

# uniapp06

## 学子商城App

我的页面

按照效果图,完成我的页面 登录页面 表单,按钮的布局

- css布局方式 flex布局 弹性盒子
- 背景图 模糊效果
- 显示密码,密文密码 input type属性password密文 text明文.
  - 通过点击小眼睛 切换type属性 type属性值是一个变量
  - 图片眼睛 分为 睁开和闭眼 眼睛的图片src也是一个变量

点击的状态为存为data

两个属性 input type属性和图片src属性 做成计算属性 通过点击状态来进行修改.

/pages/user/user.vue

```
<template>
  <view class="content">
    <!-- 背景图 -->
    <image class="img-bg" src="/static/images/pikaqiu.jpeg"></image>
    <!-- logo图片 -->
    <image class="img-logo" src="/static/icons/logo.png"></image>
    <!-- 表单区域 -->
    <!-- 用户名 -->
    <view class="input-area">
      <image src="/static/images/user.png"></image>
      <input type="text" value="" placeholder="用户名/手机号" />
    </view>
    <!-- 密码 -->
    <view class="input-area-pwd">
      <image src="/static/images/mima.png"></image>
      <input :type="pwd_type" value="" placeholder="密码" />
      <image
        @click="pwd_status = !pwd_status"
        :src="pwd_icon"
      ></image>
    </view>
    <!-- 登录按钮 -->
    <view class="btn-login">登录</view>
    <!-- 账户操作区域 -->
    <view class="account-area">
      <view>忘记密码?</view>
      <view>立即注册</view>
    </view>
    <!-- 底部按钮 第三方登录 -->
```

```

    <view class="bottom">
      <!-- 横线和第三方的登录 -->
      <view>
        <view class="line"></view>
        <view>第三方直接登录</view>
        <view class="line"></view>
      </view>
      <!-- 图标 -->
      <view>
        <view><image src="/static/images/qq.jpg" mode=""></image></view>
        <view><image src="/static/images/wx.jpg" mode=""></image></view>
        <view><image src="/static/images/wb.jpg" mode=""></image></view>
      </view>
    </view>
  </view>
</template>

<script>
export default {
  data() {
    return {
      // 是否显示密码的状态 true 为显示 false为不显示
      // 通过点击修改状态
      pwd_status: false
    };
  },
  computed: {
    // 文本显示状态 明文 密文
    pwd_type() {
      return this.pwd_status ? 'text' : 'password';
    },
    // 眼睛图片的src属性
    pwd_icon(){
      return this.pwd_status? '/static/images/eye.png': '/static/images/eye-close.png'
    }
  },
  methods: {}
};
</script>

<style lang="scss">
// 底部第三方登录
.bottom {
  position: fixed;
  bottom: 150rpx;
  left: 0;
  right: 0;
  padding: 0 100rpx;
  // 第三登录的文字
  > view:first-child {
    color: white;
    display: flex;
    align-items: center;
  }
}

```

```

    > .line {
        height: 1px;
        background-color: white;
        width: 150rpx;
    }
}
> view:last-child {
    margin-top: 30rpx;
    display: flex;
    justify-content: space-around;
    // 图标的view
    > view {
        padding: 10rpx;
        border-radius: 50%;
        overflow: hidden;
        background-color: white;
    }
    // 图片大小
    image {
        width: 80rpx;
        height: 80rpx;
    }
}
}
// 账户操作区域
.account-area {
    z-index: 1;
    margin-top: 30rpx;
    display: flex;
    justify-content: space-between;
    view {
        color: white;
        text-decoration: underline;
    }
}
// 登录按钮
.btn-login {
    border-radius: 10rpx;
    z-index: 1;
    margin-top: 30rpx;
    padding: 10rpx;
    width: 100%;
    background-color: orange;
    color: white;
    text-align: center;
}
// 密码
.input-area-pwd {
    border-radius: 10rpx;
    padding: 15rpx;
    z-index: 1;
    margin-top: 30rpx;
    background-color: white;
}

