

标题 2023版uniapp从入门到上天视频教程(java后端无废话版), 火爆更新中...

作者简介

我叫曹锋, 网名: java1234_小锋, 前世界500强央企软件工程师, 10年Java老司机, 技术专家, 高级讲师, 每天坚持锻炼身体, 坚持早睡早起, 崇尚自由, 平时喜欢带带Java学员 (已经成功指导2000+学员高薪就业), 喜欢搞搞产品, 附带搞搞技术自媒体, 喜欢研究主流技术, 热爱技术和教育。小锋网络科技 光杠司令员, 司令部: www.java1234.vip 还有一个人网站: www.java1234.com

java全栈学习路线图: <http://www.java1234.vip/article/1>

java1234 VIP特惠活动: <http://www.java1234.com/vip.html>



爆丑照, 美颜后, 依然很丑。

高清视频+源码+配套文档获取



关注 **java1234** 公众号，回复 **uni**

本套课程的基本版高清视频+源码+配套文档

进本课程专属微信交流群



加上面锋哥微信，**java8822** 备用 **java9266**，备注暗号：**XDD**

进入专属技术交流群。

即可进入uniapp课程的专属技术交流群。

uni-app简介

uni-app官网：<https://uniapp.dcloud.net.cn/>

uni-app 是一个使用 [Vue.js](#) 开发所有前端应用的框架，开发者编写一套代码，可发布到iOS、Android、Web（响应式）、以及各种小程序（微信/支付宝/百度/头条/飞书/QQ/快手/钉钉/淘宝）、快应用等多个平台。

uni-app 在手，做啥都不愁。即使不跨端，**uni-app** 也是更好的小程序开发框架（[详见](#)）、更好的App跨平台框架、更方便的H5开发框架。不管领导安排什么样的项目，你都可以快速交付，不需要转换开发思维、不需要更改开发习惯。

