

医疗陪诊小程序源代码说明

一、代码结构

1.1 目录结构

escort-end/ ├── miniprogram/ # 小程序代码 | ├── components/ # 自定义组件 | ├── images/ # 图片资源 |
| ├── pages/ # 页面文件 | ├── utils/ # 工具函数 | ├── app.js # 入口文件 | ├── app.json # 全局配置 |
app.wxss # 全局样式 ├── cloudfunctions/ # 云函数 | ├── login/ # 登录相关 | ├── orders/ # 订单相关 |
user/ # 用户相关

1.2 核心模块说明

1. 登录注册模块

avascript // pages/login/login.js // 实现微信登录、手机号获取等功能 Page({ data: { isAgree: false, loading:
false }, // 登录处理 handleLogin() { // 微信登录逻辑 }, // 注册处理 handleRegister() { // 注册逻辑 } })

2. 工作台模块

javascript // pages/workspace/workspace.js // 实现订单管理、状态切换等功能 Page({ data: { isOnline: false,
orderList: [], currentTab: 0 }, // 切换在线状态 toggleStatus() { // 状态切换逻辑 }, // 接单处理 handleOrder() {
// 接单逻辑 } })

3. 订单处理模块

avascript // cloudfunctions/orders/index.js // 实现订单相关的云函数 exports.main = async (event, context)
=> { const { type, orderId } = event switch(type) { case 'accept': // 接单处理 break case 'complete': // 完成订
单 break case 'cancel': // 取消订单 break } }

二、核心功能实现

2.1 用户认证

javascript // 微信登录 async function wxLogin() { const { code } = await wx.login() const { result } = await
wx.cloud.callFunction({ name: 'login', data: { code } }) return result } // 手机号获取 async function
getPhoneNumber(e) { const { cloudID } = e.detail const { result } = await wx.cloud.callFunction({ name:
'getPhoneNumber', data: { cloudID } }) return result }

2.2 订单管理

javascript // 订单状态处理 async function updateOrderStatus(orderId, status) { const { result } = await
wx.cloud.callFunction({ name: 'updateOrder', data: { orderId, status, updateTime: new Date() } }) return
result } // 订单列表获取 async function getOrderList(status) { const { result } = await wx.cloud.callFunction({
name: 'getOrders', data: { status } }) return result }

2.3 位置服务

```
javascript // 位置更新 async function updateLocation() { const location = await wx.getLocation() await
wx.cloud.callFunction({ name: 'updateLocation', data: location }) } // 距离计算 function
calculateDistance(lat1, lon1, lat2, lon2) { // 使用Haversine公式计算距离 }
```

三、数据库设计

3.1 数据集合

1. escorts (陪诊员信息) avascript { id: "string", // ID openid: "string", // 微信openid name: "string", // 姓名
phone: "string", // 手机号 status: "string", // 状态 createTime: "date" // 创建时间 }
2. orders (订单信息) avascript { id: "string", // 订单ID userId: "string", // 用户ID escortId: "string", // 陪诊
员ID status: "string", // 订单状态 createTime: "date" // 创建时间 }

3.2 索引设计

```
javascript // escorts集合索引 { "openid": 1, // 用户索引 "phone": 1, // 手机号索引 "status": 1 // 状态索引 } //
orders集合索引 { "userId": 1, // 用户索引 "escortId": 1, // 陪诊员索引 "status": 1, // 状态索引 "createTime": -1
// 时间索引 }
```

四、安全机制

4.1 数据安全

```
javascript // 数据库权限设置 { "read": true, // 所有用户可读 "write": "doc.openid == auth.openid" // 仅创建者
可写 } // 敏感数据加密 function encryptData(data) { // 数据加密逻辑 }
```

4.2 接口安全

```
javascript // 请求签名 function signRequest(params) { // 签名逻辑 } // 参数校验 function
validateParams(params) { // 参数验证逻辑 }
```

五、性能优化

5.1 缓存策略

```
javascript // 数据缓存 function cacheData(key, data) { wx.setStorageSync(key, { data, timestamp:
Date.now() }) } // 缓存检查 function checkCache(key) { const cache = wx.getStorageSync(key) if (!cache)
return null if (Date.now() - cache.timestamp > CACHE_TIME) return null return cache.data }
```