```

```

    color: gray;
    display: flex;
    // flex-direction: row;
    align-item: center;
    > image {
        width: 50rpx;
        height: 50rpx;
        margin: 0rpx 20rpx;
    }
}
// 用户名
.input-area {
    border-radius: 10rpx;
    padding: 15rpx;
    z-index: 1;
    margin-top: 100rpx;
    background-color: white;
    color: gray;
    display: flex;
    // flex-direction: row;
    align-item: center;
    > image {
        width: 50rpx;
        height: 50rpx;
        margin: 0rpx 20rpx;
    }
}
// logo
.img-logo {
    // logo 大小
    // rpx 微信小程序 把iphone6 屏幕宽度 分为750份 1rpx就是一份
    width: 400rpx;
    height: 100rpx;
    // 对齐方式 主轴自居中
    align-self: center;
    // 背景色
    background-color: white;
    // 填充
    padding: 5rpx 20rpx;
    // 圆角
    border-radius: 10rpx;
}
// 背景图
.img-bg {
    // 浮动
    position: fixed;
    // 调整位置
    left: 0;
    top: 0;
    // 拉伸
    width: 100%;
    height: 100%;
    // z-index 设置层

```

```

    z-index: 0;
    // 模糊
    filter: blur(2px);
  }
  // 整个内容
  .content {
    padding: 100rpx;
    // flex 布局
    display: flex;
    // 排列方式 竖向 垂直 column
    flex-direction: column;
  }
</style>

```

## 首页

使用到的组件:

- 通告栏
- 轮播图
- icon小图标
- 宫格布局 双列
- 底部图片
- 回到顶部

接口地址:

<http://www.codeboy.com:9999/data/product/index.php>

- ▶ carouselItems: [...] 轮播图
- ▶ recommendedItems: [...] 1F 首页推荐
- ▶ newArrivalItems: [...] 2F 最新上架
- ▶ topSaleItems: [...] 3F 热销单品

/pages/home/home.vue

```

<template>
  <view v-if="data">
    <!-- 通告栏 -->
    <uni-notice-bar
      style="margin: 0;"
      single
      text="基于uniapp开发的学子商城App即将完成,欢迎大家来试用! by WEB2107班级"
      scrollable
      showIcon
    >

```

```

        showClose
        :speed="30"
    ></uni-notice-bar>
    <!-- 轮播图 -->
    <swiper
        circular
        indicator-dots
        autoplay
        :interval="3000"
        :duration="1000"
    >
        <swiper-item v-for="(item, index) in carouselItems" :key="index">
            <image
                :src="baseUrl + item.img"
                style="width: 100%; height: 100%;"
            ></image>
        </swiper-item>
    </swiper>
    <!-- 楼层 商品列表 -->
    <view v-for="(item, index) in items">
        <!-- 1F 首页推荐 header -->
        <view class="header">
            <uni-icons type="shop" size="20"></uni-icons>
            <text>{{ item.title }}</text>
        </view>
        <!-- 商品列表 -->
        <uni-grid
            :column="2"
            :square="false"
            :showBorder="false"
            :highlight="false"
        >
            <uni-grid-item v-for="(item1, index) in item.data" :key="index">
                <view class="cell">
                    <!-- 图片 -->
                    <image :src="baseUrl + item1.pic"></image>
                    <!-- 商品名称 -->
                    <view>{{ item1.title }}</view>
                    <!-- 商品详情 -->
                    <view>{{ item1.details }}</view>
                    <!-- 价格 -->
                    <view>¥{{ item1.price }}</view>
                    <!-- 查看详情按钮 -->
                    <button
                        size="mini"
                        style="background-color:#0077ff;color:white"
                    >
                        查看详情
                    </button>
                </view>
            </uni-grid-item>
        </uni-grid>
    </view>

```

```

<!-- 底部图标 -->
<uni-grid :column="4" :showBorder="false" :highlight="false">
  <!-- 1,2,3,4 -->
  <uni-grid-item v-for="(item, index) in 4">
    <image :src="/static/icons/icon${item}.png" style="width: 100%;"/></image>
  </uni-grid-item>
</uni-grid>
</view>
</template>

<script>
export default {
  data() {
    return {
      data: null,
      baseUrl: 'http://www.codeboy.com:9999/'
    };
  },
  mounted() {
    this.getData();
  },
  methods: {
    getData() {
      const url = 'http://www.codeboy.com:9999/data/product/index.php';
      uni.request({
        url,
        method: 'GET',
        data: {},
        success: res => {
          console.log(res);
          this.data = res.data;
        },
        fail: () => {},
        complete: () => {}
      });
    }
  },
  // 将接口返回的数据 进行拆分
  computed: {
    // 轮播图
    carouselItems() {
      return this.data.carouselItems;
    },
    // 1F 首页推荐
    recommendedItems() {
      return this.data.recommendedItems;
    },
    // 2F 最新上架
    newArrivalItems() {
      return this.data.newArrivalItems;
    },
    // 3F 热销单品
    topSaleItems() {