学习uni-app之前，大家先学习下vue和微信小程序基础；

vue基础课程：<http://www.java1234.vip/course/72>

微信小程序基础课程: <http://www.java1234.vip/course/153>

如果有时间, 还可以学习下对应的实战课程

vue实战课程-前后端分离SpringBoot+Vue3权限系统: <https://www.bilibili.com/video/BV1GU4y1r7U>

微信小程序电商实战课程: <https://www.bilibili.com/video/BV1kP4y1F7tU/>

uni-app环境搭建

uni-app开发工具 hbuilderX安装

<https://www.dcloud.io/hbuilderx.html>

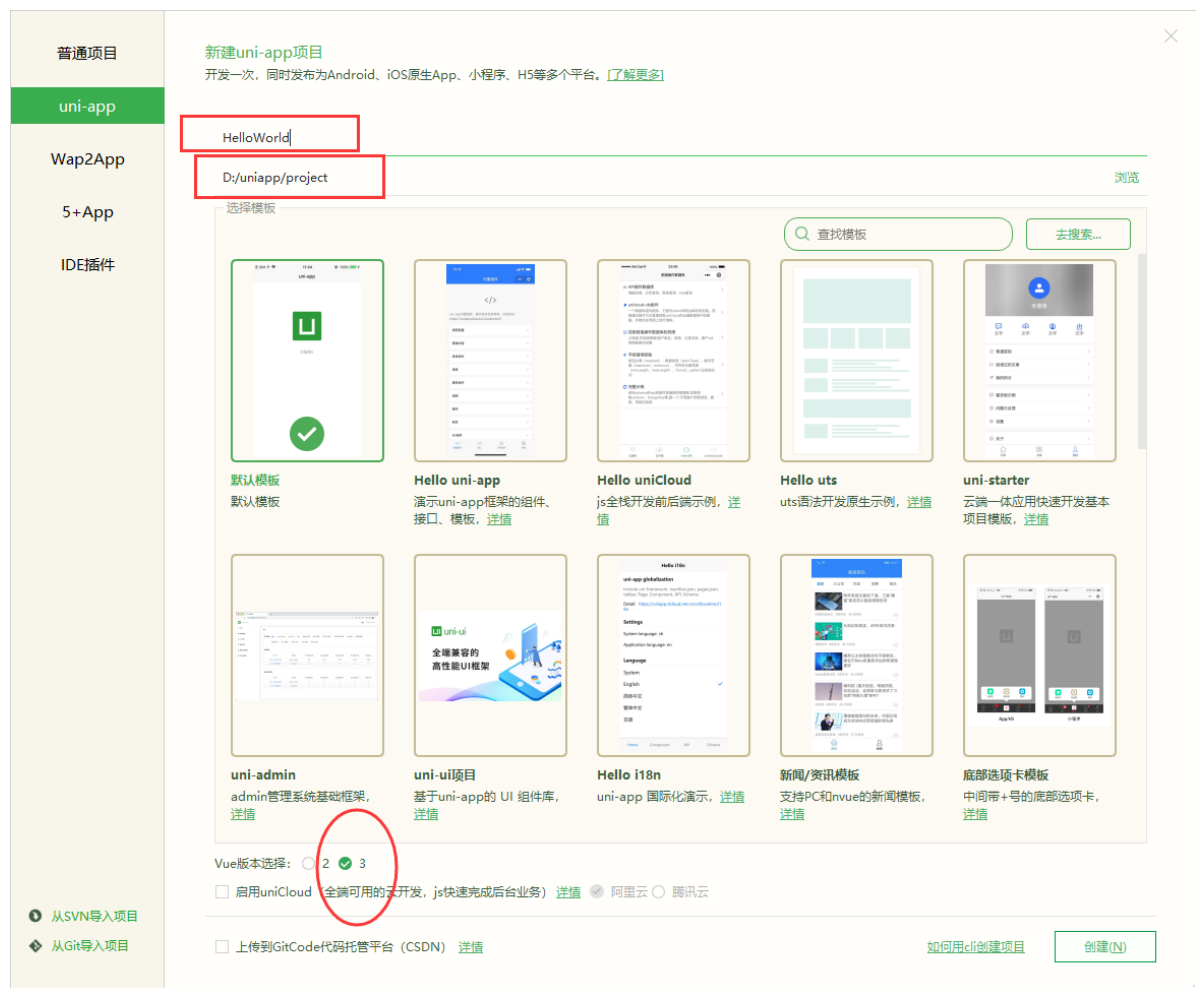
微信小程序开发工具

<https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html>

uni-app之HelloWorld实现

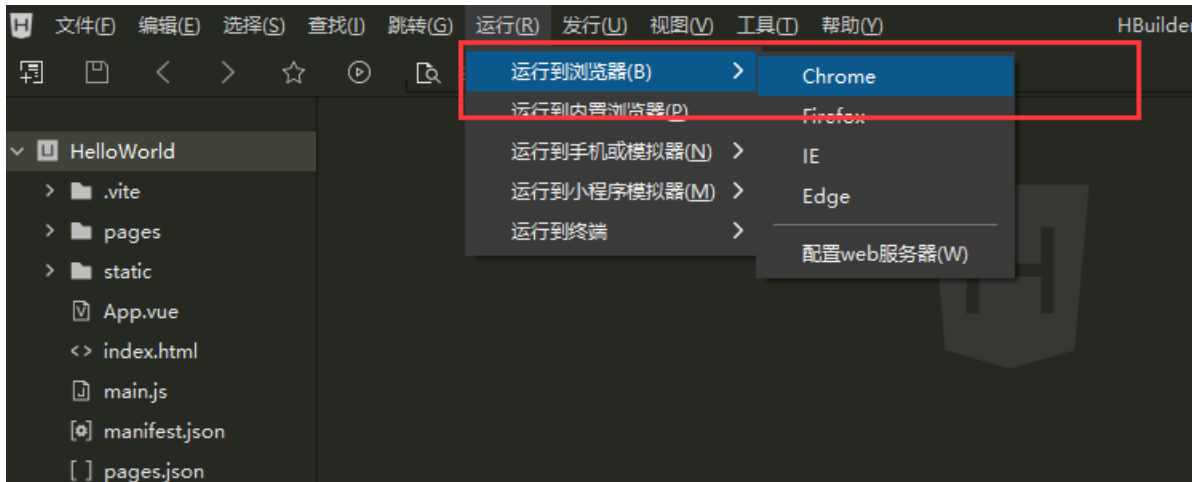
打开HBuilderX, 目录 文件 -> 新建

输入项目名称, 以及项目目录, 勾选Vue3版本, 我们刚开始学习, 选择默认模板, 然后点击“创建”按钮

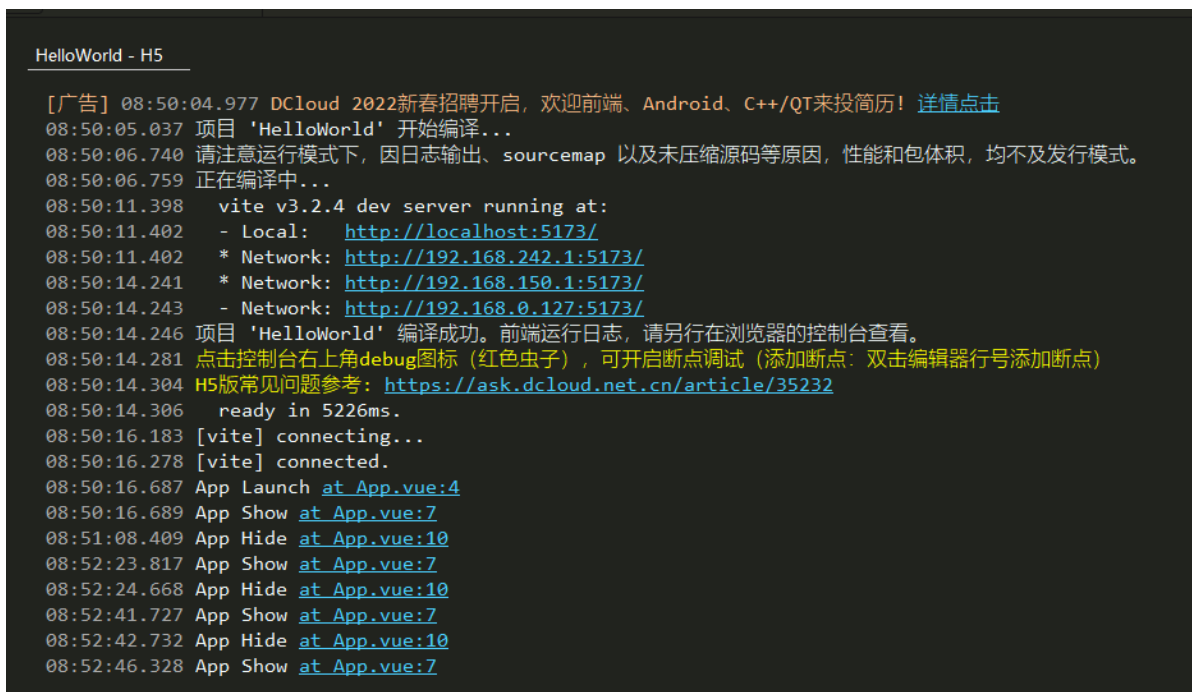


启动项目，运行到浏览器谷歌：

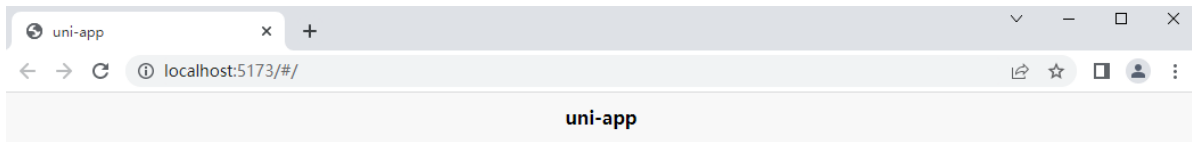
菜单目录 运行 -> 运行到浏览器 -> Chrome



HBuilderX进行编译发布

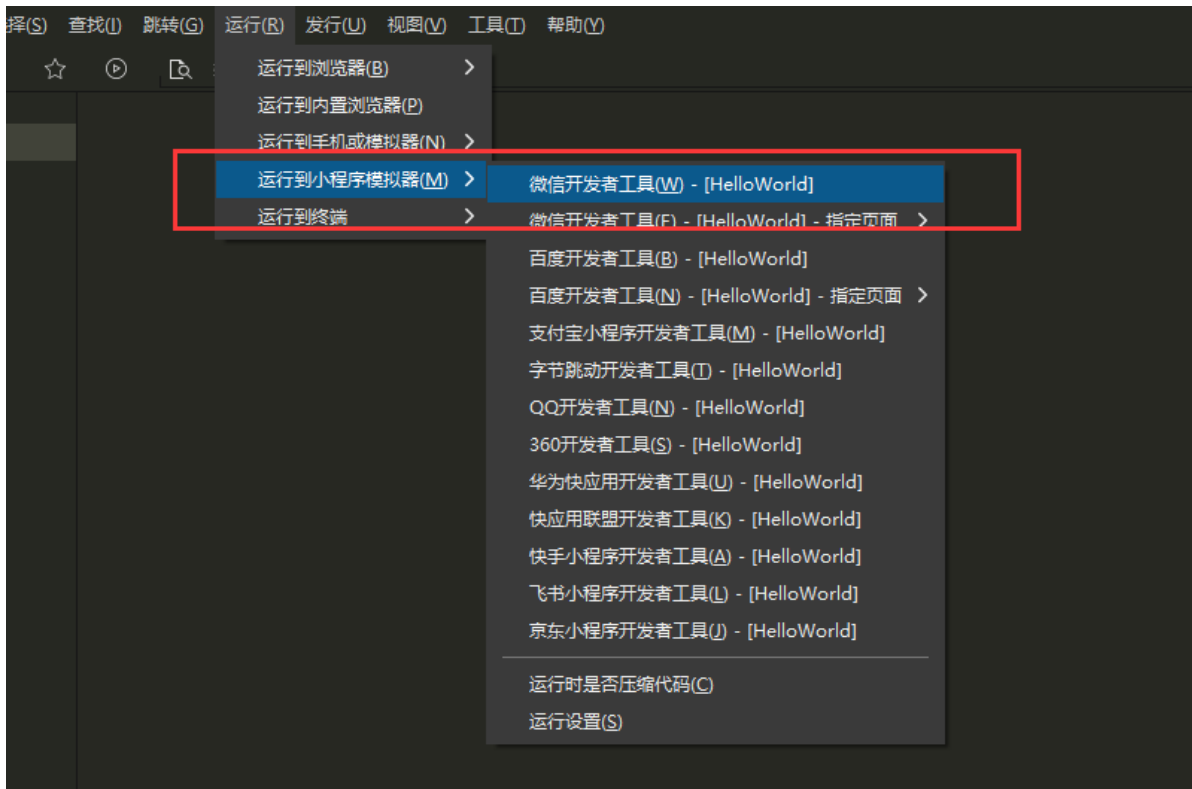


自动打开谷歌浏览器，显示页面效果：



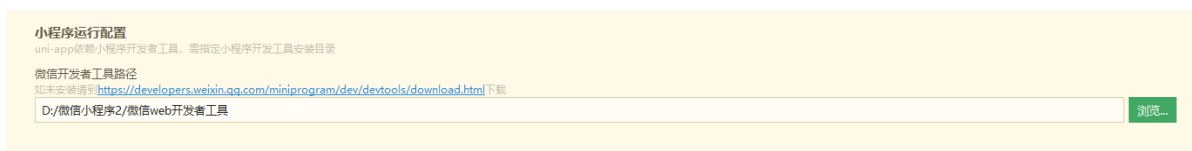
启动项目，运行到微信小程序工具：

菜单目录 运行 -> 运行到小程序模拟器 -> 微信开发者工具

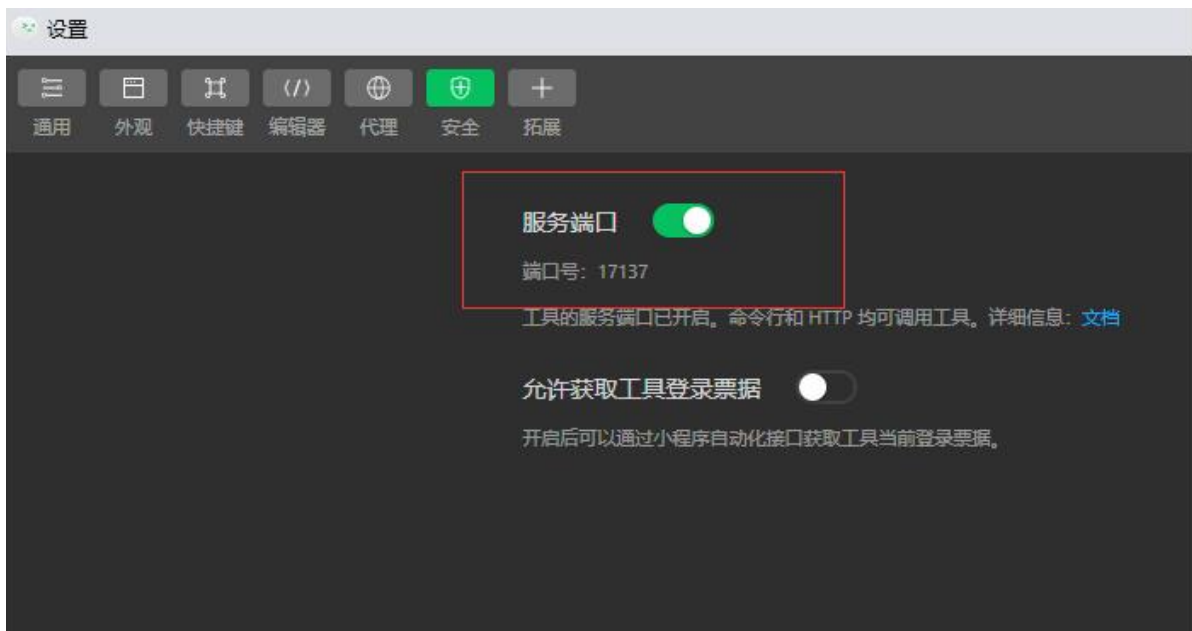


注意：

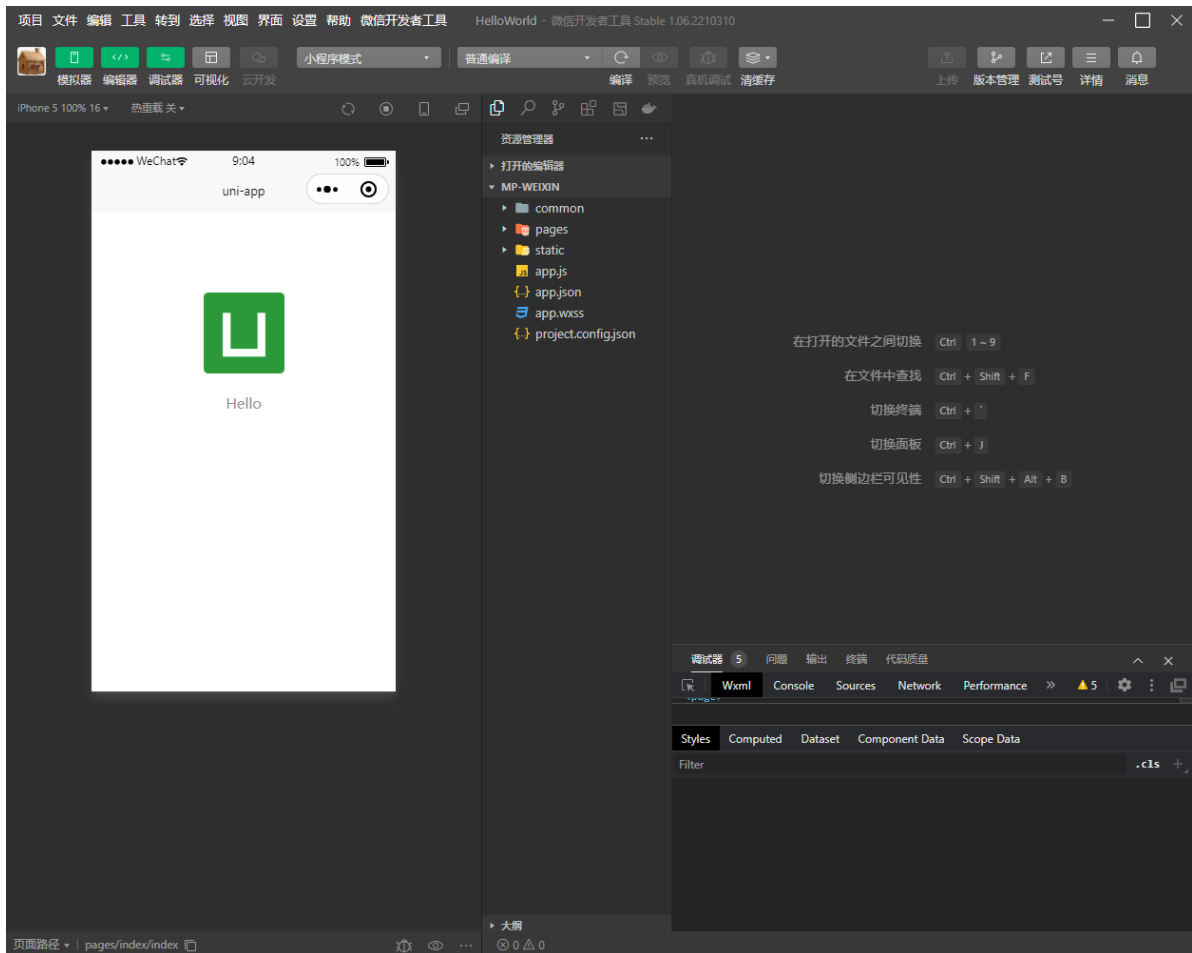
1，如果是第一次使用，需要先配置小程序ide的相关路径，才能运行成功。如下图，需在输入框输入微信开发者工具的安裝路径。



2，微信小程序开发工具，设置->安全设置 开启服务端口



成功运行。



工程目录结构介绍

一个uni-app工程，默认包含如下目录及文件：

└─pages	业务页面文件存放的目录
└─index	
└─index.vue	index页面
└─list	
└─list.vue	list页面
└─static	存放应用引用的本地静态资源（如图片、视频等）的目录，注意：静态资源只能存放于此
└─unpackage	非工程代码，一般存放运行或发行的编译结果
└─main.js	Vue初始化入口文件
└─App.vue	应用配置，用来配置App全局样式以及监听 应用生命周期
└─manifest.json	配置应用名称、appid、logo、版本等打包信息，详见
└─pages.json	配置页面路由、导航条、选项卡等页面类信息，详见
└─uni.scss	这里是uni-app内置的常用样式变量

static目录 使用注意

- 编译到任意平台时，static 目录下除不满足[条件编译](#)的文件，会直接复制到最终的打包目录，不会打包编译。非 static 目录下的文件（vue、js、css 等）只有被引用时，才会被打包编译。
- css、less/scss 等资源不要放在 static 目录下，建议这些公用的资源放在自建的 common 目录下。

Tips

- HbuilderX 1.9.0+ 支持在根目录创建 ext.json、sitemap.json 等小程序需要的文件。

pages.json之globalStyle全局页面样式配置

[globalStyle](#) 设置默认页面的窗口表现 】

颜色RGB取值网站推荐：<https://www.sojson.com/rgb.html>

用于设置应用的状态栏、导航条、标题、窗口背景色等。

属性	类型	默认值	描述	平台差异说明
navigationBarBackgroundColor	HexColor	#F7F7F7	导航栏背景颜色（同状态栏背景色）	APP与H5为#F7F7F7，小程序平台请参考相应小程序文档
navigationBarTextStyle	String	white	导航栏标题颜色及状态栏前景颜色，仅支持 black/white	支付宝小程序不支持，请使用 my.setNavigationBar
navigationBarTitleText	String		导航栏标题文字内容	
backgroundColor	HexColor	#ffffff	下拉显示出来的窗口的背景色	微信小程序
backgroundTextStyle	String	dark	下拉 loading 的样式，仅支持 dark / light	微信小程序
enablePullDownRefresh	Boolean	false	是否开启下拉刷新，详见 页面生命周期 。	
onReachBottomDistance	Number	50	页面上拉触底事件触发时距页面底部距离，单位只支持px，详见 页面生命周期	

page新建以及page外观配置

uni-app 通过 pages 节点配置应用由哪些页面组成，pages 节点接收一个数组，数组每个项都是一个对象，其属性值如下：

属性	类型	默认值	描述
path	String		配置页面路径
style	Object		配置页面窗口表现，配置项参考下方 pageStyle

新建list目录和list vue页面




```
<template>
  <view>列表页面</view>
</template>

<script>
</script>

<style>
</style>
```

pages.json配置页面

```
{
  "path": "pages/list/list",
