微信小程序开发

黑马程序员
www.itheima.com

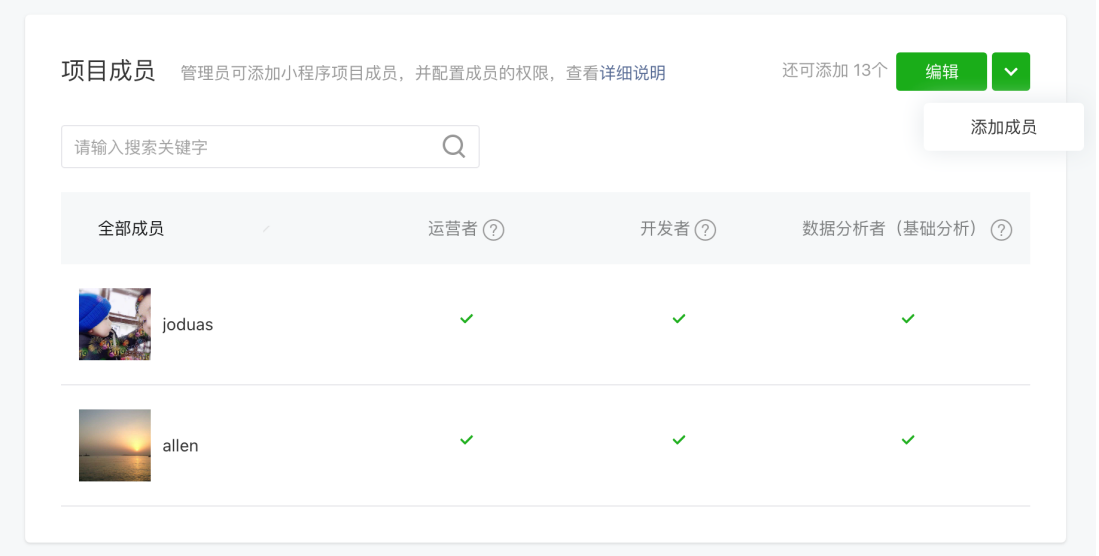
传智教育旗下
高端IT教育品牌

团队开发准备

- 申请权限
- 初始开发环境

团队开发 - 申请权限

- 申请开发权限：提供给开发者
  - 提供个人微信号
  - 获取团队小程序的 AppID
  - 创建/导入小程序
- 申请体验权限：提供给测试、产品、客户等
  - 后台添加，提供个人微信号
  - 扫码申请



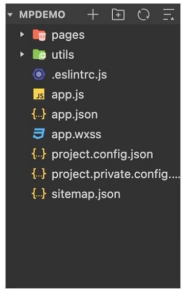
02

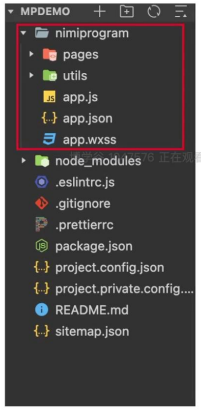
开发准备

- 项目启动
- 基础封装

团队开发 - 初始开发环境

- 优化目录结构
  - miniprogramRoot: 小程序的根目录
  - setting.packNpmManually: 自定义构建npm
  - setting.packNpmRelationList: 构建结果和package.json
- 启用 less/sass
  - setting.useCompilerPlugins 启用 less/sass
- 配置 VS Code
  - Prettier - Code formatter
  - WXML-Language Service
  - ... 其它






```
// 调整了代码结构，小程序打包时找不到代码了
// key: value（小程序业务代码放的位置）
"miniprogramRoot": "miniprogram/",
"setting": {
  // 安装的依赖需要构建，调整了目录结构，也需要自定义npm构建
  "packNpmManually": true,
  "packNpmRelationList": [
    {
      "packageJsonPath": "./package.json", // 构建需要查找的依赖
      "miniprogramNpmDistDir": "./miniprogram" // 构建后代码存放的位置，一般和业务代码放一起
    }
  ],
  // 启用sass
  "useCompilerPlugins": ["sass"],
```