```

```

        return this.data.topSaleItems;
    },
    // 通过观察发现 1F, 2F, 3F的模板结构一致 但是数据不一样 通过在模板外层再套一层循环解决
    items() {
        return [
            { title: '1F/首页推荐', data: this.recommendedItems },
            { title: '2F/最新上架', data: this.newArrivalItems },
            { title: '3F/热销单品', data: this.topSaleItems }
        ];
    }
}
};
</script>

<style lang="scss">
.header {
    padding: 15rpx;
    background-color: #eeeeee;
    > text {
        margin-left: 15rpx;
        font-weight: 400;
        font-size: 1.1em;
    }
}

.cell {
    padding: 5rpx;
    margin-left: 5rpx;
    border: 1px solid #c0c4cc;
    border-radius: 10rpx;
    > image {
        width: 100%;
        height: 280rpx;
    }
    > view {
        padding-left: 10rpx;
        // 一行 隐藏 ...
        white-space: nowrap;
        overflow: hidden;
        text-overflow: ellipsis;
        // https://www.runoob.com/cssref/css-selectors.html  css各选择器的用法
        // & 当前元素
        &:first-of-type {
            font-weight: bold;
        }
        &:nth-of-type(2) {
            color: gray;
            font-size: 0.9em;
        }
        &:last-of-type {
            color: red;
            font-size: 1.1em;
        }
    }
}
}

```



```
}  
</style>
```

## 商品列表页

功能和注意点:

- u-list 布局
- 注意需要打开pages.json里的打开下拉刷新
- 使用到的事件 `onReachBottom`[触底加载], `onPullDownRefresh`[下拉加载], `onPageScroll`

根据之前学习案例中的代码修改即可

`/pages/category/category.vue`

```
<template>  
  <view>  
    <!-- 商品列表 -->  
    <!-- 发送请求=>拿数据=>存本地=>做展示 -->  
    <!-- 4.做展示 -->  
    <!-- 远程加载数据 如果没有数据请求回来 为null 判断数据为null 就不渲染页面 -->  
    <uni-list v-if="data">  
      <uni-list-item v-for="(item, index) in data.data" :key="index">  
        <!-- 自定义插槽 -->  
        <template slot="body">  
          <view class="cell">  
            <!-- 图片 -->  
            <image  
              :src="`http://www.codeboy.com:9999/${item.pic}`"  
            ></image>  
            <view>  
              <!-- 产品名称 -->  
              <view>{{ item.title }}</view>  
              <!-- 价格 -->  
              <view>{{ item.price }}</view>  
            </view>  
          </view>  
        </template>  
      </uni-list-item>  
    </uni-list>  
    <!-- 加载组件 提示文字 -->  
    <!-- u-load -->  
    <!-- status 取值就三个 more loading noMore -->  
    <uni-load-more :status="status"></uni-load-more>  
    <!-- 回到顶部按钮 -->  
    <view class="btn-gotop" @click="goTop" v-show="showBtn">  
      <uni-icons color="white" size="30" type="arrowthinup"></uni-icons>  
    </view>  
  </view>  
</template>  
  
<script>  
export default {  
  data() {
```

```

    return {
      // 是否显示回到顶部按钮
      showBtn: false,
      data: null,
      // 加载组件的状态
      status: 'more'
    };
  },
  // 下拉事件
  onPullDownRefresh() {
    console.log('下拉刷新触发了');
    // 请求第一页
    this.getData();
  },
  // 触底事件
  onReachBottom() {
    // console.log('到底了,加载数据');
    // 触底后 加载组件的状态需要改变 loading
    this.status = 'loading';
    // 解构获取当前页码和最大页码 es6
    const { pno, pageCount } = this.data;
    if (pno == pageCount) {
      this.status = 'noMore';
      return;
    }
    // 请求数据
    this.getData(pno + 1);
  },
  // 监听滚动事件
  onPageScroll(e) {
    // 距离顶部的距离
    console.log(e.scrollTop);
    // 距离顶部1500时,就显示回到顶部按钮
    this.showBtn = e.scrollTop > 1500;
  },
  mounted() {
    this.getData();
  },
  methods: {
    // 回到顶部方法
    goTop() {
      // 滚动到页面的位置
      uni.pageScrollTo({
        // 0 为回到顶部
        scrollTop: 0
      });
    },
    getData(pno = 1) {
      // 判断默认请求第一页
      // 如果是触底事件加载 需要请求下一页
      // 1.发送请求
      uni.request({
        url: 'http://www.codeboy.com:9999/data/product/list.php',

