mpvue 项目(豆瓣图书)

尚硅谷前端研究院

版本：V 1.0

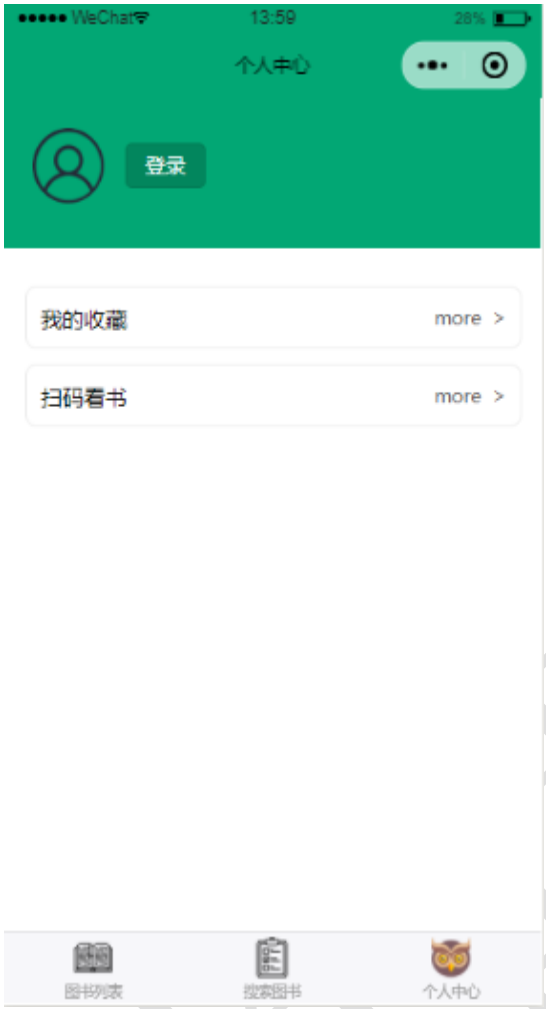
第 1 章：项目演示说明

1.1 页面演示

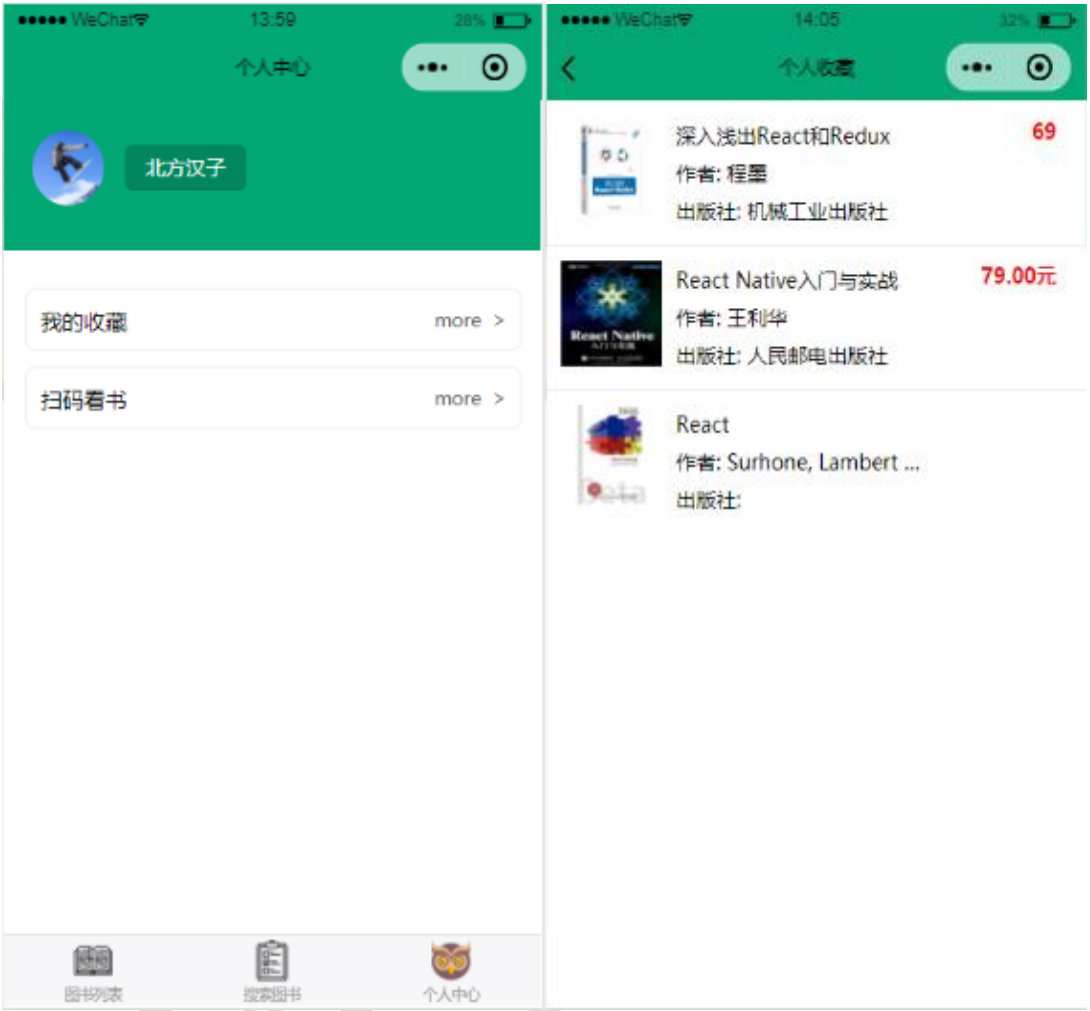
更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网