享+生活 - 项目启动

- 拉取代码

```
# gitee 仓库
git clone -b template https://gitee.com/lotjol/enjoy-plus.git
# 修改分支名称
git branch -m template main
# 修改仓库地址
git remote set-url origin '这里是你的远程仓库地址'
```

- 运行小程序
  - 通过小程序开发者工具导入项目
  - 通过 vscode 安装项目的依赖

享+生活 - 基础封装（消息反馈）

```
const utils = {
  /**
   * 消息反馈（轻提示）
   * @param {string} title 文字提示内容
   */
  toast(title = '数据加载失败...') {
    wx.showToast({
      title,
      mask: true,
      icon: 'none'
    })
  }
}

// 正常的模块导出
export default utils
// 也可以挂到全对象 wx 上
wx.utils = utils
```

```
// 导入工具方法库
import utils from "../../utils/utils"

Page({
  onLoad() {
    // 模块方式调用
    utils.toast('页面加载完毕...')
    // 全局对象方式调用
    wx.utils.toast('页面加载完毕...')
  }
})
```

安装 VSCode 插件：微信小程序开发工具

享+生活 - 基础封装（网络请求）

```
// 导入 wechat-http 模块
import http from 'wechat-http'

// 接口基础路径
http.baseURL = 'https://live-api.itheima.net'

// 响应拦截器
http.intercept.response = function ({data}) {
  return data
}

// 以模块方式导出
export default http
// 挂载到全局对象 wx 上
wx.http = http
```

```
// 执行 http.js
import './utils/http.js'

App({
  globalData: {},
})
```

```
Page({
  async onLoad() {
    // 请求接口数据（全局方式调用）
    const res = await wx.http.get('/announcement')
    console.log(res.data.data)
  }
})
```

公告管理

- 公告列表
- 公告详情

享+生活 - 公告管理（公告列表）

- 接口路径：/announcement
- 接口参数：无
- 响应数据：


```
Page({
  async onLoad() {
    // 获取公告列表数据
    this.getNotices()
  },

  // 调用公告列表接口
  async getNotices() {
    // 请求接口数据（全局方式调用）
    const {code, data: notices} = await wx.http.get('/announcement')
    // 校验接口是否调用成功
    if(code !== 10000) return wx.utils.toast()
    // 渲染数据
    this.setData({notices})
  }
})
```

享+生活 - 公告管理（公告详情）

- 接口路径：GET /announcement/{id}
- 接口参数：
  - id: 公告的 ID
- 响应数据：

```
▼ object {2}
    __webviewId__ : 35
  ▼ noticeDetail {5}
      content : <p>公告详情公告详情公告详情公告详情公告详情</p>
      createdAt : 2022-09-29 14:16:57
      creatorName : 传智教育
      id : 2
      title : 中秋、国庆温馨提示
```

```
Page({
  data: {
    noticeDetail: {}
  },
  onLoad({id}) {
    // 获取公告详情
    this.getNoticeDetail(id)
  },
  // 调用公告详情接口
  async getNoticeDetail(id) {
    // id 不存在时不必执行
    if(typeof id === undefined) return
    // 调用接口
    const {code, data: noticeDetail} = await wx.http.get('/announcement/' + id)
    // 验证接口是否调用成功
    if(code !== 10000) return wx.utils.toast()
    // 数据渲染
    this.setData({noticeDetail})
  }
})
```

04

用户管理

- 登录检测
- 短信验证码
- 登录/注册
- 头像和昵称

享+生活 - 用户管理（登录检测）

- 封装组件: authorization
  - 定义和注册组件
  - 组件作为页面的根节点

```
<!-- 作为页面的根点使用 -->
<authorization>
  <view class="profile">…
  </view>
</authorization>
```