```

```

        method: 'GET',
        // 页码参数是pno
        data: { pno: pno },
        success: res => {
            // 2.拿数据
            console.log(res);
            // 3.存本地
            // 如果页数大于1需要进行拼接新旧数据
            if (pno > 1) {
                res.data.data = [...this.data.data, ...res.data.data];
            }
            this.data = res.data;
            // 停止下拉刷新 放在请求加载完毕 并且数据返回后
            uni.stopPullDownRefresh();
        },
        fail: () => {},
        complete: () => {}
    });
}
};
</script>

<style scoped lang="scss">
.cell {
    padding: 15rpx;
    // 弹性布局
    display: flex;
    > image {
        width: 180rpx;
        height: 180rpx;
        // 方法一:最小大小
        // min-width: 180rpx;
        // 方法二: 不要弹性压缩
        // flex 代表以下三个属性 none 0 0 auto
        // flex-grow 一个数字, 规定项目将相对于其他灵活的项目进行扩展的量。
        // flex-shrink 一个数字, 规定项目将相对于其他灵活的项目进行收缩的量。
        // flex-basis 项目的长度。合法值: "auto"、"inherit" 或一个后跟 "%", "px", "em" 或任何其他长度
        // 单位的数字。
        // 不要收缩
        // flex:none
        // 方法三
        flex-shrink: 0;
    }
    > view {
        // 弹性布局
        display: flex;
        // 排列方向
        flex-direction: column;
        // 对齐方式 上下两边
        justify-content: space-between;
        // 商品名称
        // 第一个view

```

```

    > view:first-of-type {
      // 两行 隐藏 ...
      text-overflow: -o-ellipsis-lastline;
      overflow: hidden;
      text-overflow: ellipsis;
      display: -webkit-box;
      -webkit-line-clamp: 2;
      line-clamp: 2;
      -webkit-box-orient: vertical;
    }
    // 价格
    // 最后一个view
    > view:last-of-type {
      color: red;
      font-size: 1.1em;
    }
  }
}
.btn-gotop {
  background-color: black;
  // 浮动
  position: fixed;
  // 定位位置
  bottom: 100rpx;
  right: 50rpx;
  // 透明度
  opacity: 0.7;
  width: 70rpx;
  height: 70rpx;
  text-align: center;
  padding: 40rpx;
  border-radius: 50%;
  overflow: hidden;
}
</style>

```

## 商品详情页

注意点:

- 商品列表点击之后传递商品id到商品详情页
- 商品详情页接收商品id并发送请求获取到商品详情信息
- 根据效果展示页面结构

接口地址:

<http://www.codeboy.com:9999/data/product/details.php?lid=商品id>

```

<template>
  <view v-if="data">
    <!-- 头部的标题 -->
    <view
      style="padding: 10rpx;border-bottom: 1px solid gray;font-size: 30rpx;"
    >

```

```

    >
      {{ details.lname }}
    </view>
    <!-- 轮播图 -->
    <swiper
      :indicator-dots="true"
      :autoplay="true"
      :interval="3000"
      :duration="1000"
      style="width: 100%;height: 700rpx;"
    >
      <swiper-item v-for="(item, index) in picList">
        <image
          :src="baseUrl + item.md"
          style="width: 100%; height: 100%;"
        ></image>
      </swiper-item>
    </swiper>
    <!-- 商品介绍 -->
    <view style="padding-left: 30rpx;">
      <view style="font-weight: bold;">{{ details.title }}</view>
      <view style="color:gray;font-size:0.8em;margin: 15rpx 0;">
        {{ details.subtitle }}
      </view>
      <view style="color:red;font-size:1.2em;font-weight: bold;">
        ¥{{ details.price }}
      </view>
    </view>
    <!-- 同类商品 -->
    <uni-collapse>
      <uni-collapse-item title="更多商品推荐" open>
        <uni-list>
          <uni-list-item
            :title="item.spec"
            v-for="(item, index) in laptopList"
            showArrow
          ></uni-list-item>
        </uni-list>
      </uni-collapse-item>
    </uni-collapse>
    <!-- html页面 介绍图片等 -->
    <view v-html="html_details"></view>
    <view style="height: 200rpx;"></view>
    <!-- 商品导航 -->
    <view style="position: fixed;bottom: 0;width: 100%;">
      <uni-goods-nav/>
    </view>
  </view>
</template>

<script>
export default {
  // 接收路由参数

