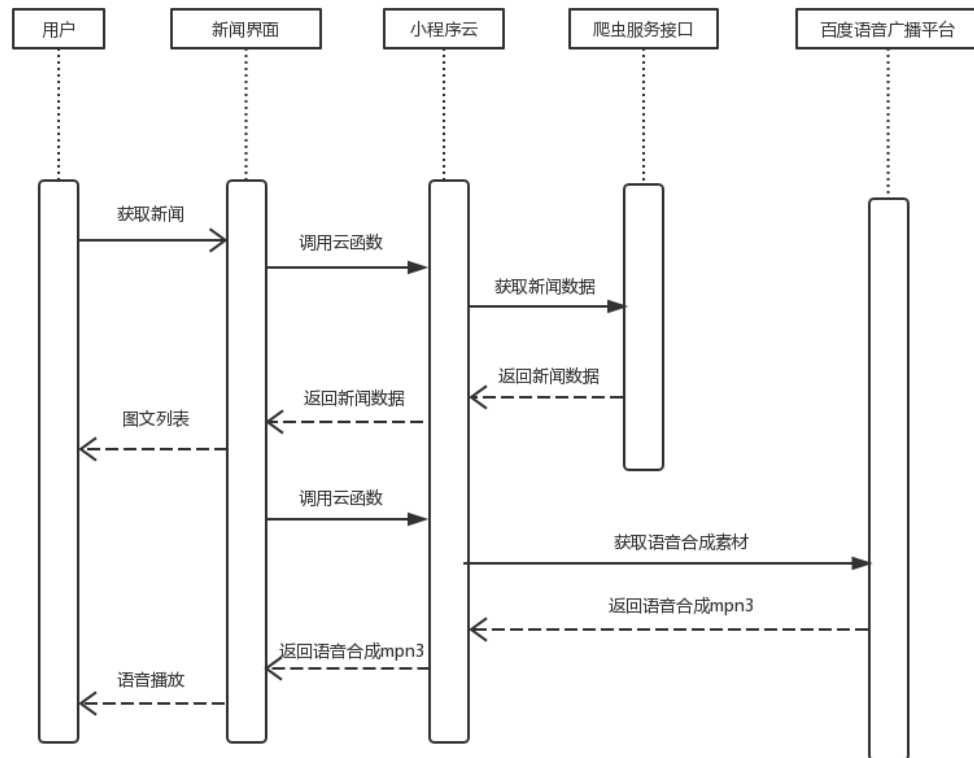
基本功能：使用 Github api 对开发者开放的基本功能，用户可以实时获取日，周，月的 github trending 中的优质项目，选择不同的编程语言，以及对项目，用户的查阅，因个人开发者小程序类别的限制，部分内容小程序提供了链接需要用户自行前往浏览器查阅。时序图如下：



● Developer News

基本功能:使用网络爬虫获取优质的互联网科技新闻后存储到本地的 MongoDB 数据库中 ,其中网络爬虫使用 PySpider 爬虫框架 ,并设置定时增量爬虫 ,每一个小时更新数据库中的新闻 ;在小程序端实现一个云函数使用 mongodb 模块对接远程的爬虫

服务接口，调用云函数获取新闻内容并展示；同时使用百度的第三方语音广播开放平台的 API，获取到新闻语音并在小程序端播放，时序图如下：



● Code Everyday

基本功能：使用网络爬虫一次性获取到 leetcode 官方的题库并保存到小程序的云数据库中，并设置该集合的权限为所有用户仅刻度，每天小程序推送 5 道随机的题目，并且用户记录下签到日，以及对题目进行收藏和笔记。

- 使用到的组件:

- ColorUI : ColorUI 专注于视觉交互,拥有极其鲜亮的高饱和色彩,多种页面元素可供选择,高度自定义的组件,丰富的扩展引入。

(<https://github.com/weilanwl/ColorUI>)

- WxParse:微信小程序富文本解析组件

(<https://github.com/icindy/wxParse>)

- Towxml:HTML,Markdown 转信小程序 WXML 渲染组件

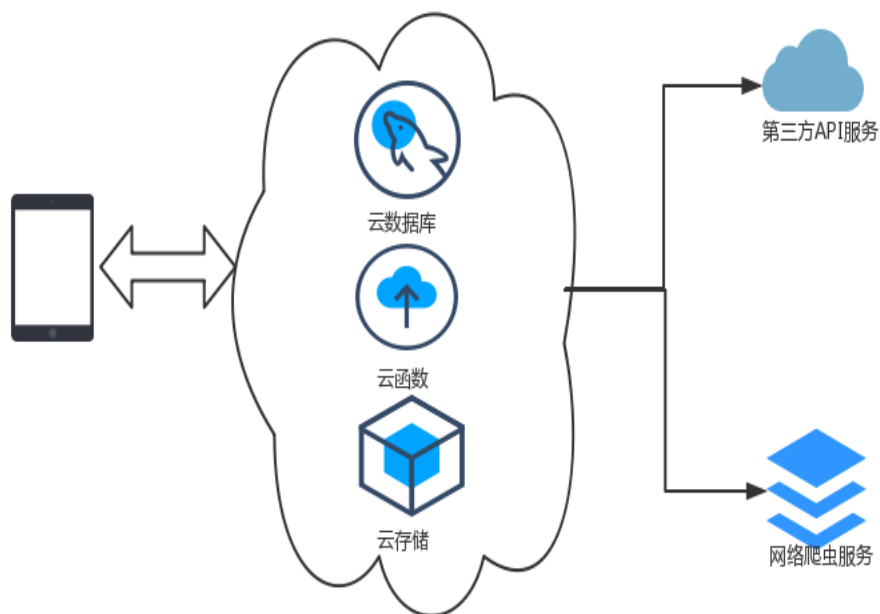
(<https://github.com/sbfkcel/towxml>)