5.2 请求优化

```
avascript // 请求合并 async function batchRequest(requests) { // 合并请求逻辑 } // 请求限流 function
throttleRequest(fn, limit) { // 限流逻辑 }
```

六、部署说明

6.1 环境配置

```
javascript // 云环境配置 wx.cloud.init({ env: 'production-xxx', traceUser: true }) // 数据库初始化 async
function initDatabase() { // 数据库初始化逻辑 }
```

6.2 发布流程

- 1. 代码审查
- 2. 测试验证
- 3. 上传代码
- 4. 提交审核
- 5. 发布上线

七、更新记录

V1.0.0 (2024-03-20)

- 初始版本发布
- 实现基础功能
- 完成核心模块

用户端 |—— miniprogram/ # 小程序代码 | |—— components/ # 自定义组件 | |—— pages/ # 页面文件 | |—— images/ # 图片资源 | |—— utils/ # 工具函数 | |—— app.js # 入口文件 | |—— app.json # 全局配置 | |—— app.wxss # 全局样式 |—— cloudfunctions/ # 云函数 |—— login/ # 登录相关 |—— orders/ # 订单相关 |—— user/ # 用户相关

1.2 核心文件说明

- 1. 小程序端 javascript // app.js - 小程序入口 App({ onLaunch: function () { // 初始化云开发 wx.cloud.init({ env: 'cloud-env-id', traceUser: true }) }, globalData: { userInfo: null, isLogin: false } })
- 2. 云函数 avascript // login/index.js - 登录云函数 exports.main = async (event, context) => { const { OPENID } = cloud.getWXContext() // 处理登录逻辑 return { success: true, data: userInfo } }

二、核心模块说明

2.1 登录注册模块

- 1. 登录页面 avascript // pages/login/login.js Page({ // 微信登录 handleLogin() { wx.cloud.callFunction({ name: 'login', success: res => { // 处理登录结果 } }) } })
- 2. 注册页面 javascript // pages/register/register.js Page({ // 提交注册 submitForm() { wx.cloud.callFunction({ name: 'submitRegistration', data: this.data.formData, success: res => { // 处理注册结果 } }) } })

2.2 工作台模块

- 1. 工作台页面 javascript // pages/workspace/workspace.js Page({ // 切换在线状态 toggleOnline() { wx.cloud.callFunction({ name: 'updateStatus', data: { isOnline: true } }) }, // 接单处理 acceptOrder(orderId) { wx.cloud.callFunction({ name: 'acceptOrder', data: { orderId } }) } })
- 2. 订单详情 javascript // pages/order-detail/order-detail.js Page({ // 更新订单状态 updateOrderStatus(status) { wx.cloud.callFunction({ name: 'updateOrderStatus', data: { orderId: this.data.orderId, status } }) } })

2.3 收入模块

1. 收入统计 `avascript // pages/income/income.js Page({ // 获取收入数据 getIncomeStats() {
 wx.cloud.callFunction({ name: 'getIncomeStats', success: res => { this.setData({ stats: res.result.data
 }) }) }) })`
2. 提现功能 `avascript // pages/withdraw/withdraw.js Page({ // 提现请求 requestWithdraw() {
 wx.cloud.callFunction({ name: 'withdraw', data: { amount: this.data.amount } }) })`

三、数据库设计

3.1 数据集合

1. escorts (陪诊员信息) `javascript { id: "string", // ID openid: "string", // 微信openid name: "string", // 姓名 phone: "string", // 手机号 status: "string", // 状态 createTime: "Date" // 创建时间 }`
2. orders (订单信息) `avascript { id: "string", // 订单ID escortId: "string", // 陪诊员ID status: "string", // 订单状态 amount: "number", // 订单金额 createTime: "Date" // 创建时间 }`

3.2 索引设计

1. escorts集合
 - openid: 唯一索引
 - phone: 唯一索引
 - status: 普通索引
2. orders集合
 - escortId: 普通索引
 - status: 普通索引
 - createTime: 普通索引