```

```

onLoad(options) {
    // console.log(options);
    this.lid = options.lid;
},
data() {
    return {
        lid: null,
        data: null,
        baseUrl: 'http://www.codeboy.com:9999/'
    };
},
mounted() {
    this.getData();
},
methods: {
    getData() {
        const url =
            'http://www.codeboy.com:9999/data/product/details.php?lid=' +
            this.lid;

        uni.request({
            url,
            method: 'GET',
            data: {},
            success: res => {
                console.log(res);
                this.data = res.data;
            },
            fail: () => {},
            complete: () => {}
        });
    }
},
// 处理数据
computed: {
    details() {
        return this.data.details;
    },
    family() {
        return this.data.family;
    },
    picList() {
        return this.details.picList;
    },
    laptopList() {
        return this.family.laptopList;
    },
    html_details() {
        let details = this.details.details;
        // 添加域名
        // 
        details = details.replace(
            /src="img/g,

```

```

        'src="' + this.baseUrl + 'img'
    );
    // 限制显示大小
    details = details.replace(/<img/g, '<img width="100%"');
    return details;
}
};
</script>

<style></style>

```

## 直播列表

接口地址:

所有分类: <http://capi.douyucdn.cn/api/v1/getColumnList>

分类下具体内容: <http://capi.douyucdn.cn/api/v1/getColumnDetail?shortName=分类名>

某分类下列表: <https://m.douyu.com/api/room/list?page=1&type=LOL>

</pages/live/live>

```

<template>
  <view>
    <uni-collapse v-if="tags">
      <uni-collapse-item :title="tags[current].cate_name" open>
        <uni-tag
          style="margin: 10rpx;display: inline-block;"
          :text="item.cate_name"
          type="warning"
          :inverted="current == index ? false : true"
          v-for="(item, index) in tags"
          :key="index"
          @click="doCheck(index)"
        ></uni-tag>
      </uni-collapse-item>
    </uni-collapse>
    <!-- 分类显示 -->
    <uni-grid
      :column="3"
      :showBorder="false"
      :highlight="false"
      :square="false"
    >
      <uni-grid-item v-for="(item, index) in cate">
        <view @click="gotoRooms(index)">
          <image
            :src="item.pic_name"
            style="width: 100%;height:300rpx"
          ></image>
          <view>{{ item.tag_name }}</view>
        </view>
      </uni-grid-item>
    </uni-grid>
  </view>
</template>

```

```

        </view>
      </uni-grid-item>
    </uni-grid>
  </view>
</template>

<script>
export default {
  data() {
    return {
      tags: null,
      current: 0,
      cate: null
    };
  },
  mounted() {
    this.getTags();
  },
  methods: {
    // 前往直播房间列表页面
    gotoRooms(index) {
      // console.log('去房间列表');
      console.log(this.cate);
      const { short_name, tag_name } = this.cate[index];
      console.log(short_name, tag_name);
      // 路由跳转带传参
      uni.navigateTo({
        // https://m.douyu.com/api/room/list?page=1&type=LOL
        url: `/pages/rooms/rooms?type=${short_name}&cate_name=${tag_name}`
      });
    },
    // 切换标签
    doCheck(index) {
      this.current = index;
      console.log(this.tags[index]);
      const { short_name, cate_name } = this.tags[index];
      // console.log(cate_name);
      // 修改折叠栏上面的文字
      this.curruent_catename = cate_name;
      // console.log(this.tags[index]);
      // 分类切换之后需要获取分类数据
      // #ifdef H5
      const url = '/capi/api/v1/getColumnDetail?shortName=' + short_name;
      // #endif
      // #ifndef H5
      const url =
        'http://capi.douyucdn.cn/api/v1/getColumnDetail?shortName=' +
        short_name;
      // console.log(url);
      // #endif
      uni.request({
        url,
        method: 'GET',