  "style": {
    "navigationBarBackgroundColor": "#FF00FF",
    "navigationBarTitleText": "列表页"
  }
}
```

tabbar组件

如果应用是一个多 tab 应用，可以通过 tabBar 配置项指定一级导航栏，以及 tab 切换时显示的对应页。

在 pages.json 中提供 tabBar 配置，不仅仅是为了方便快速开发导航，更重要的是在App和小程序端提升性能。在这两个平台，底层原生引擎在启动时无需等待js引擎初始化，即可直接读取 pages.json 中配置的 tabBar 信息，渲染原生tab。

Tips

- 当设置 position 为 top 时，将不会显示 icon
- tabBar 中的 list 是一个数组，只能配置最少2个、最多5个 tab，tab 按数组的顺序排序。
- tabBar 切换第一次加载时可能渲染不及时，可以在每个tabbar页面的onLoad生命周期里先弹出一个等待雪花（hello uni-app使用了此方式）
- tabBar 的页面展现过一次后就保留在内存中，再次切换 tabBar 页面，只会触发每个页面的 onShow，不会再触发onLoad。
- 顶部的 tabBar 目前仅微信小程序上支持。需要用到顶部选项卡的话，建议不使用 tabBar 的顶部设置，而是自己做顶部选项卡，可参考 hello uni-app->模板->顶部选项卡。

属性说明：

属性	类型	必填	默认值	描述	平台差异说明
color	HexColor	是		tab 上的文字默认颜色	
selectedColor	HexColor	是		tab 上的文字选中时的颜色	
backgroundColor	HexColor	是		tab 的背景色	
borderStyle	String	否	black	tabbar 上边框的颜色，可选值 black/white，也支持其他颜色值	App 2.3.4+、H5 3.0.0+
blurEffect	String	否	none	iOS 高斯模糊效果，可选值 dark/extralight/light/none（参考: 使用说明 ）	App 2.4.0+ 支持、H5 3.0.0+（只有最新版浏览器才支持）
list	Array	是		tab 的列表，详见 list 属性说明，最少2个、最多5个 tab	
position	String	否	bottom	可选值 bottom、top	top 值仅微信小程序支持
fontSize	String	否	10px	文字默认大小	App 2.3.4+、H5 3.0.0+
iconWidth	String	否	24px	图标默认宽度（高度等比例缩放）	App 2.3.4+、H5 3.0.0+
spacing	String	否	3px	图标和文字的间距	App 2.3.4+、H5 3.0.0+
height	String	否	50px	tabBar 默认高度	App 2.3.4+、H5 3.0.0+
midButton	Object	否		中间按钮 仅在 list 项为偶数时有效	App 2.3.4+、H5 3.0.0+
iconfontSrc	String	否		list设置 iconfont 属性时，需要指定字体文件路径	App 3.4.4+、H5 3.5.3+

其中 list 接收一个数组，数组中的每个项都是一个对象，其属性值如下：

属性	类型	必填	说明	平台差异
pagePath	String	是	页面路径，必须在 pages 中先定义	
text	String	是	tab 上按钮文字，在 App 和 H5 平台为非必填。例如中间可放一个没有文字的+号图标	
iconPath	String	否	图片路径，icon 大小限制为 40kb，建议尺寸为 81px * 81px，当 position 为 top 时，此参数无效，不支持网络图片，不支持字体图标	
selectedIconPath	String	否	选中时的图片路径，icon 大小限制为 40kb，建议尺寸为 81px * 81px，当 position 为 top 时，此参数无效	
visible	Boolean	否	该项是否显示，默认显示	App (3.2.10+)、H5 (3.2.10+)
iconfont	Object	否	字体图标，优先级高于 iconPath	App (3.4.4+) 、H5 (3.5.3+)

```

"tabBar": {
  "color": "#FF8C00",
  "selectedColor": "#B22222",
  "backgroundColor": "#FFFAFA",
  "borderStyle": "black",
  "position": "bottom",
  "list": [
    {
      "text": "主页",
      "pagePath": "pages/index/index",
      "iconPath": "static/tabbar/icon/_home.png",
      "selectedIconPath": "static/tabbar/icon/home.png"
    },
    {
      "text": "分类",
      "pagePath": "pages/classification/classification",
      "iconPath": "static/tabbar/icon/_classification.png",
      "selectedIconPath": "static/tabbar/icon/classification.png"
    },
    {
      "text": "购物车",
      "pagePath": "pages/cart/cart",
      "iconPath": "static/tabbar/icon/_cart.png",
      "selectedIconPath": "static/tabbar/icon/cart.png"
    },
    {
      "text": "我的",