更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网



1.2 动态效果演示

豆瓣图书演示.gif

1.3 项目技术架构说明

1) 本项目为小程序项目，参考豆瓣图书板块创建

2) 技术选型: mpvue + 原生小程序 API + flyio + koa + 豆瓣开发接口

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

3) 本项目为模块化，组件化，工程化的项目

4) 使用 ES6 模块化

5) 使用 Stylus 作为 css 预编译语言

6) 使用 webpack 进项项目构建

7) 使用 eslint 进行语法检查

第 2 章：启动项目

2.1 初始化项目

1) npm install vue-cli -g 全局安装 vue 脚手架

2) vue init mpvue/mpvue-quickstart my-project 初始化 mpvue 项目

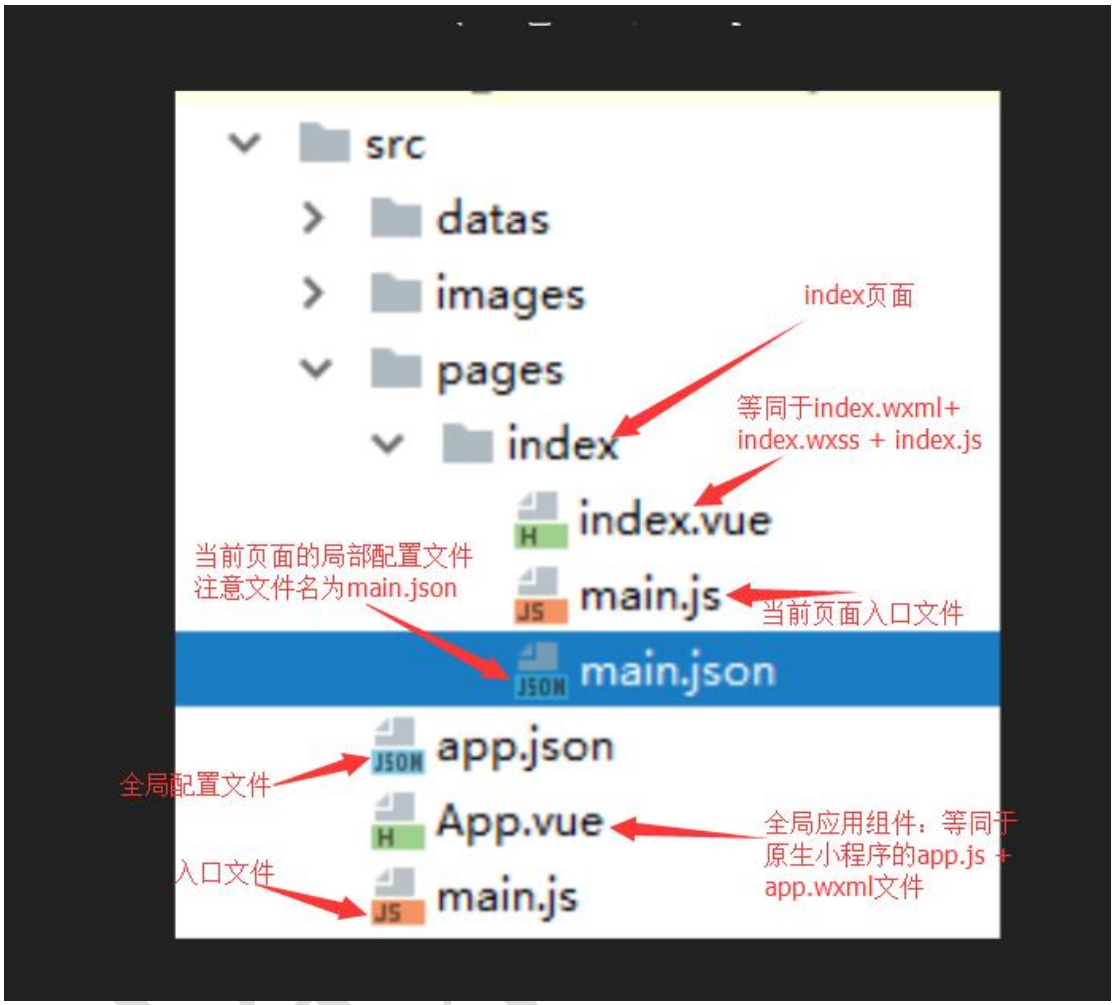
3) cd my-project 进入项目根目录

4) npm install 安装依赖

5) npm run dev || npm start 启动初始化项目

2.2初始化项目文件说明

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网



第 3 章：项目页面完成

3.1 主页

3.1.1 技术预览

1. swiper 滑屏组件(原生小程序提供 API)
2. v-for 遍历数据
3. wx.navigateTo()实现路由跳转(原生小程序提供 API)

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

3.1.2 页面结构

```
<div id="booksContainer">
  <swiper indicator-dots="true" indicator-color="pink"
indicator-active-color="#02a774">
    <swiper-item v-for="(item, key) in carousel" :key="index"
@click="toDetail(item)">
      <img :src="item.image" alt="" >
    </swiper-item>
  </swiper>
  <div id="AllBooks">
    <div class="nav">
      <span class="all">全部商品</span>
      <span class="more" @click="toBookList">&gt;</span>
    </div>
    <ul class="booksList">
      <li class="bookItem" v-for="(item, index) in booksList" :key="index"
@click="toDetail(item)">
        <img :src="item.image" alt="">
        <p class="bookName">
          《{{item.title}}》
        </p>
        <p class="bookDir">