四、接口说明

4.1 云函数接口

1. login `avascript // 请求参数 { userType: "escort" } // 返回数据 { success: true, data: { userInfo: Object } }`
2. acceptOrder `javascript // 请求参数 { orderId: "string" } // 返回数据 { success: true, data: { orderInfo: Object } }`

4.2 数据库操作

1. 查询操作 `avascript // 获取陪诊员信息 db.collection('escorts') .where({ openid: openid }) .get()`
2. 更新操作 `javascript // 更新订单状态 db.collection('orders') .doc(orderId) .update({ data: { status: 'ACCEPTED' } })`

五、工具函数

5.1 通用工具

1. 时间格式化 javascript // utils/format.js export const formatTime = date => { const year = date.getFullYear() const month = date.getMonth() + 1 const day = date.getDate() return `\${year}-\${month}-\${day} } }
2. 数据验证 javascript // utils/validate.js export const validatePhone = phone => { return /^1[3-9]\d{9}\$/.test(phone) }

5.2 业务工具

1. 订单状态处理 javascript // utils/order.js export const getOrderStatusText = status => { const statusMap = { PENDING: '待接单', ACCEPTED: '已接单', COMPLETED: '已完成' } return statusMap[status] }

六、组件说明

6.1 自定义组件

1. 订单卡片 javascript // components/order-card/order-card.js Component({ properties: { order: Object }, methods: { handleAccept() { this.triggerEvent('accept') } } })

6.2 页面组件

1. 工作台页面 javascript // pages/workspace/workspace.js Page({ data: { orders: [], isOnline: false }, onLoad() { this.loadOrders() } })

七、部署说明

7.1 环境配置

1. 开发环境
 - 微信开发者工具
 - Node.js环境
 - 云开发环境
2. 生产环境
 - 云开发环境ID
 - 小程序AppID
 - 域名配置

7.2 部署步骤

1. 云函数部署
 - 安装依赖
 - 上传函数
 - 配置触发器
2. 数据库初始化
 - 创建集合


- 设置权限
- 创建索引

8 源码

{{ORDER_STATUS_TEXT[order.status]}} 订单号: {{order.orderNo}}
{{formatTime(order.statusInfo.PAIDTime)}} 订单支付 {{formatTime(order.statusInfo.ACCEPTEDTime)}} 接单
{{formatTime(order.statusInfo.ARRIVEDTime)}} 到达医院
{{formatTime(order.statusInfo.PROCESSINGTime)}} 开始服务
{{formatTime(order.statusInfo.COMPLETEDTime)}} 完成服务 就医信息 就医人 {{order.patientInfo.name}} 联系电话 {{order.patientInfo.phone}} ☐ 就医医院 {{order.hospitalInfo.hospitalName}} 就诊科室
{{order.hospitalInfo.departmentName}} 预约时间 {{order.serviceInfo.appointmentDate}}
{{order.serviceInfo.timeSlot}}  {{order.hospitalInfo.address}} 距您{{distance}}km 服务信息 服务类型
{{order.serviceInfo.serviceType}} 预计时长 {{order.serviceInfo.duration}}小时 特殊要求
{{order.serviceInfo.requirements}} 服务记录 {{formatTime(item.time)}} {{item.content}}  订单金额 陪诊费用 ¥{{order.paymentInfo.escortAmount}} 保险费用 ¥{{order.paymentInfo.insuranceAmount}} 平台服务费 ¥
{{order.paymentInfo.platformFee}} 合计金额 ¥{{order.paymentInfo.totalAmount}} 接单 到达医院 开始服务
添加记录 完成服务  提交成功 您的注册申请已提交, 我们将在1-3个工作日内完成审核

```
<view class="info-list">
  <view class="info-item">
    <text class="label">审核进度查询</text>
    <text class="value">请前往"我的-注册状态"查看</text>
  </view>
  <view class="info-item">
    <text class="label">审核通过通知</text>
    <text class="value">我们将通过短信通知您审核结果</text>
  </view>
  <view class="info-item">
    <text class="label">客服咨询</text>
    <text class="value">工作日 9:00-18:00</text>
  </view>
</view>

<view class="btn-group">
  <button class="primary-btn" bindtap="checkStatus">查看审核状态</button>
  <button class="outline-btn" open-type="contact">联系客服</button>
</view>
```