```

```
        "pagePath": "pages/my/my",
        "iconPath": "static/tabbar/icon/_my.png",
        "selectedIconPath": "static/tabbar/icon/my.png"
      }
    ]
  }
}
```

condition启动模式配置

启动模式配置，仅开发期间生效，用于模拟直达页面的场景，如：小程序转发后，用户点击所打开的页面。

属性说明：

属性	类型	是否必填	描述
current	Number	是	当前激活的模式，list节点的索引值
list	Array	是	启动模式列表

list说明：

属性	类型	是否必填	描述
name	String	是	启动模式名称
path	String	是	启动页面路径
query	String	否	启动参数，可在页面的 onLoad 函数里获得

参考demo：

```
"condition": {
  "current": 0,
  "list": [
    {
      "name": "详情页",
      "path": "pages/detail/detail",
      "query": "id=66"
    }
  ]
}
```

uni.scss内置全局样式变量

uni.scss 文件的用途是为了方便整体控制应用的风格。比如按钮颜色、边框风格，uni.scss 文件里预置了一批scss变量预置。

uni-app 官方扩展插件（uni ui）及 [插件市场](#) 上很多三方插件均使用了这些样式变量，如果你是插件开发者，建议你使用 scss 预处理，并在插件代码中直接使用这些变量（无需 import 这个文件），方便用户通过搭积木的方式开发整体风格一致的App。

`uni.scss` 是一个特殊文件，在代码中无需 `import` 这个文件即可在 `scss` 代码中使用这里的样式变量。`uni-app` 的编译器在 `webpack` 配置中特殊处理了这个 `uni.scss`，使得每个 `scss` 文件都被注入这个 `uni.scss`，达到全局可用的效果。如果开发者想要 `less`、`stylus` 的全局使用，需要在 `vue.config.js` 中自行配置 `webpack` 策略。

注意：

1. 如要使用这些常用变量，需要在 `HBuilderX` 里面安装 `scss` 插件；
2. 使用时需要在 `style` 节点上加上 `lang="scss"`。

```
<style lang="scss">
</style>
```

3. `pages.json` 不支持 `scss`，原生导航栏和 `tabbar` 的动态修改只能使用 `js api`

App.vue 生命周期 & 全局样式 & 全局存储 globalData

`App.vue` 是 `uni-app` 的主组件，所有页面都是在 `App.vue` 下进行切换的，是页面入口文件。但 `App.vue` 本身不是页面，这里不能编写视图元素，也就是没有 `<template>`。

这个文件的作用包括：调用应用生命周期函数、配置全局样式、配置全局的存储 `globalData`

应用生命周期仅可在 `App.vue` 中监听，在页面监听无效。

应用生命周期

`uni-app` 支持如下应用生命周期函数：

函数名	说明
<code>onLaunch</code>	当 <code>uni-app</code> 初始化完成时触发（全局只触发一次）
<code>onShow</code>	当 <code>uni-app</code> 启动，或从后台进入前台显示
<code>onHide</code>	当 <code>uni-app</code> 从前台进入后台
<code>onError</code>	当 <code>uni-app</code> 报错时触发
<code>onUniNViewMessage</code>	对 <code>nvue</code> 页面发送的数据进行监听，可参考 nvue 向 vue 通讯
<code>onUnhandledRejection</code>	对未处理的 <code>Promise</code> 拒绝事件监听函数（2.8.1+）
<code>onPageNotFound</code>	页面不存在监听函数
<code>onThemeChange</code>	监听系统主题变化

示例代码

```
<script>
// 只能在App.vue里监听应用的生命周期
export default {
  onLaunch: function() {
    console.log('App Launch')
  },
  onShow: function() {
    console.log('App Show')
  }
}
```

```
    },
    onHide: function() {
      console.log('App Hide')
    }
  }
}
</script>
```

注意

- 应用生命周期仅可在 `App.vue` 中监听，在其它页面监听无效。
- 应用启动参数，可以在API `uni.getLaunchOptionsSync` 获取，[详见](#)
- `onLaunch`里进行页面跳转，如遇白屏报错，请参考<https://ask.dcloud.net.cn/article/35942>
- `App.vue` 不能写模板
- `onPageNotFound` 页面实际上已经打开了（比如通过分享卡片、小程序码）且发现页面不存在，才会触发，api 跳转不存在的页面不会触发（如 `uni.navigateTo`）

globalData

小程序有globalData，这是一种简单的全局变量机制。这套机制在uni-app里也可以使用，并且全端通用。

以下是 `App.vue` 中定义globalData的相关配置：

```
<script>
  export default {
    globalData: {
      text: 'text'
    }
  }
</script>
```

js中操作globalData的方式如下：`getApp().globalData.text = 'test'`

在应用onLaunch时，getApp对象还未获取，暂时可以使用this.globalData获取globalData。

如果需要把globalData的数据绑定到页面上，可在页面的onShow页面生命周期里进行变量重赋值。

nvue的weex编译模式中使用globalData的话，由于weex生命周期不支持onShow，但熟悉5+的话，可利用监听webview的addEventListener show事件实现onShow效果，或者直接使用weex生命周期中的beforeCreate。但建议开发者使用uni-app编译模式，而不是weex编译模式。

globalData是简单的全局变量，如果使用状态管理，请使用 `vuex`（main.js中定义）

全局样式

在 `App.vue` 中，可以定义一些全局通用样式，例如需要加一个通用的背景色，首屏页面渲染的动画等都可以写在App.vue中。

注意如果工程下同时有vue和nvue文件，全局样式的所有css会应用于所有文件，而nvue支持的css有限，编译器会在控制台报警，提示某些css无法在nvue中支持。此时需要把nvue不支持的css写在单独的条件编译里。如：

```
<style>
  /* #ifndef APP-PLUS-NVUE */
  @import './common/uni.css';
  /* #endif */
</style>
```

js语法

uni-app 的js API由标准ECMAScript的js API 和 uni 扩展 API 这两部分组成。

标准ECMAScript的js仅是最基础的js。浏览器基于它扩展了window、document、navigator等对象。小程序也基于标准js扩展了各种wx.xx、my.xx、swan.xx的API。node也扩展了fs等模块。

uni-app基于ECMAScript扩展了uni对象，并且API命名与小程序保持兼容。

标准js和浏览器js的区别

uni-app 的js代码，h5端运行于浏览器中。非h5端（包含小程序和App），Android平台运行在v8引擎中，iOS平台运行在iOS自带的jscore引擎中，都没有运行在浏览器或webview里。

非H5端，虽然不支持window、document、navigator等浏览器的js API，但也支持标准ECMAScript。

请注意不要把浏览器里的js扩展对象等价于标准js。

所以uni-app的非H5端，一样支持标准js，支持if、for等语法，支持字符串、数字、时间、布尔值、数组、自定义对象等变量类型及各种处理方法。仅仅是不支持window、document、navigator等浏览器专用对象。

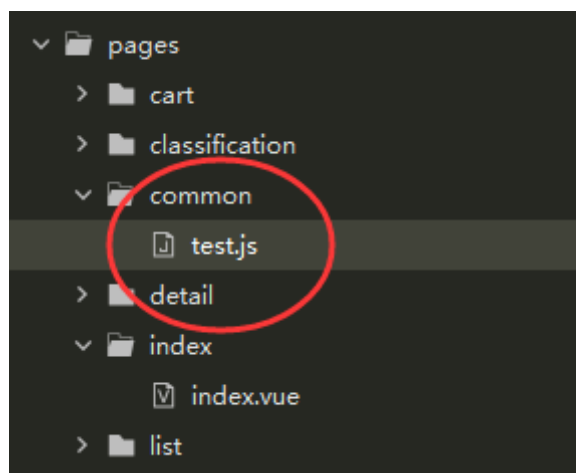
ES6 支持

uni-app 在支持绝大部分 ES6 API 的同时，也支持了 ES7 的 await/async。

ES6 API 的支持，详见如下表格部分（x 表示不支持，无特殊说明则表示支持）：

es6课程：<http://www.java1234.vip/course/71>

pages下新建common目录，再新建test.js



```

const a=2;

const obj={
  name:'jack',
  age:11,
  say(){
    console.log("姓名: "+this.name+" 年龄: "+this.age)
  },
  say2:function(){
    console.log("say2 姓名: "+this.name+" 年龄: "+this.age)
  },
  say3:(name,age)=>{
    console.log("say3 姓名: "+name+" 年龄: "+age)
  }
}

function func(m,n){
  return m+n;
}

export{
  a,
  obj,
  func
}

```

index.vue修改:

```

<template>
  <view class="content">
    <image class="logo" src="/static/logo.png"></image>
    <view class="text-area">
      <view class="title">{{title}}</view>
      <view class="title common_style">{{text}}</view>
      <view class="title">{{text2}}</view>
    </view>
  </view>
</template>

<script>
import {a,obj,func} from "@pages/common/test.js"
export default {
  data() {
    return {
      title: 'Hello',
      text: '',
      text2:0
    }
  },
  onLoad() {
    getApp().globalData.text="修改数据"
    this.text=getApp().globalData.text;
    this.text2=a;
    console.log(obj.name+": "+obj.age)
  }
}

```



```

        obj.say();
        obj.say2();
        obj.say3("marry", "12");
        console.log(func(1,2))
    },
    methods: {

    }
}
</script>