          {{item.author}}
        </p>
      </li>
    </ul>
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
  </div>
</div>
```

3.1.3 页面样式

```
#booksContainer
  swiper
    width 100%
    height 400rpx
    img
      height 100%
      width 100%
  #AllBooks
    .nav
      padding 10rpx
    .more
      float right
      width 80rpx
      height 100%
      text-align right
      color #02a774
    .booksList
      display flex
      flex-wrap wrap
      .bookItem
        width 50%
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
        box-sizing border-box
        border-bottom 1rpx solid #eee
        text-align center
        padding 20rpx
        &:nth-child(2n + 1)
          border-right 1rpx solid #eee
          border-left 1rpx solid #eee
        img
          width 200rpx
          height 200rpx
        .bookName
          font-size 30rpx
          line-height 60rpx
        .bookDir
          font-size 28rpx
```

3.1.4 页面 js 处理逻辑

```
import datas from '../../../static/datas/data.json'
export default {
  data(){
    return {
      booksList: []
    }
  },
  mounted(){
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
    this.booksList = datas
  },
  computed: {
    carousel(){
      // 注意 splice 会修改原数组
      return [...this.booksList].splice(0, 4)
    }
  },
  methods: {
    toBookList(){
      wx.navigateTo({
        url: '/pages/bookList/main'
      })
    },
    toDetail(item){
      wx.navigateTo({
        url: '/pages/detail/main?bookItem=' + JSON.stringify(item)
      })
    }
  }
}
```