 温馨提示 · 请确保提交的证件照片清晰完整 · 审核通过后即可开始接单 · 如有疑问请联系在线客服
{{item}}

```
<scroll-view class="training-list" scroll-y refresher-enabled refresher-triggered="{{refreshing}}"
bindrefresherrefresh="onRefresh" bindscrolltolower="onLoadMore">
```

```
<view class="training-item" wx:for="{{trainings}}" wx:key="_id">
  <view class="training-header">
```

```
<text class="title">{{item.title}}</text>
<text class="status {{item.isCompleted ? 'completed' : ''}}">
  {{item.isCompleted ? '已完成' : '待学习'}}
</text>
</view>


<view class="training-content">
  <view class="info-row">
    <text class="label">培训类型: </text>
    <text class="value">{{item.type}}</text>
  </view>
  <view class="info-row">
    <text class="label">培训时长: </text>
    <text class="value">{{item.duration}}分钟</text>
  </view>
  <view class="info-row">
    <text class="label">开始时间: </text>
    <text class="value">{{item.startTime}}</text>
  </view>
  <view class="info-row">
    <text class="label">截止时间: </text>
    <text class="value">{{item.endTime}}</text>
  </view>
</view>

<view class="training-footer">
  <view class="score" wx:if="{{item.isCompleted}}">
    <text class="label">考试得分: </text>
    <text class="value {{item.score >= 60 ? 'pass' : 'fail'}}">
{{item.score}}分</text>
  </view>
  <button
    class="start-btn {{item.isCompleted ? 'disabled' : ''}}
    bindtap="startTraining"
    data-id="{{item._id}}"
    disabled="{{item.isCompleted}}"
  >
    {{item.isCompleted ? '已完成' : '开始学习'}}
  </button>
</view>
</view>

<!-- 空状态 -->
<view class="empty-tip" wx:if="{{trainings.length === 0}}">
  <image src="/images/empty.png"></image>
  <text>暂无{{currentTab === 0 ? '待学习' : '已完成'}}的培训</text>
</view>

<!-- 加载更多 -->
<view class="loading-more" wx:if="{{hasMore}}">
```

```
<text>加载中...</text>
</view>
```