```



```

        data: {},
        success: res => {
            console.log(res);
            this.cate = res.data.data;
        },
        fail: () => {},
        complete: () => {}
    });
},
// 获取分类标签数据
getTags() {
    // #ifndef H5
    const url = 'http://capi.douyucdn.cn/api/v1/getColumnList';
    // #endif
    // #ifdef H5
    const url = '/capi/api/v1/getColumnList';
    // #endif
    uni.request({
        url,
        method: 'GET',
        data: {},
        success: res => {
            console.log(res);
            this.tags = res.data.data;
        },
        fail: () => {},
        complete: () => {}
    });
}
}
};
</script>

<style></style>

```

/pages/room/room

```

<template>
    <!-- v-if 判断数据没有加载回来 就不渲染 -->
    <view v-if="data">
        <!-- 发请求 拿数据 存本地 做展示 -->
        <!-- 4.做展示 -->
        <uni-grid
            :column="2"
            :showBorder="false"
            :highlight="false"
            :square="false"
        >
            <uni-grid-item v-for="(item, index) in data.list">
                <view class="item">
                    <!-- 图片和角标 -->
                    <view>
                        <!-- 图片 -->
                        <image :src="item.roomSrc"></image>
                    </view>
                </view>
            </uni-grid-item>
        </uni-grid>
    </view>

```

```

        <!-- 左下角 -->
        <view>{{ item.nickname }}</view>
        <!-- 右上角 -->
        <view>{{ item.hn }}</view>
    </view>
    <view>{{ item.roomName }}</view>
    <!-- 房间名称 -->
    </view>
</uni-grid-item>
</uni-grid>
<!-- 加载组件 -->
<uni-load-more :status="status"></uni-load-more>
</view>
</template>

<script>
export default {
  data() {
    return {
      // 初始化数据
      data: null,
      // 加载状态
      status: 'more',
      // 当前type类型的直播
      type:null
    };
  },
  //onLoad初始化获取路由传递参数
  onLoad(options) {
    console.log(options);
    const {short_name, cate_name} = options
    // 传递type给请求接口方法    short_name
    this.type = short_name
    // 修改头部的导航的标题内容    cate_name
    uni.setNavigationBarTitle({
      title: cate_name
    });
  },
  // 下拉刷新
  onPullDownRefresh() {
    this.getData()
    uni.stopPullDownRefresh()
  },
  // 触底事件
  onReachBottom() {
    // 触底时 修改加载组件的状态为loading
    this.status = 'loading';
    console.log('到底了,加载数据吧');
    // 获取当前页数和最大页数
    // es6 解构语法 前面的名称 必须是后面数据 this.data中存在key
    const { nowPage, pageCount } = this.data;
    // 判断如果当前页数就是最大页数 后续没有数据了 就退出
    if (nowPage === pageCount) {

```

```

        this.status = 'noMore';
        return;
    }
    // 请求翻页 +1 翻页操作
    this.getData(nowPage + 1);
},
mounted() {
    this.getData();
},
methods: {
    // 再获取一次数据
    getMore() {
        const url = '/douyu/api/room/list?page=2';
        uni.request({
            url: url,
            method: 'GET',
            data: {type:this.type},
            success: res => {
                // 2.拿数据
                console.log(res);
                // 3.存本地
                // 旧数据+新数据
                // console.log(this.data);
                // console.log(res.data.data.list);
                res.data.data.list = [
                    ...this.data.list,
                    ...res.data.data.list
                ];
                this.data = res.data.data;
            },
            fail: () => {},
            complete: () => {}
        });
    },
    getData(page=1) {
        // const url = 'https://m.douyu.com/api/room/list?page=1&type=yx'
        // 配置proxy代理地址需要修改 添加一个标识头
        // 代理服务捕捉到代理标识后, 会进行对应的代理操作
        const url = '/douyu/api/room/list';
        // 1.发请求
        uni.request({
            url: url,
            method: 'GET',
            data: {page:page,type:this.type},
            success: res => {
                // 2.拿数据
                console.log(res);
                // 3.存本地
                // this.data = res.data.data;
                // 多获取一次数据调用
                // 如果非第一页数据 就不需要再多加载了
                if(page == 1){
                    this.getMore();
                }
            }
        });
    }
}