3.2 列表页

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

3.2.1 技术预览

1. v-for 遍历数据
2. wx.navigateTo()实现路由跳转(原生小程序提供 API)
3. 路由传参: url?key=value

3.2.2 页面结构

```
<div>
  <div class="bookItem" v-for="(item, index) in booksList" :key="index"
@click="toDetail(item)">
    <img :src="item.image" alt="">
    <div class="bookInfo">
      <span>{{item.title}}</span>
      <span>作者: {{item.author}}</span>
      <span>出版社: {{item.publisher}}</span>
    </div>
    <p class="price">{{item.price}}</p>
  </div>
</div>
```

3.2.3 页面样式

```
.bookItem
  display flex
  padding 10rpx
  border-bottom 1rpx solid #eee
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
  img
    width 140rpx
    height 140rpx
  .bookInfo
    display flex
    flex-direction column
    max-width 70%
    margin-left 10rpx
    line-height 50rpx
    font-size 28rpx
  .price
    position absolute
    right 30rpx
    color red
    font-size 32rpx
    font-weight bold
```

3.2.4 页面 js 处理逻辑

```
import datas from '../../../static/datas/data.json'
export default {
  data() {
    return {
        booksList: []
    }
  },
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
  mounted(){
    this.booksList = datas
  },
  methods: {
    toDetail(item){
      wx.navigateTo({
        url: '/pages/detail/main?bookItem=' + JSON.stringify(item)
      })
    }
  }
}
```

3.3 详情页

3.3.1 技术预览

1. 动态获取路由参数: this.$mp.query
2. wx.setNavigationBarTitle 动态设置窗口文字(原生小程序提供 API))

3.3.2 页面结构

```
<div id="detailContainer">
  <p class="bookName">{{bookItem.title}}</p>
  <img :src="bookItem.image" alt="">
  <div class="bookInfo">
    <span>作者: {{bookItem.author}}</span>
    <span>出版社: {{bookItem.publisher}}</span>
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
    <span>出版日期: {{bookItem.pubdate}}</span>
    <span>定价: {{bookItem.price}}</span>
  </div>
  <button>收藏该文章</button>
  <article class="authorInfo">
    <h1>作者简介</h1>
    <section>{{bookItem.author_intro}}</section>
  </article>
  <article class="contentInfo">
    <h1>文章简介</h1>
    <section>{{bookItem.summary}}</section>
  </article>
</div>
```

3.3.3 页面样式

```
#detailContainer
  display flex
  flex-direction column
  .bookName
    font-size 40rpx
    line-height 80rpx
    font-weight 700
    text-align center
  img
    width 70%
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
  height 700rpx
  margin 20rpx auto
.bookInfo
  display flex
  flex-direction column
  font-size 32rpx
  width 100%
  padding-left 15%
button
  width 300rpx
  height 60rpx
  font-size 32rpx
  line-height 60rpx
  margin 20rpx auto
article
  width 100%
  padding 10%
  box-sizing border-box
  h1
    text-align center
    color #02a774
    font-weight bold
  section
    font-size 34rpx
    text-indent 34rpx
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

3.3.4 页面 js 逻辑处理逻辑

```
export default {
  data(){
    return {
      bookItem: {}
    }
  },
  mounted(){
    // 更新 data 中的 bookItem 的数据
    this.bookItem = JSON.parse(this.$mp.query.bookItem)
    // 更新窗口标题文字
    wx.setNavigationBarTitle({
      title: this.bookItem.title
    })
  }
}
```

3.4 搜索图书页

3.4.1 技术预览

1. <input confirm-type='search'/> // 可以回车以后模拟手机搜索

3.4.2 页面结构

```
<div id="searchContainer">
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
  <div class="head">
    <input confirm-type="search" @confirm="confirm" v-model="searchContent"
class="searchInput" type="text" placeholder="书中自有黄金屋"
placeholder-class="placeholder">
    <span @click='searchContent = ""' v-show="isShow" class="clean">X</span>
  </div>