 {{escortInfo.name}} 评分 {{escortInfo.rating || '5.0'}} 接单状态 {{todayOrders || 0}} 今日订单 {{monthOrders || 0}} 本月订单 {{totalOrders || 0}} 总订单 {{item}} {{newOrders.length}}

```
<scroll-view class="order-list" scroll-y refresher-enabled refresher-triggered="{{refreshing}}"
bindrefresherrefresh="onRefresh" bindscrolltolower="onLoadMore">
```

```
<!-- 新订单 -->
<block wx:if="{{currentTab === 0}}">
  <view class="order-item" wx:for="{{newOrders}}" wx:key="_id">
    <view class="order-header">
      <text class="order-no">订单号: {{item.orderNo}}</text>
      <text class="time">{{item.createTime}}</text>
    </view>
    <view class="order-content">
      <view class="hospital-info">
        <text class="hospital">{{item.hospitalName}}</text>
        <text class="department">{{item.departmentName}}</text>
      </view>
      <view class="service-info">
        <text class="service-type">{{item.serviceType}}</text>
        <text class="price">¥{{item.price}}</text>
      </view>
      <view class="patient-info">
        <text class="label">就医人: </text>
        <text>{{item.patientName}}</text>
      </view>
      <view class="appointment-info">
        <text class="label">就医时间: </text>
        <text>{{item.appointmentDate}} {{item.timeSlot}}</text>
      </view>
    </view>
    <view class="order-footer">
      <button
        class="accept-btn"
        bindtap="acceptOrder"
        data-id="{{item._id}}"
      >接单</button>
    </view>
  </view>
</block>

<view class="empty-tip" wx:if="{{newOrders.length === 0}}">
  <image src="/images/empty.png"></image>
  <text>暂无新订单</text>
</view>

<!-- 进行中订单 -->
```



```
<block wx:if="{{currentTab === 1}}">
  <view class="order-item" wx:for="{{processingOrders}}" wx:key="_id">
    <view class="order-header">
      <text class="order-no">订单号: {{item.orderNo}}</text>
      <text class="status">进行中</text>
    </view>
    <view class="order-content">
      <!-- 订单内容同上 -->
    </view>
    <view class="order-footer">
      <button
        class="complete-btn"
        bindtap="completeOrder"
        data-id="{{item._id}}"
      >完成服务</button>
    </view>
  </view>
  <view class="empty-tip" wx:if="{{processingOrders.length === 0}}">
    <image src="/images/empty.png"></image>
    <text>暂无进行中订单</text>
  </view>
</block>

<!-- 历史订单 -->
<block wx:if="{{currentTab === 2}}">
  <view class="order-item" wx:for="{{historyOrders}}" wx:key="_id">
    <view class="order-header">
      <text class="order-no">订单号: {{item.orderNo}}</text>
      <text class="status">已完成</text>
    </view>
    <view class="order-content">
      <!-- 订单内容同上 -->
    </view>
    <view class="order-footer">
      <text class="complete-time">完成时间: {{item.completeTime}}</text>
    </view>
  </view>
  <view class="empty-tip" wx:if="{{historyOrders.length === 0}}">
    <image src="/images/empty.png"></image>
    <text>暂无历史订单</text>
  </view>
</block>

<!-- 加载更多 -->
<view class="loading-more" wx:if="{{hasMore}}">
  <text>加载中...</text>
</view>
```

总收入(元) {{totalIncome || '0.00'}} {{monthlyIncome || '0.00'}} 本月收入 {{orderCount || '0'}} 服务订单
{{rating || '0.0'}} 综合评分

```
<scroll-view class="income-list" scroll-y refresher-enabled refresher-triggered="{{refreshing}}"
bindrefresherrefresh="onRefresh" bindscrolltolower="onLoadMore"
```

```
<view class="list-header">
  <text class="title">收入明细</text>
  <picker
    mode="date"
    fields="month"
    value="{{currentMonth}}"
    bindchange="onMonthChange"
  >
    <view class="month-picker">
      <text>{{currentMonth}}</text>
      <text class="icon icon-calendar"></text>
    </view>
  </picker>
</view>

<view class="income-item" wx:for="{{incomeList}}" wx:key="_id">
  <view class="order-info">
    <text class="order-no">订单号: {{item.orderNo}}</text>
    <text class="income" {{item.status === 'PENDING' ? 'pending' : ''}}">
      {{item.status === 'PENDING' ? '待结算' : '+' + item.amount}}
    </text>
  </view>
  <view class="service-info">
    <text class="hospital">{{item.hospitalName}}</text>
    <text class="department">{{item.departmentName}}</text>
  </view>
  <view class="time-info">
    <text class="time">{{item.createTime}}</text>
    <text class="status">{{item.statusText}}</text>
  </view>
</view>

<!-- 加载更多 -->
<view class="loading-more" wx:if="{{hasMore}}">
  <text>加载中...</text>
</view>

<!-- 无数据提示 -->
<view class="empty-tip" wx:if="{{incomeList.length === 0}}">
  <image src="/images/empty.png"></image>
  <text>暂无收入记录</text>
</view>
```