  - 条件渲染插槽

```
<!-- 未登录时不使用插槽 -->
<slot wx:if="{{isLogin}}"></slot>
```

享+生活 - 用户管理（登录检测）

1. 读取 token：onLaunch

```
App({
  onLaunch() {
    // 获取 token
    this.getToken()
  },
  getToken() {
    // 读取本存储的 token
    this.token = wx.getStorageSync('token')
  }
})
```

2. 检测 token：attached

```
attached() {
  // 获取登录状态
  const isLogin = !!getApp().token
  // 更新登录状态
  this.setData({isLogin})
  // 未登录时跳转到登录页
  if(!isLogin) {
    wx.redirectTo({
      url: '/pages/login/index',
    })
  }
}
```

享+生活 - 用户管理（短信验证码）

- 倒计时组件
- 表单数据验证
- 接口调用

享+生活 - 用户管理（短信验证码）

- 倒计时组件: van-count-down
  - time 指定倒计时的时长
  - change 监听时间的变化
  - use-slot 启用插槽

```
<van-count-down
  wx:else
  use-slot
  time="{{6000}}"
  bind:change="countDownChange"
>
  <text>{{timeData.seconds}}秒后重新获取</text>
</van-count-down>
```

```
Page({
  data: {
    countDownVisible: false,
  },
  // 监听时间变化
  countDownChange(ev) {
    this.setData({
      timeData: ev.detail,
      countDownVisible: ev.detail.minutes === 1 || ev.detail.seconds > 0
    })
  },
  // 获取短信验证码
  getSMSCode() {
    this.setData({countDownVisible: true})
  }
})
```

享+生活 - 用户管理（短信验证码）

- 表单数据验证: npm install wechat-validate
  - 引用 wechat-validate

```
// 导入 Behavior 封装
import wxValidate from 'wechat-validate'
Page({
  // 引用 wechat-validate
  behaviors: [wxValidate],
})
```

  - rules 属性定义数据验证规则

```
// 导入 Behavior 封装
import wxValidate from 'wechat-validate'
Page({
  behaviors: [wxValidate],
  rules: {
    mobile: [
      { required: true, message: '请填写手机号码!' },
      { pattern: /^1[3-8]\d{9}$/, message: '请填写正确的手机号码!' },
    ],
  },
})
```

  - validate 方法返回验证结果

```
// 获取短信验证码
getSMSCode() {
  // 验证 mobile
  const {valid, message} = this.validate('mobile')
  if(!valid) return wx.utils.toast(message)

  // 开始倒计时
  this.setData({countDownVisible: true})
}
```

享+生活 - 用户管理（短信验证码）

- 接口路径：GET /code
- 接口参数：
  - mobile: 手机号码
- 响应数据：


```
{code: 10000, message: "此验证码仅用于测试", data: {code: "163611"}}
  code: 10000
  data: {code: "163611"}
    code: "163611"
  message: "此验证码仅用于测试"
```

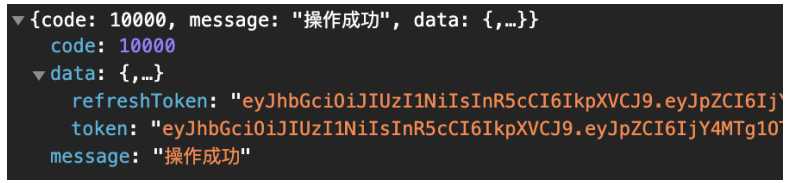

```
// 获取短信验证码
async getSMSCode() {
  // 验证 mobile
  const {valid, message} = this.validate('mobile')
  if(!valid) return wx.utils.toast(message)

  // 开始倒计时
  this.setData({countDownVisible: true})

  // 发送手机号获取验证码
  const {code, data} = await wx.http.get('/code', {mobile: this.data.mobile})
  // 检验验证码是否发送成功
  if(code !== 10000) return wx.utils.toast('发送失败，稍后重试!')
}
```