```

```

        }
        if(page>1){
            res.data.data.list = [...this.data.list,...res.data.data.list]
        }
        this.data = res.data.data
    },
    fail: () => {},
    complete: () => {}
  });
}
}
};
</script>

<style scoped lang="scss">
.item {
  padding: 10rpx;
  image {
    width: 100%;
    height: 250rpx;
    border-radius: 6rpx;
    display: block;
  }
  > view:first-of-type {
    // 相对
    position: relative;
    > view:first-of-type {
      // 左下角
      // 绝对
      position: absolute;
      left: 0;
      bottom: 0;
      color: white;
      background-color: rgba(0, 0, 0, 0.5);
      border-radius: 6rpx;
      font-size: 0.9em;
    }
    > view:last-of-type {
      // 右上角
      // 绝对
      position: absolute;
      right: 0;
      top: 0;
      color: white;
      background-color: rgba(0, 0, 0, 0.5);
      border-radius: 6rpx;
      font-size: 0.9em;
    }
  }
  > view:last-of-type {
    font-size: 1.2em;
    overflow: hidden;
    white-space: nowrap;
  }
}

```

```
text-overflow: ellipsis;

}

}

</style>
```

## 英雄列表

接口地址

英雄列表

[https://game.gtimg.cn/images/lol/act/img/js/heroList/hero\\_list.js?v=50](https://game.gtimg.cn/images/lol/act/img/js/heroList/hero_list.js?v=50)

英雄详情

<https://game.gtimg.cn/images/lol/act/img/js/hero/> 英雄id.js

头像地址

<https://game.gtimg.cn/images/lol/act/img/champion/> 英雄alias.png

根据之前 [案例侧边栏导航](#) 参考完成即可。

## uni-app打包

### 微信小程序

运行到微信开发者工具,后续操作按照之前学习微信小程序开发的操作即可。

### 手机App打包

苹果打包发布需要 购买开发者证书 个人\$99/每年 大约RMB700? 企业\$299

安卓的APK,可以直接免费云打包。

#### ①APP相关配置



## ②打包

xueziApp - App打包

应用名称：学子商城WEB2107

应用版本号：100

修改manifest配置

☒ Android (apk包)

☐ iOS (ipa包)

Android设置

iOS设置

Android包名

uni.UNI9B1029E

Android证书使用指南

☐ 使用自有证书 [如何生成证书](#)

☒ 使用云端证书 [详情](#)

☐ 使用公共测试证书 [详情](#)

☐ 使用DCloud老版证书

证书别名

证书私钥密码

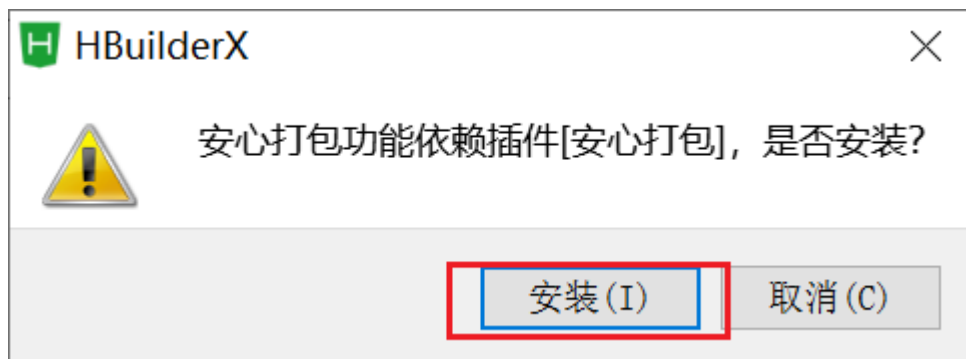
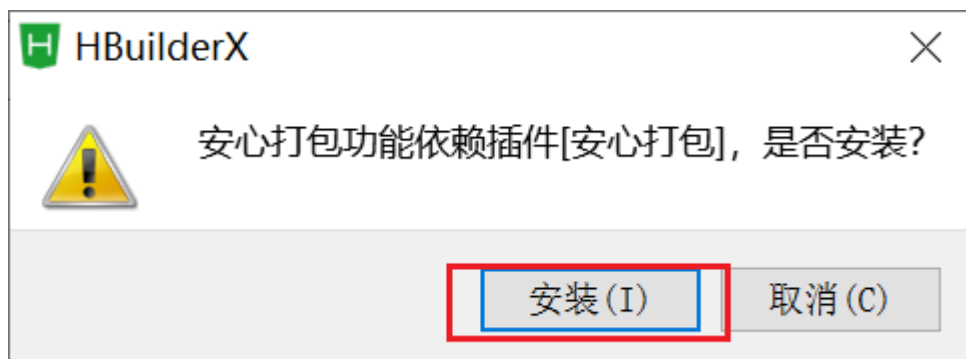
证书文件