提现 const app = getApp() const db = wx.cloud.database()

```
Page({ data: { totalIncome: '0.00', monthlyIncome: '0.00', orderCount: 0, rating: '0.0', currentMonth: '',
incomeList: [], refreshing: false, hasMore: false, page: 1, pageSize: 20 },
```

```
onLoad() { // 设置当前月份 const now = new Date() this.setData({ currentMonth:
`${now.getFullYear()}-${String(now.getMonth() + 1).padStart(2, '0')}})
```

```
    this.loadIncomeStats()
    this.loadIncomeList()
```

```
  },

  // 加载收入统计 async loadIncomeStats() { try { const { result } = await wx.cloud.callFunction({ name:
'getIncomeStats' })
```

```
    if (result.success) {
      this.setData({
        totalIncome: result.data.totalIncome.toFixed(2),
        monthlyIncome: result.data.monthlyIncome.toFixed(2),
        orderCount: result.data.orderCount,
        rating: result.data.rating.toFixed(1)
      })
    }
  } catch (err) {
    console.error('加载收入统计失败', err)
    wx.showToast({
      title: '加载失败',
      icon: 'none'
    })
  }
}
```

```
  },

  // 加载收入明细 async loadIncomeList(append = false) { try { const { result } = await wx.cloud.callFunction({
name: 'getIncomeList', data: { month: this.data.currentMonth, page: this.data.page, pageSize:
this.data.pageSize } })
```

```
    if (result.success) {
      const list = result.data.list.map(item => ({
        ...item,
        createTime: this.formatTime(item.createTime),
        amount: item.amount.toFixed(2),
        statusText: this.getStatusText(item.status)
      }))

      this.setData({
        incomeList: append ? [...this.data.incomeList, ...list] : list,
```

```
      hasMore: list.length === this.data.pageSize
    })
  }
} catch (err) {
  console.error('加载收入明细失败', err)
  wx.showToast({
    title: '加载失败',
    icon: 'none'
  })
}
```

},

// 下拉刷新 async onRefresh() { this.setData({ refreshing: true, page: 1 })

```
try {
  await Promise.all([
    this.loadIncomeStats(),
    this.loadIncomeList()
  ])
} finally {
  this.setData({
    refreshing: false
  })
}
```

},

// 加载更多 async onLoadMore() { if (!this.data.hasMore || this.data.refreshing) return

```
this.setData({
  page: this.data.page + 1
})

await this.loadIncomeList(true)
```

},

// 切换月份 onMonthChange(e) { this.setData({ currentMonth: e.detail.value, page: 1 }) this.loadIncomeList()
},

// 显示提现弹窗 showWithdrawModal() { wx.showModal({ title: '提现金额', editable: true, placeholderText: '请输入提现金额', success: (res) => { if (res.confirm) { const amount = parseFloat(res.content) if (isNaN(amount) || amount <= 0) { wx.showToast({ title: '请输入正确金额', icon: 'none' }) return } this.requestWithdraw(amount) } } }) },