```

css语法

uni-app 的 css 与 web 的 css 基本一致。

尺寸单位

uni-app 支持的通用 css 单位包括 px、rpx

- px 即屏幕像素
- rpx 即响应式 px，一种根据屏幕宽度自适应的动态单位。以 750 宽的屏幕为基准，750rpx 恰好为屏幕宽度。屏幕变宽，rpx 实际显示效果会等比放大，但在 App（vue2 不含 nvue）端和 H5（vue2）端屏幕宽度达到 960px 时，默认将按照 375px 的屏幕宽度进行计算

```

<template>
  个人中心页面
  <view class="c1"></view>
  <view class="c2"></view>
</template>

<script>
</script>

<style lang="scss">
  .c1{
    width: 750rpx;
    height: 750rpx;
    background-color: $uni-color-error;
  }
  .c2{
    margin-top: 10px;
    width: 375rpx;
    height: 375rpx;
    background-color: $uni-bg-color-mask;
  }
</style>

```



样式导入

使用 `@import` 语句可以导入外联样式表，`@import` 后跟需要导入的外联样式表的相对路径，用 `;` 表示语句结束。

common目录下新建test.css

```
.c3{
  margin-top: 10px;
  background-color: blue;
  width: 100rpx;
  height: 100rpx;
}
```

my.vue页面修改

加入 @import url("../common/test.css");

```
1 <template>
2   个人中心页面
3   <view class="c1"></view>
4   <view class="c2"></view>
5   <view class="c3"></view>
6 </template>
7
8 <script>
9 </script>
10
11 <style lang="scss">
12   @import url("../common/test.css");
13   .c1{
14     width: 750rpx;
15     height: 750rpx;
16     background-color: $uni-color-error;
17   }
18   .c2{
19     margin-top: 10px;
20     width: 375rpx;
21     height: 375rpx;
22     background-color: $uni-bg-color-mask;
23   }
24 </style>
```

内联样式

框架组件上支持使用 style、class 属性来控制组件的样式。

- style: 静态的样式统一写到 class 中。style 接收动态的样式，在运行时会进行解析，请尽量避免将静态的样式写进 style 中，以免影响渲染速度。

```
<view :style="{color:color}" />
```

- class：用于指定样式规则，其属性值是样式规则中类选择器名(样式类名)的集合，样式类名不需要带上.，样式类名之间用空格分隔。

```
<view class="normal_view" />
```

选择器

目前支持的选择器有：

选择器	样例	样例描述
.class	.intro	选择所有拥有 class="intro" 的组件
#id	#firstname	选择拥有 id="firstname" 的组件
element	view	选择所有 view 组件
element, element	view, checkbox	选择所有文档的 view 组件和所有的 checkbox 组件
::after	view::after	在 view 组件后边插入内容， 仅 vue 页面生效
::before	view::before	在 view 组件前边插入内容， 仅 vue 页面生效

注意：

- 在 uni-app 中不能使用 * 选择器。
- 微信小程序自定义组件中仅支持 class 选择器
- page 相当于 body 节点，例如：

```
<!-- 设置页面背景颜色，使用 scoped 会导致失效 -->
page {
  background-color: #ccc;
}
```

全局样式与局部样式

定义在 App.vue 中的样式为全局样式，作用于每一个页面。在 pages 目录下的 vue 文件中定义的样式为局部样式，只作用在对应的页面，并会覆盖 App.vue 中相同的选择器。

注意：

- App.vue 中通过 @import 语句可以导入外联样式，一样作用于每一个页面。
- nvue 页面暂不支持全局样式

CSS变量

uni-app 提供内置 CSS 变量

CSS 变量	描述	App	小程序	H5
--status-bar-height	系统状态栏高度	系统状态栏高度 、nvue 注意见下	25px	0
--window-top	内容区域距离顶部的距离	0	0	NavigationBar 的高度
--window-bottom	内容区域距离底部的距离	0	0	TabBar 的高度

注意：

- `var(--status-bar-height)` 此变量在微信小程序环境为固定 25px，在 App 里为手机实际状态栏高度。
- 当设置 `"navigationStyle": "custom"` 取消原生导航栏后，由于窗体为沉浸式，占据了状态栏位置。此时可以使用一个高度为 `var(--status-bar-height)` 的 view 放在页面顶部，避免页面内容出现在状态栏。
- 由于在 H5 端，不存在原生导航栏和 tabbar，也是前端 div 模拟。如果设置了一个固定位置的居底 view，在小程序和 App 端是在 tabbar 上方，但在 H5 端会与 tabbar 重叠。此时可使用 `--window-bottom`，不管在哪个端，都是固定在 tabbar 上方。
- 目前 nvue 在 App 端，还不支持 `--status-bar-height` 变量，替代方案是在页面 onLoad 时通过 `uni.getSystemInfoSync().statusBarHeight` 获取状态栏高度，然后通过 style 绑定方式给占位 view 设定高度。下方提供了示例代码

代码块

快速书写 css 变量的方法是：在 css 中敲 hei，在候选助手中即可看到 3 个 css 变量。（HBuilderX 1.9.6 以上支持）

示例 1 - 普通页面使用 css 变量：

```

<template>
  <!-- HBuilderX 2.6.3+ 新增 page-meta，详情：
https://uniapp.dcloud.io/component/page-meta -->
  <page-meta>
    <navigation-bar />
  </page-meta>
  <view>
    <view class="status_bar">
      <!-- 这里是状态栏 -->
    </view>
    <view>状态栏下的文字</view>
  </view>
</template>
<style>
  .status_bar {
    height: var(--status-bar-height);
    width: 100%;
  }
</style>

```

```

<template>
  <view>
    <view class="toTop">
      <!-- 这里可以放一个向上箭头，它距离底部tabbar上浮10px-->
    </view>
  </view>
</template>
<style>
  .toTop {
    bottom: calc(var(--window-bottom) + 10px);
  }
</style>

```

示例 2 - nvue 页面获取状态栏高度

```

<template>
  <view class="content">
    <view :style="{ height: iStatusBarHeight + 'px' }"></view>
  </view>
</template>

<script>
  export default {
    data() {
      return {
        iStatusBarHeight: 0,
      };
    },
    onLoad() {
      this.iStatusBarHeight = uni.getSystemInfoSync().statusBarHeight;
    },
  };
</script>

```

固定值

uni-app 中以下组件的高度是固定的，不可修改：

组件	描述	App	H5
NavigationBar	导航栏	44px	44px
TabBar	底部选项卡	HBuilderX 2.3.4 之前为 56px，2.3.4 起和 H5 调为一致，统一为 50px。（但可以自主更改高度）	50px

各小程序平台，包括同小程序平台的 iOS 和 Android 的高度也不一样。

Flex布局

为支持跨平台，框架建议使用 Flex 布局，关于 Flex 布局可以参考外部文档[A Complete Guide to Flexbox](#)、[阮一峰的 flex 教程](#)等。

背景图片

`uni-app` 支持使用在 `css` 里设置背景图片，使用方式与普通 `web` 项目大体相同，但需要注意以下几点：

- 支持 base64 格式图片。
- 支持网络路径图片。
- 小程序不支持在 `css` 中使用本地文件，包括本地的背景图和字体文件。需以 base64 方式方可使用。
- 使用本地路径背景图片需注意：
 1. 为方便开发者，在背景图片小于 40kb 时，`uni-app` 编译到不支持本地背景图的平台时，会自动将其转化为 base64 格式；
 2. 图片大于等于 40kb，会有性能问题，不建议使用太大的背景图，如开发者必须使用，则需自己将其转换为 base64 格式使用，或将其挪到服务器上，从网络地址引用。
 3. 本地背景图片的引用路径推荐使用以 `~@` 开头的绝对路径。

```
.test2 {  
  background-image: url('~@/static/logo.png');  
}
```

注意

- 微信小程序不支持相对路径（真机不支持，开发工具支持）

字体图标

`uni-app` 支持使用字体图标，使用方式与普通 `web` 项目相同，需要注意以下几点：

- 支持 base64 格式字体图标。
- 支持网络路径字体图标。
- 小程序不支持在 `css` 中使用本地文件，包括本地的背景图和字体文件。需以 base64 方式方可使用。
- 网络路径必须加协议头 `https`。
- 从 <http://www.iconfont.cn> 上拷贝的代码，默认是没加协议头的。
- 从 <http://www.iconfont.cn> 上下载的字体文件，都是同名字体（字体名都叫 iconfont，安装字体文件时可以看到），在 `nvue` 内使用时需要注意，此字体名重复可能会显示不正常，可以使用工具修改。
- 使用本地路径图标字体需注意：
 1. 为方便开发者，在字体文件小于 40kb 时，`uni-app` 会自动将其转化为 base64 格式；
 2. 字体文件大于等于 40kb，仍转换为 base64 方式使用的话可能有性能问题，如开发者必须使用，则需自己将其转换为 base64 格式使用，或将其挪到服务器上，从网络地址引用；
 3. 字体文件的引用路径推荐使用以 `~@` 开头的绝对路径。

```
@font-face {
  font-family: test1-icon;
  src: url('~@/static/iconfont.ttf');
}
```

nvue 中不可直接使用 css 的方式引入字体文件，需要使用以下方式在 js 内引入。nvue 内不支持本地路径引入字体，请使用网络链接或者 base64 形式。src 字段的 url 的括号内一定要使用单引号。

```
var domModule = weex.requireModule('dom');
domModule.addRule('fontFace', {
  fontFamily: 'fontFamilyName',
  src: "url('https://...')",
});
```