浏览...

☐ 传统打包 (上传代码及证书, DCloud承诺不保留)

☒ 安心打包 (不上传代码及证书) [详情](#)

打包(P)



[HBuilder] 10:51:52.559 周五傍晚等高峰期打包排队较长, 请耐心等待。  
[HBuilder] 10:51:52.559 如果是为了三方SDK调试, 请使用自定义调试基座 (菜单运行-手机或模拟器-制作自定义调试基座), 不要反复打包。

[HBuilder] 10:51:52.612 项目 xueziApp [\_\_UNI\_\_9B1029E]的打包状态: **等待打包完成**  
[HBuilder] 10:51:52.612 时间: 2021-12-08 10:51:50 类型: Android云端证书 队列中  
[HBuilder] 10:51:52.613 目前云打包排队人数较多, 当前打包任务位于队列第 31 位, 预计 2 分钟内进入打包状态。

[HBuilder] 10:55:14.710 项目 xueziApp [\_\_UNI\_\_9B1029E]的打包状态: 安装包位置  
[HBuilder] 10:55:14.710 时间: 2021-12-08 10:51:50 类型: Android云端证书 打包成功 安装包位置: D:/code/uni-app/day06/xueziApp/unpackage/  
release/apk/\_\_UNI\_\_9B1029E\_\_20211208105150.apk [打开所在目录] [一键上传到uniCloud (免费cdn, 长期稳定)]

获取到apk文件之后,通过 微信,QQ,数据线传输 到手机安装使用.

## H5

打包成为网页







## H5发行

网站标题 学子商城WEB2107-H5

网站域名 输入网站域名

域名没有可以不填写

☐ 将编译后的资源部署到前端网页托管

发行(P)

取消(C)

H5打包之后生成的文件

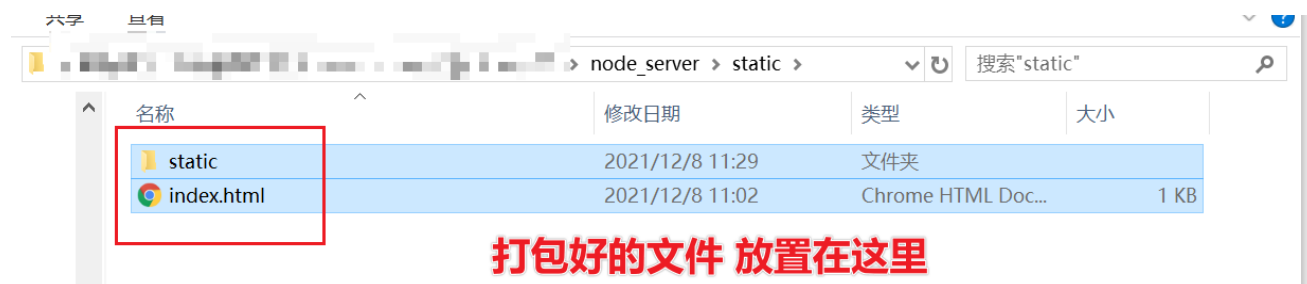


在公司业务上线,使用的是Nginx,也可以使用express起一个服务.

### ①建立项目包并安装express

```
# 项目目录初始化
npm init -y
# 安装express框架
npm i -S express
```

### ②打包好的文件放在在static目录



### ③编写服务文件并启动

app.js

```
const express = require('express')
const app = express()
app.use(express.static('static'))
app.listen(5000, () => {
  console.log('server is running at http://localhost:5000');
})
```