  <div class="bookListContainer" v-if="booksList.length">
    <p class="title">搜索到的图书共 20 本</p>
    <BookList :booksList="booksList"/>
  </div>
</div>
```

3.4.3 页面样式

```
#searchContainer
  .head
    position relative
    width 80%
    height 80rpx
    border-bottom 1rpx solid #02a774
    margin 0 auto
    .searchInput
      height 100%
    .placeholder
        color #02a774
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
      font-size 28rpx
      text-align center
    .clean
      position absolute
      right 20rpx
      top 20rpx
      z-index 99


  .bookListContainer
    &>p
      font-size 30rpx
      color #333
      padding 20rpx
      border 1rpx solid #eee
```

3.4.4 页面 js 逻辑处理逻辑

```
import request from '../../utils/request'
import config from '../../utils/config'


import BookList from '../bookList/bookList.vue'
export default {
  components: {
    BookList
  },
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
  data(){
    return {
      searchContent: '',
      isShow: false, // 标识清除内容按钮是否显示
      booksList: []
    }
  },
  watch: {
    searchContent(){
      this.isShow = this.searchContent.length?true : false
    }
  },
  methods: {
    async confirm(){
      console.log('确认搜索');
      // 发送请求获取图书列表，携带参数
      let url = '/searchBook?searchName=' + this.searchContent
      let result = await request(url)
      console.log(result);
      this.booksList = result.data.books
      console.log(this.booksList);
    }
  }
}
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

3.5 个人中心

3.5.1 获取用户登录 openid： App 中完成

```
wx.login({
  success: async (res) => {
    console.log(res); // {errMsg: "login:ok", code:
"011a7JmS1cFy541rmymS1KCQmS1a7Jm4"}
    let code = res.code // 用户登录的凭证，通过凭证可以获取用户的 openId
    let result = await request('/getOpenId?code=' + code)

    // 缓存用户 openID
    wx.setStorage({
      key: 'openID',
      data: result.openid
    })
  }
})
```

3.5.2 技术预览

1. wx.getUserInfo() 获取用户登录基本信息，但无法弹出授权窗口
2. <button open-type='getUserInfo'> 弹出授权弹窗
3. wx.scanCode() 扫码功能

3.5.3 页面结构

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
<div id="personalContainer">
  <div class="head">

<img :src="userInfo.avatarUrl?userInfo.avatarUrl:'/static/imgs/personal/pers
onal.png'" alt="">
    <button open-type="getUserInfo"
@getuserinfo="handleGetUserInfo">{{userInfo.nickName?userInfo.nickName:'登录
'}}</button>
  </div>
  <div class="cardList">
    <div class="card" @click="handleCollection">
      <span>我的收藏</span>
      <span class="more">more ></span>
    </div>
    <div class="card" @click="toScan">
      <span>扫码看书</span>
      <span class="more">more ></span>
    </div>
  </div>
</div>
```

3.5.4 页面样式

```
#personalContainer
  display flex
  flex-direction column
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
.head
  width 100%
  background #02a774
  padding 40rpx 0
  img
    width 100rpx
    height 100rpx
    vertical-align middle
    margin 0 40rpx
    border-radius 50rpx
  button
    display inline-block
    height 60rpx
    line-height 60rpx
    background rgba(0,0,0, .2)
    vertical-align middle
    color #fff
    font-size 28rpx


.cardList
  .card
    width 90%
    margin 20rpx auto
    border 1rpx solid #eeeeee
    padding 20rpx
    border-radius 10rpx
    font-size 30rpx
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
    .more
      float right
```

3.5.5 页面 js 处理逻辑