示例：

```
<template>
  <view>
    <view>
      <text class="test">&#xe600;</text>
      <text class="test">&#xe687;</text>
      <text class="test">&#xe60b;</text>
    </view>
  </view>
</template>
<style>
  @font-face {
    font-family: 'iconfont';
    src: url('https://at.alicdn.com/t/font_865816_17gjspmrrkti.ttf')
    format('truetype');
  }
  .test {
    font-family: iconfont;
    margin-left: 20rpx;
  }
</style>
```

vue语法

vue基础课程：<https://www.bilibili.com/video/BV1s4411G7o4/>

vue3实战课程：<https://www.bilibili.com/video/BV1GU4y1r7UV/>

掌握以上两个课程 vue功底就相当流弊了。

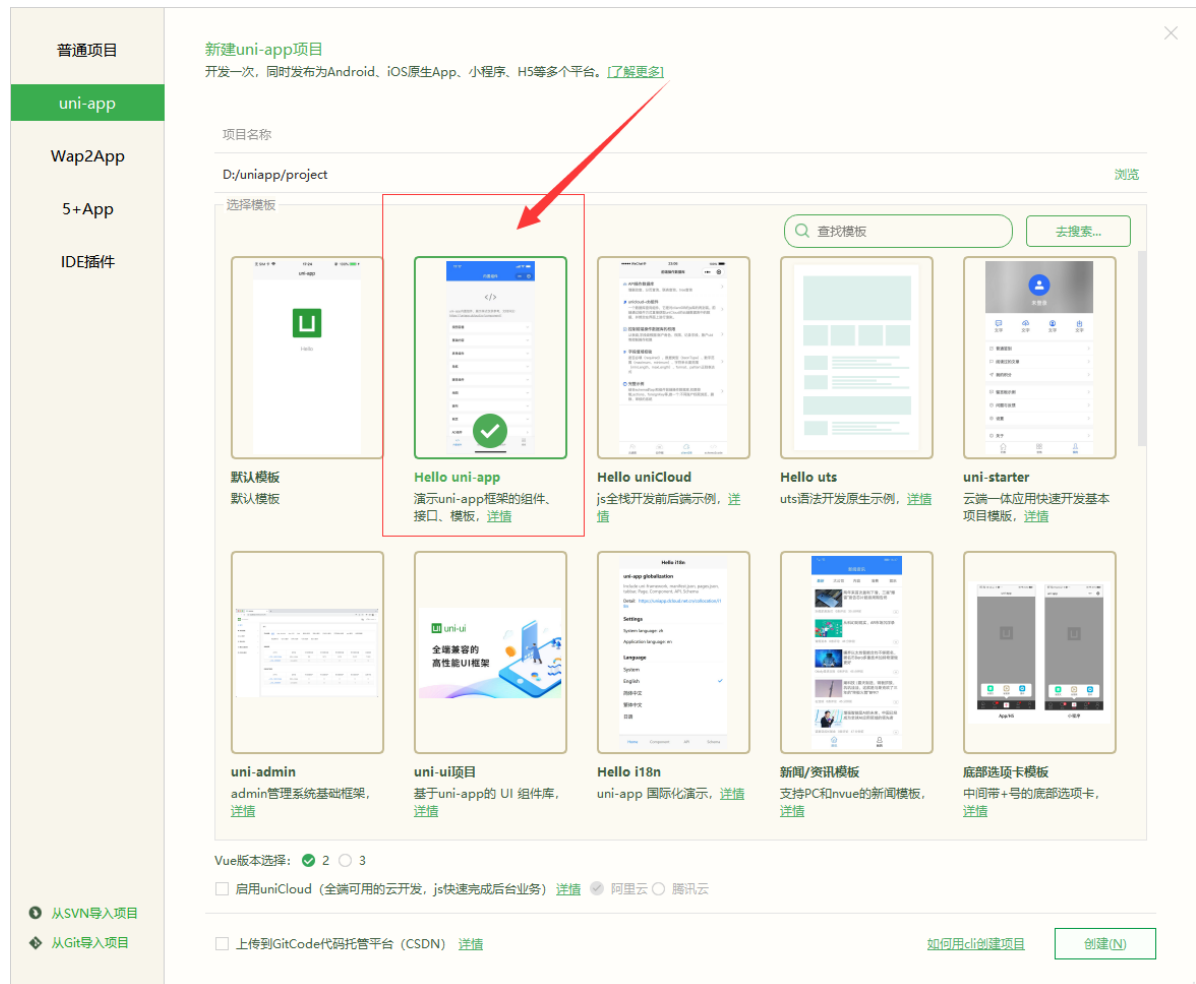
[uni-app vue文档](#)

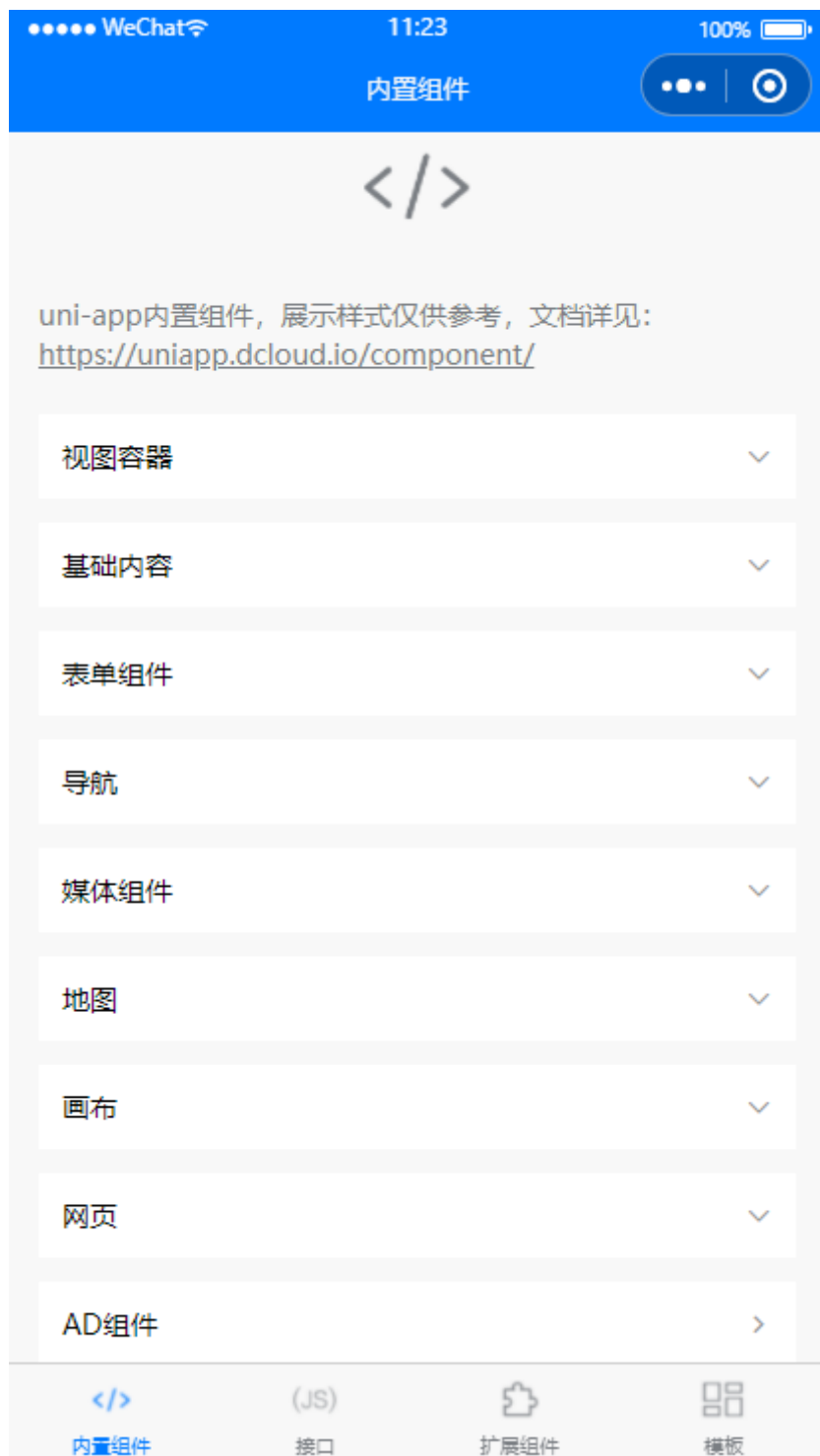
uni-app内置组件

uni-app基础知识掌握后，我们要对uni-app内置的组件有一个基础的全局认识，用到的时候，再根据文档具体研究使用细节。

<https://uniapp.dcloud.net.cn/component/view.html#>

我们学习的时候，新建项目，选择第二个 Hello uni-app模版。





uni-app API接口

uni-app 的 js API 由标准 ECMAScript 的 js API 和 uni 扩展 API 这两部分组成。

标准 ECMAScript 的 js 仅是最基础的 js。浏览器基于它扩展了 window、document、navigator 等对象。小程序也基于标准 js 扩展了各种 wx.xx、my.xx、swan.xx 的 API。node 也扩展了 fs 等模块。

uni-app 基于 ECMAScript 扩展了 uni 对象，并且 API 命名与小程序保持兼容。

文档：<https://uniapp.dcloud.net.cn/api/>

uni-app扩展组件(uni-ui)

uni-ui是DCloud提供的一个跨端ui库，它是基于vue组件的、flex布局的、无dom的跨全端ui框架。

uni-ui不包括基础组件，它是基础组件的补充。

uni-ui产品特点

1、高性能

目前为止，在小程序和混合app领域，uni-ui是性能的标杆。

- 自动增量更新数据

虽然uni-app支持小程序自定义组件，所有小程序的ui库都可以用。但小程序自定义组件的ui库都需要使用setData手动更新数据，在大数据量时、或高频更新数据时，很容易产生性能问题。

而uni-ui属于vue组件，uni-app引擎底层自动diff更新数据。当然其实插件市场里众多vue组件都具备这个特点。

- 优化逻辑层和视图层通讯折损

非H5，不管是小程序还是App，不管是app的webview渲染还是原生渲染，全都是逻辑层和视图层分离的。这里就有一个逻辑层和视图层通讯的折损问题。比如在视图层拖动一个可跟手的组件，由于通讯的损耗，用js监听很难做到实时跟手。