// 申请提现 requestWithdraw(amount) { wx.showLoading({ title: '处理中...' })

```
wx.cloud.callFunction({
  name: 'escortWithdraw',
  data: { amount }
}).then(res => {
  if (res.result.success) {
    wx.showToast({
      title: '提现申请已提交',
      icon: 'success'
    })
    // 刷新数据
    this.loadIncomeData()
  } else {
    throw new Error(res.result.error)
  }
}).catch(err => {
  wx.showToast({
    title: err.message || '提现失败',
    icon: 'none'
  })
}).finally(() => {
  wx.hideLoading()
})
```

```
},
```

```
// 格式化时间 formatTime(timestamp) { const date = new Date(timestamp) return
`${date.getFullYear()}-${String(date.getMonth() + 1).padStart(2,
'0')}-${String(date.getDate()).padStart(2, '0')}`
`${String(date.getHours()).padStart(2,
'0')}:${String(date.getMinutes()).padStart(2, '0')}`,
```

```
// 获取状态文本 getStatusText(status) { const statusMap = { 'PENDING': '待结算', 'COMPLETED': '已结算',
'PROCESSING': '结算中', 'FAILED': '结算失败' } return statusMap[status] || status } }) Page({ data: {
registrationId: '' // 注册申请ID },
```

```
onLoad(options) { if (options.id) { this.setData({ registrationId: options.id }) } },
```

```
// 查看审核状态 checkStatus() { wx.navigateTo({ url: '/pages/register-status/register-status' }) },
```

```
// 返回首页 goHome() { wx.switchTab({ url: '/pages/index/index' }) },
```

```
// 防止用户返回到注册页 onUnload() { const pages = getCurrentPages() if (pages.length > 1) { const
prePage = pages[pages.length - 2] if (prePage.route.includes('register')) { wx.switchTab({ url:
'/pages/index/index' }) } } } }) const app = getApp()
```

```
Page({ data: { currentTab: 0, tabs: ['待学习', '已完成'], trainings: [], refreshing: false, hasMore: false, page: 1,
pageSize: 20 },
```

```
onLoad() { this.loadTrainings() },
```

```
// 切换标签 switchTab(e) { const tab = e.currentTarget.dataset.tab this.setData({ currentTab: parseInt(tab),
page: 1 }, () => { this.loadTrainings() }) },

// 加载培训列表 async loadTrainings(append = false) { const { currentTab, page, pageSize } = this.data const
status = currentTab === 0 ? 'upcoming' : 'completed'
```

```
try {
  const { result } = await wx.cloud.callFunction({
    name: 'getTrainings',
    data: { status, page, pageSize }
  })

  if (result.success) {
    this.setData({
      trainings: append ? [...this.data.trainings, ...result.data.list] :
result.data.list,
      hasMore: result.data.list.length === pageSize
    })
  }
} catch (err) {
  console.error('加载培训列表失败', err)
  wx.showToast({
    title: '加载失败',
    icon: 'none'
  })
}
```

```
},
```

```
// 下拉刷新 async onRefresh() { this.setData({ refreshing: true, page: 1 })
```

```
await this.loadTrainings()
this.setData({ refreshing: false })
```

```
},
```

```
// 加载更多 async onLoadMore() { if (!this.data.hasMore || this.data.refreshing) return
```

```
this.setData({
  page: this.data.page + 1
})
await this.loadTrainings(true)
```

```
},
```

```
// 开始培训 startTraining(e) { const id = e.currentTarget.dataset.id wx.navigateTo({ url: /pages/training-  
detail/training-detail?id=${id} }) }) const app = getApp()
```

```
Page({ data: { training: null, currentSection: 0, progress: 0 },
```

```
onLoad(options) { if (options.id) { this.loadTrainingDetail(options.id) } },
```

```
// 加载培训详情 async loadTrainingDetail(id) { try { const { result } = await wx.cloud.callFunction({ name:  
'getTrainingDetail', data: { id } })
```

```
    if (result.success) {  
      this.setData({  
        training: result.data  
      })  
    }  
  } catch (err) {  
    console.error('加载培训详情失败', err)  
    wx.showToast({  
      title: '加载失败',  
      icon: 'none'  
    })  
  }  
}
```

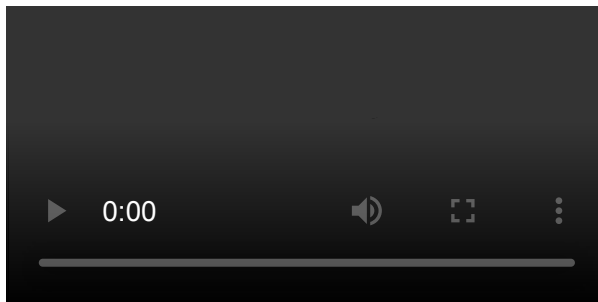
```
},
```

```
// 切换章节 switchSection(e) { const index = e.currentTarget.dataset.index this.setData({ currentSection:  
index }) },
```

```
// 更新学习进度 updateProgress() { const progress = Math.min( ((this.data.currentSection + 1) /  
this.data.training.sections.length) * 100, 100 ) this.setData({ progress }) },
```

```
// 开始考试 startExam() { wx.navigateTo({ url: /pages/training-exam/training-exam?  
id=${this.data.training._id} }) })
```

{{training.title}} 培训类型: {{training.type}} 培训时长: {{training.duration}}分钟 开始时间:
{{training.startTime}} 截止时间: {{training.endTime}} 学习进度 {{progress}}% 课程章节 {{item.name}}



{{item.duration}}分钟

{{progress < 100 ? '请先完成学习' : '开始考试'}}