享+生活 - 用户管理（登录/注册）

- 调用接口
- 存储登录状态（token）
- 重定向（跳转页面）

享+生活 - 用户管理（登录/注册）

- 接口路径： POST /login
- 接口参数：
  - mobile: 手机号码
  - code: 验证码
- 响应数据：




```
// 提交表单完成登录/注册
async submitForm() {
  // 验证全部数据时自动报错提示
  if(!this.validate()) return
  // 调用登录接口
  const {code, data} = await wx.http.post('/login', {
    mobile: this.data.mobile,
    code: this.data.code
  })
  // 验证码接口是否成功返回数据
  if(code !== 10000) return wx.utils.toast('登录失败，稍后重试！')
}
```

享+生活 - 用户管理（登录/注册）

- 存储登录状态：
  - 存储到应用实例中
  - 存储到本地存储中

```
setToken(key, token) {
  // 将 token 存储到应用实例中
  this[key] = token
  // 将 token 存储到本地存储中
  wx.setStorageSync(key, token)
}
```

```
// 提交表单完成登录/注册
async submitForm() {
  // 验证全部数据时自动报错提示
  if(!this.validate()) return
  // 调用登录接口
  const {code, data} = await wx.http.post('/login', {
    mobile: this.data.mobile,
    code: this.data.code
  })
  // 验证码接口是否成功返回数据
  if(code !== 10000) return wx.utils.toast('登录失败，稍后重试!')
  // 存储登录状态
  app.setToken('token', data.token)
  app.setToken('refreshToken', data.refreshToken)
}
```

享+生活 - 用户管理（登录/注册）

- 重定向（页面跳转）：
  - authorization 组件中
    - getCurrentPages 获取页面路径
    - ? 号拼凑地址参数

```
// 获取页面路径
const pageStack = getCurrentPages()
const {route} = pageStack.pop()
// 未登录的情况下跳转到登录页面
if(!isLogin) {
  wx.redirectTo({
    url: '/pages/login/index?redirectURL=/' + route,
  })
}
```

  - login 页面中
    - onLoad 接收参数
    - wx.redirectTo 跳转页面

```
onLoad({redirectURL}) {
  // 获取地址中的参数
  this.redirectURL = redirectURL
},
```

```
// 重定向（页面跳转）
wx.redirectTo({url: this.redirectURL})
```

享+生活 - 用户管理（头像和昵称）

- 请求拦截器
  - 自定义 Authorization 头信息
- 接口路径：GET /userInfo
- 接口参数：无
- 响应数据：


```
{code: 10000, message: "操作成功", data: {avatar: null, id: "68631
  code: 10000
  data: {avatar: null, id: "6863118240485376", nickName: null}
    avatar: null
    id: "6863118240485376"
    nickName: null
  message: "操作成功"
```


```
// 配置请求拦截器
http.intercept.request = function (options) {
  // 扩展头信息
  const defaultHeader = {}
  // 身份认证
  defaultHeader.Authorization = 'Bearer ' + getApp().token
  // 合并头信息
  options.header = Object.assign({}, defaultHeader, options.header)
  // 处理后的请求参数
  return options
}
```

```
async getUserProfile() {
  // 调用用户信息接口
  const {code, data: {avatar, nickName}} = await wx.http.get('/userInfo')
  // 检测是否调用成功
  if(code !== 10000) return wx.utils.toast()
  // 渲染数据
  this.setData({avatar, nickName})
},
```

享+生活 - 用户管理（头像和昵称）

- 接口路径：PUT /userInfo
- 接口参数：
  - nickName：用户昵称
- 响应数据：


```
{code: 10000, message: "操作成功", data: {avatar: null, id: "68631182
  code: 10000
  data: {avatar: null, id: "6863118240485376", nickName: "李四"}
    avatar: null
    id: "6863118240485376"
    nickName: "李四"
  message: "操作成功"
```


```
getUserNickname(ev) {
  // 更新用户昵称
  this.updateNickname(ev.detail.value)
},
async updateNickname(nickName) {
  // 调用更新昵称接口
  const {code} = await wx.http.put('/userInfo', {nickName})
  if(code !== 10000) return wx.utils.toast('更新昵称失败!')
  // 借助页面栈实例更新数据
  pageStack[0].setData({nickName});
}
```