这时就需要使用css动画以及平台底层提供的wxs、bindingx等技术。不过这些技术都比较复杂，所以uni-ui里做了封装，在需要跟手式操作的ui组件，比如swiperaction列表项左滑菜单，就在底层使用了这些技术，实现了高性能的交互体验

- 背景停止

很多ui组件是会一直动的，比如轮播图、跑马灯。即便这个窗体被新窗体挡住，它在背景层仍然在消耗着硬件资源。在Android的webview版本为chrome66以上，背景操作ui会引发很严重的性能问题，造成前台界面明显卡顿。

而uni-ui的组件，会自动判断自己的显示状态，在组件不再可见时，不会再消耗硬件资源。

2、全端

uni-ui的组件都是多端自适应的，底层会抹平很多小程序平台的差异或bug。

比如导航栏navbar组件，会自动处理不同端的状态栏；比如swiperaction组件，在app和微信小程序上会使用交互体验更好的wxs技术，但在不支持wxs的其他小程序端会使用js模拟类似效果。

uni-ui还支持nvue原生渲染。

uni-ui还支持pc等宽屏设备，可以通过PC浏览器访问<https://hellouniapp.dcloud.net.cn/pages/extUI/badge/badge>体验

3、风格扩展

uni-ui的默认风格是中型的，与uni-app基础组件风格一致。但它支持uni.scss，可以方便的扩展和切换应用的风格。

ui是一种需求非常发散的产品，DCloud官方也无意用uni-ui压制第三方ui插件的空间，但官方有义务在性能和跨端方面提供一个开源的标杆给大家。

我们欢迎更多优秀的ui组件出现，也欢迎更多人贡献uni-ui的主题风格，满足更多用户的需求。

4、与uniCloud协作

uni-ui里很多组件与uniCloud打通，可大幅提升开发效率

5、与uni统计自动集成实现免打点

uni统计是优秀的多端统计平台，见tongji.dcloud.net.cn。

除了一张报表看全端，它的另一个重要特点是免打点。比如使用uni-ui的navbar标题栏、收藏、购物车等组件，均可实现自动打点，统计页面标题等各种行为数据。当然你也可以关闭uni统计，这不是强制的。

6、uni-ui符合全套DCloud组件规范

包括easycom、uni_module、datacom，全部遵循。

uni-app模版（扩展插件）

uni-app 积极拥抱社区，创建了开放、兼容的插件系统。

- [uni-app插件市场](#)，有数千款插件，支持前端组件、js sdk、页面模板、项目模板、原生插件等多种类型。在生态建设上远远领先于竞品。
- 兼容 微信小程序 JS SDK

丰富的小程序生态内容可直接引入 uni-app，并且在App侧通用。以前的跨平台开发框架普遍缺少三方 SDK，由于大量 SDK 厂商均原厂维护小程序 SDK，使得 uni-app 成为跨平台开发框架里生态最丰富的平台[参考](#)

- 兼容 微信小程序自定义组件

小程序自定义组件是一种 ui 组件，uni-app 里可以在 App、H5、微信小程序、QQ 小程序同时兼容微信小程序自定义组件，[参考](#)

- 兼容 NPM 包管理系统
- 兼容 weex 插件生态

uni-app 内置了 weex，weex 的原生插件或 ui 库均可使用。注意 weex 的生态不如 uni-app 丰富，一般情况建议使用 uni-app 的插件市场。

- 兼容 普通 web 库

uni-app 的 H5 端支持所有浏览器 API。但众所周知，由于小程序的 js 不运行在浏览器里，所以小程序里不支持 HTML 和 DOM 的 API。

uni-app 的 App 端虽然和小程序是相同的架构，逻辑层也运行在独立 jscore 而不是浏览器里，但 App 端和小程序还是有区别的：

- 一方面可通过 web-view 组件加载 HTML，引入 web 相关库；
- 另一方面可通过 [renderjs](#) 实现在渲染层执行 js，此时完整 echart、threejs 等 web 库均可使用。（但为了全端使用，仍然建议减少对 dom 库的依赖，在 uni-app 的插件市场可寻找全端兼容的库来替代）
- App 端支持各种调用原生能力的方式
 1. 支持 原生[混合开发](#)
 2. 支持 比小程序能力更多的[plus JSAPI](#)
 3. 支持 [Native.js](#) 直接调用原生 api
 4. 支持 [uts 插件](#)
 5. 支持 [原生语言插件](#)
- App 端支持双渲染引擎 uni-app 逻辑层在独立 jscore，而渲染层可选 webview 渲染和 weex 引擎渲染。

1. 使用webview渲染则整个架构与小程序相同，此时页面后缀为vue文件。
2. 使用weex引擎（经过改造）渲染，则整个架构与快应用相同，此时页面后缀为nvue文件。使用webview渲染时，可以指定由系统webview渲染还是由x5引擎渲染。

SpringBoot后端实例搭建

新建数据库db_uni



打开idea 新建项目uni-pro1

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.3.2.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.java1234</groupId>
  <artifactId>uni-pro1</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>uni-pro1</name>
  <description>uni-pro1</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
```

```

        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>

    <!-- 连接池 -->
    <dependency>
        <groupId>com.alibaba</groupId>
        <artifactId>druid</artifactId>
        <version>1.1.10</version>
    </dependency>
    <!-- mybatis-plus -->
    <dependency>
        <groupId>com.baomidou</groupId>
        <artifactId>mybatis-plus-boot-starter</artifactId>
        <version>3.3.2</version>
    </dependency>
    <dependency>
        <groupId>com.baomidou</groupId>
        <artifactId>mybatis-plus-boot-starter</artifactId>
        <version>3.3.2</version>
    </dependency>

    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <optional>true</optional>
    </dependency>

</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <version>2.3.2.RELEASE</version>
        </plugin>
    </plugins>
</build>

</project>

```

application.yml

```

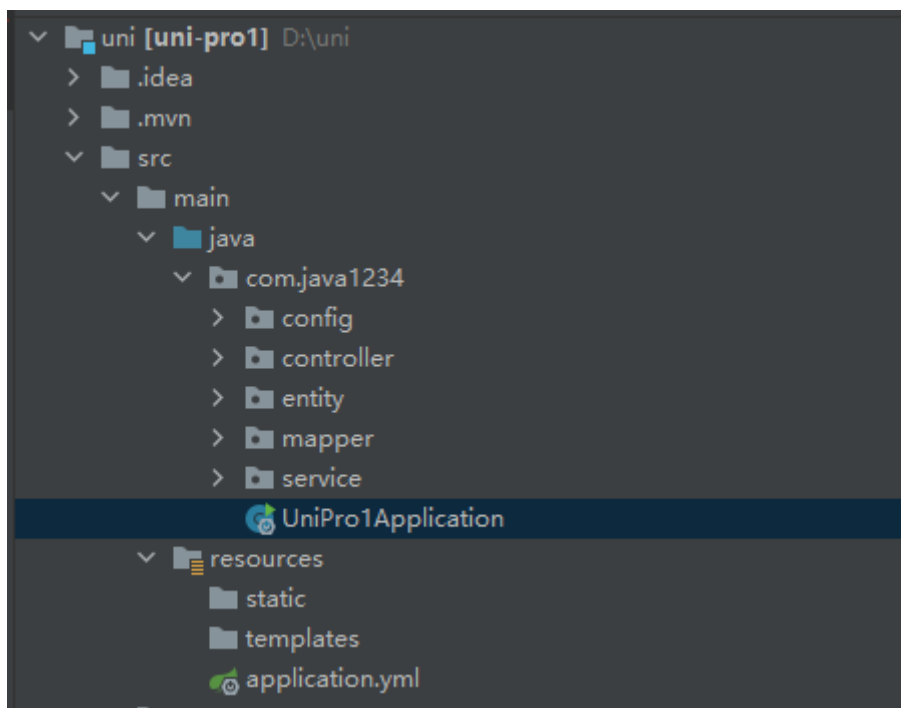
server:
  port: 80
  servlet:
    context-path: /

```

```
spring:
  datasource:
    type: com.alibaba.druid.pool.DruidDataSource
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/db_uni?serverTimezone=Asia/Shanghai
    username: root
    password: 123456

mybatis-plus:
  global-config:
    db-config:
      id-type: auto
  configuration:
    map-underscore-to-camel-case: true
    auto-mapping-behavior: full
    log-impl: org.apache.ibatis.logging.stdout.StdOutImpl
  mapper-locations: classpath:mapper/*.xml
```

目录结构:



启动类 UniPro1Application 加下 扫描注解:

```
@MapperScan("com.java1234.mapper")
```

新建NewsController类

```
/**
 * 新闻Controller控制器
 * @author java1234_小锋 （公众号：java1234）
```

```

* @site www.java1234.vip
* @company 南通小锋网络科技有限公司
*/
@RestController
@RequestMapping("/news")
public class NewsController {

    @Autowired
    private NewsService newsService;

    /**
     * 测试
     * @return
     */
    @GetMapping("/test")
    public R test(){
        return R.ok("测试");
    }

}

```

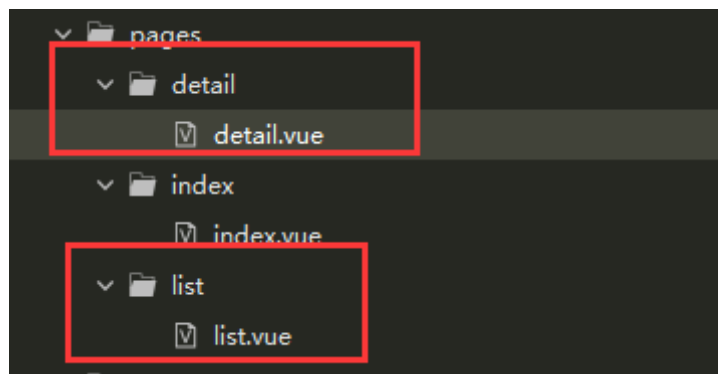
测试: <http://localhost/news/test>



uni-app前端实例搭建

新建uni-app项目 java1234Pro2

新建两个页面



pages.json