```
import request from '../../utils/request'
export default {
  data() {
    return {
      userInfo: {}
    }
  },
  mounted(){
    // 通过 wx 提供的 api 获取用户登录信息， 不能主动调起用户授权阶段，需要使用 button
    wx.getUserInfo({
      success: (msg) => {
        // 更新状态
        this.userInfo = msg.userInfo
      }
    })
  },
  methods: {
    // 用户点击 button 的时候会弹出授权窗口
    handleGetUserInfo(info){
      // 点击允许的时候 detail 中才有 userInfo 选项
      if(info.mp.detail.userInfo){
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
      // 更新状态
      this.userInfo = info.mp.detail.userInfo
       let openID = wx.getStorageSync('openID')
      wx.setStorage({
        key: openID,
        data: info.mp.detail.userInfo
       })
    }
  },
  toScan(){
    wx.navigateTo({
      url: '/pages/scan/main'
    })
  },
  async handleCollection(){
    // 查看用户是否登录
    let openID = wx.getStorageSync('openID')
    let userInfo = wx.getStorageSync(openID)
    if(userInfo){
      // 用户登录
      let result = await request('/getCollections', {openID})
      console.log(result, typeof result);
      wx.navigateTo({
        url: '/pages/bookList/main?booksList=' + JSON.stringify(result)
      })
    }else {
      wx.showToast({
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
          title: '请先登录',
          icon: "none"
        })
      }


    }
  }
}
```

3.6 详情页

3.6.1 技术预览

1. wx.scanCode() 扫码功能

3.6.2 页面结构

```
<div id="scan">
  <button @click="handleScan">扫码查看图书</button>
</div>
```

3.6.3 页面样式

3.6.4 页面 js 逻辑处理逻辑

```
import request from '../../utils/request'

export default {
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
  methods: {
    handleScan(){
      wx.scanCode({
        async success(res){
          let scanCode = res.result;
          let result = await request('/scanCode', {scanCode});
          wx.navigateTo({
            url: '/pages/detail/main?bookItem=' + JSON.stringify(result.data)
          })
        },
        fail(){
          console.log('扫码失败。。。');
        }
      })
    }
  }
}
```

第 4 章：服务器搭建

4.1 安装依赖

1) npm install koa
2) npm install koa-router
3) npm install flyio // 发送 ajax 请求

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

4) npm install nodemon // 自动更新 node 项目(文件变化时，不需要每次手动重启项目)

5) npm install koa-bodyparser --save // 获取 post 请求参数

4.2 注册应用

```
// 生成应用实例
let app = new Koa();
```

4.3 注册路由

```
// 生成路由器
let router = new KoaRouter();
// 注册路由
// 搜索图书路由
router.get('/searchBook', async (ctx, next) => {
  let searchName = ctx.query.searchName;
  let url = 'https://api.douban.com/v2/book/search?q=' + searchName;
  let response = await get(url);
  ctx.body = response;
})

// 声明使用路由器及允许使用路由方法
app
  .use(router.routes())
  .use(router.allowedMethods())
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

4.4 监听端口号

```
app.listen('4000', () => {
  console.log('服务器开启成功');
})
```

4.5 启动服务器

1. nodemon server.js
2. package.json 配置，配置后: npm run server

```
"scripts": {
  "server": "nodemon server.js"
}
```