- Calendar: 支持 1900 年 1 月 ~ 2099 年 12 月这两百年间的公历和农历显示,

(<https://github.com/czcaiwj/calendar>)

2. 云开发服务端

2.1 主要技术架构:



2.2 云数据库字段设计:

User:

```
{  
  "_id": "openid",  
  "userinfo": {  
    "avatarUrl": "",  
    "nickName": "",  
    "province": ""  
  }  
}
```

```
}  
}
```

News:

```
{  
  
  "_id":ObjectId(""),  
  
  "url" : "",  
  
  "simple_content" : "",  
  
  "time" : "",  
  
  "title" : "",  
  
  "author" : "",  
  
  "content" : "",  
  
  "cover_img" : "",  
  
  "label" : [],  
  
  "from" : ""  
  
}
```

checkin_date:

```
{  
  
  "_id": "",  
  
  "checkin": [  
  
    {  
  
      "background": "",  
  
      "color": "",  
  
    }  
  
  ]  
  
}
```

```
        "day": "",
        "month": ""
    }
],
"count": 1,
"userid": "openid"
}
```

Leetcode_favors:

```
{
    "_id": "",
    "_openid": "openid",
    "createtime": "",
    "problemid": ""
}
```

Leetcode_notes:

```
{
    "_id": "",
    "content": "",
    "difficulty": "",
    "fronted_id": "",
    "id": "",

```

```
"slug": "",  
  
"title": ""  
  
}
```

2.3 云函数设计:

axios 函数 (代替小程序端对第三方 api 的请求访问 , 可以省去域名备案):

```
// 云函数入口文件  
  
const cloud = require('wx-server-sdk')  
  
const axios = require('axios')  
  
cloud.init()  
  
axios.defaults.validateStatus = () => true  
  
// 云函数入口函数  
  
exports.main = async (event, context) => {  
  
  return new Promise((resolve, reject) => {  
  
    axios(event).then(({ status, headers, data }) => {  
  
      resolve({ status, headers, data })  
  
    }).catch(err => {
```

```
        reject(err)

    })

  })
}
```

login 函数 (用户第一次登录则向数据库中存储用户的 userInfo 并以 openid 为主键):

```
// 云函数入口文件

const cloud = require('wx-server-sdk')

const axios = require('axios')

cloud.init()

axios.defaults.validateStatus = () => true

// 云函数入口函数

exports.main = async (event, context) => {

  return new Promise((resolve, reject) => {

    axios(event).then(({status, headers, data}) => {

      resolve({status, headers, data})

    }).catch(err => {

      reject(err)

    })

  })
}
```

```
}
```

Get_news 函数(通过 mongodb 模块远程建立爬虫数据库的连接并进行数据的读取)：

```
// 云函数入口文件

const cloud = require('wx-server-sdk')

cloud.init()

var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://106.14.159.80:27017";

var db = cloud.database();

var _ = db.command;

// 云函数入口函数

exports.main = async(event, context) => {

  const {

    skip , limit

  } = event

  return new Promise((resolve,reject)=>{

    MongoClient.connect(url, {

      useNewUrlParser: true

    }, function (err, db) {

      if (err) {

        console.log(err)

      }

    })

  })

}
```

```

        throw err

    };

    var dbo = db.db("techweb");

    dbo.collection("news").find({

    }).sort({

        "time": -1

    }).limit(limit).skip(skip).toArray(function (err, result) {

        if (err) {

            reject(err)

        }

        console.log(result)

        resolve(result)

    });

    });

})

}

```

Baiduparse 云函数 (使用百度语音广播开放平台的 api 获取新闻内容语音合成):

```

// 云函数入口文件

const cloud = require('wx-server-sdk')

const request = require('request')

```

```
cloud.init()

// 云函数入口函数

exports.main = async(event, context) => {

  const wxContext = cloud.getWXContext()

  const {

    url

  } = event

  const parse = await new Promise((reslove, reject) => {

    request.post({

      url: 'https://developer.baidu.com/vcast/parserPageUrl',

      form: {

        url

      }

    },

    (err, httpResponse, body) => {

      data = JSON.parse(body)

      reslove(data)

    })

  }).then(data => {

    obj = new Object(data)

    let defaultoption = {
```



```

    sex: 4,

    speed: 5,

    volumn: 7,

    pit: 5,

    method: 'URL'
  }

  option = {

    title: obj.title.replace('TechWeb', ''),

    content: content,

    ...defaultoption
  }

  return option

  console.log(option)
})

const audio = await new Promise((reslove, reject) => {

  request.post({

    url: 'https://deveLoper.baidu.com/vcast/getVcastInfo',

    form: parse,

    headers: {

      Cookie: ''

    }

  }, (err, httpResponse, body) => {

    reslove(body)
  })
})

```

```
    })  
  })  
  
  return {  
    code:1,  
    audio:new Object(JSON.parse(audio))  
  }  
}
```