享+生活 - 用户管理（头像和昵称）

- 文件上传：wx.uploadFile

```
// 基本语法
wx.uploadFile({
  url: '接口地址',
  filePath: '待上传文件的路径',
  name: '接收数据时的key（后端规定）',
  header: {}, // 头信息
  formData: {}, // 除上传文件外的其它数据
  success: () => {}
})
```

- 服务器域名：后台添加


```
getUserAvatar(ev) {
  // 更新用户头像
  this.updateAvatar(ev.detail.avatarUrl)
},
updateAvatar(avatar) {
  // 调用API上传文件
  wx.uploadFile({
    // 接口地址
    url: wx.http.baseURL + '/upload',
    // 待上传文件的路径
    filePath: avatar,
    name: 'file',
    header: {
      // 登录状态信息
      Authorization: 'Bearer ' + getApp().token
    },
    formData: {
      type: 'avatar'
    },
    success(result) {
      // 处理返回的数据
      const data  = JSON.parse(result.data)
      // 渲染头像
      pageStack[0].setData({avatar: data.data.url})
    }
  })
```

享+生活 - 用户管理（头像和昵称）

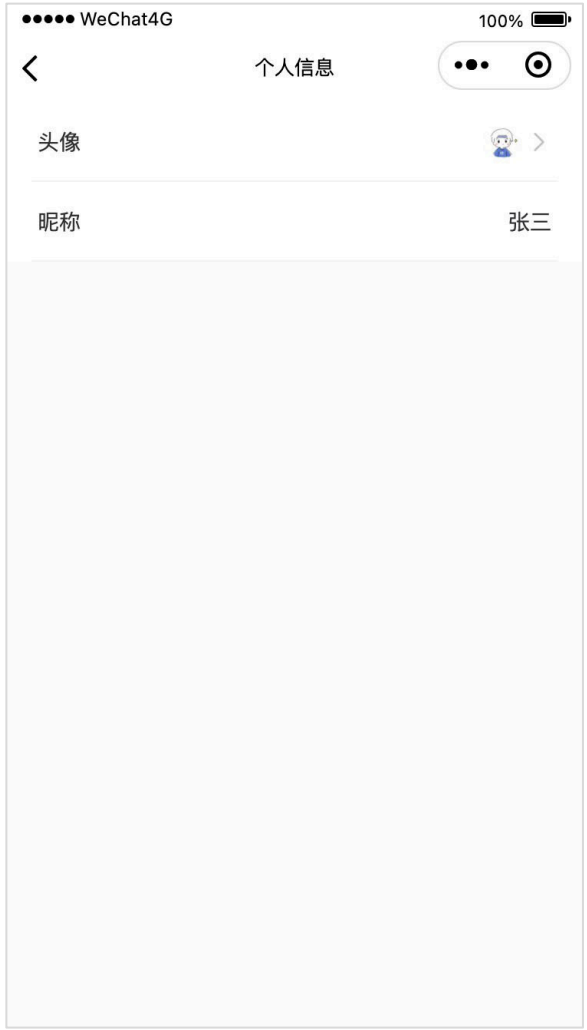
- 默认头像和昵称

```
// 存储到全局实例中
app.userProfile = {avatar, nickName}
```

```
onLoad() {
  // 获取应用实例
  const app = getApp()
  // 渲染头像和昵称
  this.setData({...app.userProfile})
},
```

- 修复上一小节小 bug

```
// 预览用户头像
this.setData({avatar: data.data.url})
```



传智教育旗下高端IT教育品牌