4.6 路由一览表

```
//  搭建服务器的核心文件

let Koa = require('koa');
let KoaRouter = require('koa-router');
let Fly=require("flyio/src/node")


var jwt = require('jsonwebtoken');

let fly=new Fly;
// 1. 生成应用及路由器实例
const app = new Koa();
const router = new KoaRouter();


// 核心代码
router.get('/', (ctx, next) => {
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
  // 1. 获取请求的参数

  // 2. 根据请求的地址和参数处理数据

  // 3. 响应数据
  ctx.body = '服务器返回的数据 11111111111';
});

// 搜索图书的接口
let datas = require('./datas/data.json');
router.get('/searchBooks', (ctx, next) => {
  // 1. 获取请求的参数
  let req = ctx.query.req;
  console.log('获取请求参数： ', req);
  // 2. 根据请求的地址和参数处理数据
  let booksArr = datas;
  // 3. 响应数据
  ctx.body = booksArr;
});


// 获取用户 openId 的接口
router.get('/getOpenId', async (ctx, next) => {
  // 1. 获取请求的参数
  let code = ctx.query.code;
  let appId = 'wxc8c8e9c8a5a17f8f';
  let appSecret = 'e39bc25777c70399a2e27caaafbb7ef5';
  // 2. 根据请求的地址和参数处理数据
  let url =
`https://api.weixin.qq.com/sns/jscode2session?appid=${appId}&secret=${appSecret}&js
_code=${code}&grant_type=authorization_code`
  // 发送请求给微信接口，获取 openId
  let result = await fly.get(url);
  userInfo = JSON.parse(result.data)

  // 将用户的 openId 存入数据库， openId: {userName: 'xx', money: 'yyy'}

  // 自定义登录状态，就是根据用户的 openid 和 sessionKey 进行加密生成 token，返回给前端的
  // 对 openId 和 sessionKey 进行加密，自定义登录状态
  let token = jwt.sign(userInfo, 'atguigu');
  console.log(token);
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
  //
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzZXNzaW9uX2tleSI6IlN6WWJyNnBLUGN0WkFqMWJKUFk1ckE9PSIsIm9wZW5pZCI6Im9sblFfNWNyb0pfUXR5NTFx
  //
cktUQzktd1pKeVkiLCJpYXQiOjE1NzEyMjc2NjB9.lopVRU3qyb4q47mn21tJ8T5Z3pot4l5_zSf2-g-FGg
o
  // 3. 响应数据
  ctx.body = token;
});


// 测试验证身份 token 的接口
router.get('/test', (ctx, next) => {
  // 获取 token 的值
  console.log(ctx.request.header.authorization)
  let token = ctx.request.header.authorization;

  // { session_key: 'bvVTSxZf3pzi5yKpCwQSxA==',
  //   openid: 'olnQ_5croJ_Qty51qrKTC9-wZJyY',
  //   iat: 1571228656 } iat: 加密时的时间
  let result;
  try{
    result = jwt.verify(token, 'atguigu')
    console.log('验证结果', result);
    ctx.body = '验证成功'
  }catch (e) {
    ctx.body = '验证失败'
  }
});
// 2. 使用路由器及路由
app
  .use(router.routes()) // 声明使用路由
  .use(router.allowedMethods()) // 允许使用路由的方法

// 3. 监听端口
app.listen('3000', () => {
  console.log('服务器启动成功');
  console.log('服务器地址： http://localhost:3000');
})
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

第 5 章：获取用户凭证(openID)

5.1 提供当前用户 AppID，小程序密钥，登录 code

5.1.1 用户 AppID，小程序密钥获取方式

5.1.2 登录 code

```
wx.login({
  success: (res) => {
    console.log(res); // {errMsg: "login:ok", code:
"011a7JmS1cFy541rmymS1KCQmS1a7Jm4"}
    let code = res.code // 用户登录的凭证，通过凭证可以获取用户的 openId
  }
})
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

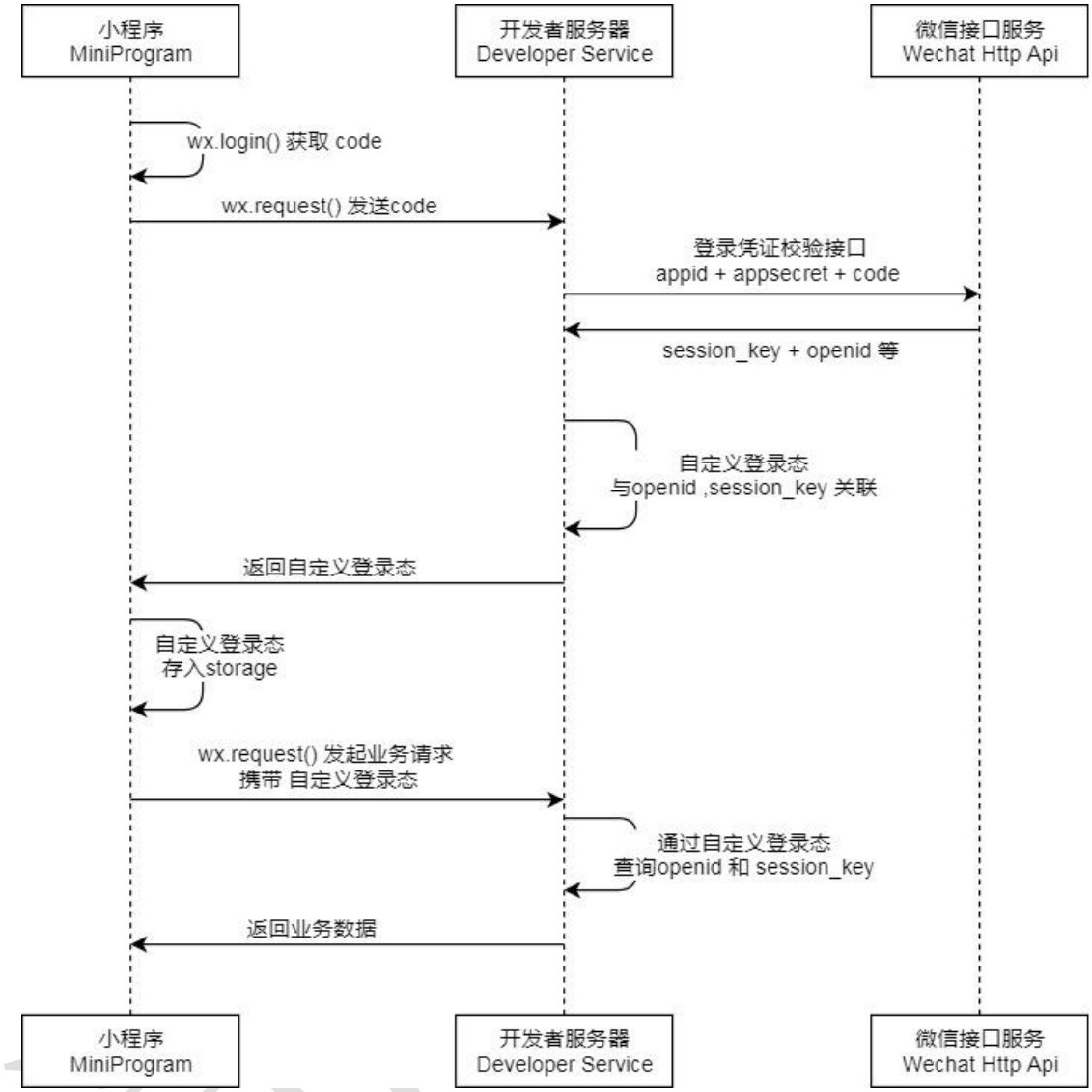
5.1.3 获取用户 openID

接口	https://api.weixin.qq.com/sns/jscode2session?appid=${appId}&secret=${appKey}&js_code=${code}&grant_type=authorization_code
参数	Appid， appkey， code

返回数据	{
	"session_key": "szCT2H6daI7hGFamVcmK9g==", // 会话密钥，对用户数据进行签名加密的密钥
	"openid": "ozByZ5RqVxvECZFaHC3JF8vVifUU" // 用户唯一标识
	}

5.2 官方建议登录流程

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网



第6章： 自定义登录状态 token 验证

6.1 jsonwebtoken 库使用

1. 下载： `npm install jsonwebtoken`

2. 生成 token

```
let token = jwt.sign(userInfo, 'atguigu')
```

3. 验证 token

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
jwt.verify(token, 'atguigu')
```

6.2 参考地址

https://github.com/auth0/node-jsonwebtoken

第 7 章：工具类库封装

7.1 浏览器端发送请求封装

```
let host = 'http://localhost:4000';

let config = {

  host

}


export default config
```

发送 ajax 请求封装

```
import config from './config'


export default (url, data='', method='GET') => {

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

    wx.request({

      url: config.host + url,

      method,

      data,

      success: (res) => {
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
        resolve(res.data)
      },
      fail: (error) => {
        reject(error)
      }
    })
  })
}
```

7.2 服务器端发送请求封装

```
let Fly = require('flyio/src/node');

let fly = new Fly;

// 暴露请求的方法
exports.get = function (url) {
  return new Promise((resolve, reject) => {
    fly.get(url)
      .then((response) => {
        console.log('请求成功');
        resolve(response);
      })
      .catch((error) => {
        reject(error);
```


更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网

```
    })
  })
}
```

尚硅谷

更多 Java –大数据 –前端 –python 人工智能资料下载，可访问百度：尚硅谷官网