```
{
```



```

"pages": [ //pages数组中第一项表示应用启动页，参考：
https://uniapp.dcloud.io/collocation/pages
    {
        "path": "pages/list/list",
        "style": {
            "navigationBarTitleText": "列表页",
            "enablePullDownRefresh": true
        }
    },
    {
        "path": "pages/detail/detail",
        "style": {
            "navigationBarTitleText": "详情页"
        }
    },
    {
        "path": "pages/index/index",
        "style": {
            "navigationBarTitleText": "uni-app"
        }
    }
],
"globalStyle": {
    "navigationBarTextStyle": "black",
    "navigationBarTitleText": "uni-app",
    "navigationBarBackgroundColor": "#F8F8F8",
    "backgroundColor": "#F8F8F8"
},
"uniIdRouter": {}
}

```

数据列表显示

后端：

```

/**
 * 分页查询新闻信息
 * @return
 */
@GetMapping("/list")
public R list() throws InterruptedException {
    List<News> newsList = newsService.list();
    Map<String, Object> resultMap = new HashMap<>();
    resultMap.put("newsList", newsList);
    return R.ok(resultMap);
}

```

前端样式：

```

.uni-list{
    background-color: #FFFFFF;
    position: relative;
}

```

```

width: 100%;
display: flex;
flex-direction: column;
.uni-list-cell {
    position: relative;
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: center;
    .uni-media-list {
        padding: 22rpx 30rpx;
        box-sizing: border-box;
        display: flex;
        width: 100%;
        flex-direction: row;
        .uni-media-list-logo {
            width: 5.625rem;
            height: 4.375rem;
            margin-right: 0.625rem;
        }

        .uni-media-list-body {
            height: auto;
            display: flex;
            flex: 1;
            flex-direction: column;
            justify-content: space-around;
            align-items: flex-start;
            overflow: hidden;
            .uni-media-list-text-top {
                height: 2.3125rem;
                font-size: 0.875rem;
                overflow: hidden;
                width: 100%;
                line-height: 1.125rem;
            }
            .uni-media-list-text-bottom {
                display: flex;
                flex-direction: row;
                justify-content: space-between;
                width: 100%;
                line-height: 0.9375rem;
                font-size: 0.8125rem;
                color: #8f8f94;
            }
        }
    }
}
.uni-list-cell::after {
    position: absolute;
    z-index: 3;
    right: 0;
    bottom: 0;
    left: 0.9375rem;
    height: 1px;
    content: '';
    transform: scaleY(.5);
}

```

```

        background-color: #c8c7cc;
    }

}

```

静态页面:

```

<template>
  <view class="uni-list">
    <view class="uni-list-cell">
      <view class="uni-media-list">
        <image class="uni-media-list-logo"
src="http://localhost/image/20230115051209000000878.png"></image>
        <view class="uni-media-list-body">
          <view class="uni-media-list-text-top">为什么自动驾驶诉讼不断?
        </view>
          <view class="uni-media-list-text-bottom">
            <text>未来汽车日报</text>
            <text>2020-04-11 17:07:48</text>
          </view>
        </view>
      </view>
    </view>
    <view class="uni-list-cell">
      <view class="uni-media-list">
        <image class="uni-media-list-logo"
src="http://localhost/image/20230115051147000000020.png"></image>
        <view class="uni-media-list-body">
          <view class="uni-media-list-text-top">2020数字中国创新大赛-数字
政府赛道21强出炉，四大赛题紧贴政府数字化发展需求</view>
          <view class="uni-media-list-text-bottom">
            <text>36氪深度服务</text>
            <text>2020/04/11-16:59</text>
          </view>
        </view>
      </view>
    </view>
  </view>
</template>

```

动态页面:

```

<block v-for="(value,index) in listData" :key="index">
  <view class="uni-list-cell" @click="goDetail(value)">
    <view class="uni-media-list">
      <image class="uni-media-list-logo"
:src="'http://localhost/image/'+value.coverImage"></image>
      <view class="uni-media-list-body">
        <view class="uni-media-list-text-top">{{value.title}}</view>
        <view class="uni-media-list-text-bottom">
          <text>{{value.author}}</text>
          <text>{{value.releaseDate}}</text>
        </view>
      </view>
    </view>
  </view>
</block>

```

```
</view>
</block>
```

```
export default{
  data(){
    return{
      listData:[
      ]
    }
  },
  onLoad(){
    this.getList();
  },
  methods:{
    getList(){
      uni.request({
        url: 'http://localhost/news/list',
        method: 'POST',
        success: data=>{
          let result=data.data;
          if(result.code===200){
            this.listData=result.newsList;
          }
        },
        fail:(data,code)=>{
          console.log("fail:" + JSON.stringify(data))
        }
      })
    }
  }
}
```

下拉刷新实现

列表页面开启下拉刷新

"enablePullDownRefresh": true

```
index.vue    pages.json
1 {
2   "pages": [ //pages数组中第一项表示应用启动页，参考：https://uniapp.d
3     {
4       "path": "pages/list/list",
5       "style": {
6         "navigationBarTitleText": "列表页",
7         "enablePullDownRefresh": true
8       }
9     },
10    {
11      "path": "pages/detail/detail",
12      "style": {
13        "navigationBarTitleText": "详情页"
14      }
15    },
16    {
17      "path": "pages/index/index",
18      "style": {
19        "navigationBarTitleText": "uni-app"
20      }
21    }
22  ],
23  "globalStyle": {
24    "navigationBarTextStyle": "black",
25    "navigationBarTitleText": "uni-app",
26    "navigationBarBackgroundColor": "#F8F8F8",
27    "backgroundColor": "#F8F8F8"
28  },
29  "uniIdRouter": {}
30 }
```

```
onPullDownRefresh(){
  console.log("下拉刷新");
  this.listData=[];
  this.getList();
},
```

getList方法，success回调方法里，加下停止下拉刷新

```
uni.stopPullDownRefresh();
```

```

    },
    methods:{
        getList(){
            this.status='loading';
            uni.request({
                url:'http://localhost/news/list',
                data:this.queryParams,
                method:'POST',
                success:(data=>{
                    uni.stopPullDownRefresh();
                    let result=data.data;
                    if(result.code==200){
                        this.listData=[...this.listData,...result.newsList]
                        this.totalPage=result.totalPage;
                    }
                },
                fail:(data,code)=>{
                    console.log("fail:" + JSON.stringify(data))
                }
            })
        },
        goDetail:function(e){
            uni.navigateTo({

```

上拉分页实现

后端:

```

/**
 * 分页查询新闻信息
 * @param pageBean
 * @return
 */
@PostMapping("/list")
public R list(@RequestBody PageBean pageBean) throws InterruptedException {
    Thread.sleep(1000);
    Page<News> pageResult = newsService.page(new Page<>(pageBean.getPageNum(),
    pageBean.getPageSize()));
    Map<String, Object> resultMap=new HashMap<>();
    resultMap.put("newsList", pageResult.getRecords());
    resultMap.put("total", pageResult.getTotal());
    resultMap.put("totalPage", pageResult.getPages());
    return R.ok(resultMap);
}

```

前端加下参数:

```

queryParams:{
    pageNum:1,
    pageSize:10
},
totalPage:0,

```

```

},
methods:{
  getList(){
    this.status='loading';
    uni.request({
      url:'http://localhost/news/list',
      data:this.queryParams,
      method:'POST',
      success:data=>{
        uni.stopPullDownRefresh();
        let result=data.data;
        if(result.code==200){
          this.listData=[...this.listData,...result.newsList]
          this.totalPage=result.totalPage;
        }
      },
      fail:(data,code)=>{
        console.log("fail:" + JSON.stringify(data))
      }
    })
  }
}

```

```

},
onPullDownRefresh(){
  console.log("下拉刷新");
  this.listData=[];
  this.queryParams.pageNum=1;
  this.getList();
},
methods:{

```

uni-load-more 组件

<https://ext.dcloud.net.cn/plugin?id=29>

```

<uni-load-more :status="status" :icon-size="16" :content-text="contentText">
</uni-load-more>

```

```

export default{
  data(){
    return{
      queryParams:{
        pageNum:1,
        pageSize:10
      },
      totalPage:0,
      listData:[
      ],
      status:'more',
      contentText: {
        contentdown: '上拉加载更多',
        contentrefresh: '加载中',

```

```

        contentnomore: '没有更多'
      }
    }
  },
  onLoad(){
    this.getList();
  },
  onReachBottom(){
    console.log("下拉触底")
    if(this.queryParams.pageNum>=this.totalPage){
      console.log("没有下一页数据")
      // uni.showToast({
      //   title: '没有下一页数据了'
      // });
      this.status="no-more";
    }else{
      console.log("有下一页数据")
      this.queryParams.pageNum++;
      this.getList();
    }
  },
  onPullDownRefresh(){
    console.log("下拉刷新");
    this.listData=[];
    this.queryParams.pageNum=1;
    this.getList();
  },
  methods:{
    getList(){
      this.status='loading';
      uni.request({
        url:'http://localhost/news/list',
        data:this.queryParams,
        method:'POST',
        success:data=>{
          uni.stopPullDownRefresh();
          let result=data.data;
          if(result.code==200){
            this.listData=[...this.listData,...result.newsList]
            this.totalPage=result.totalPage;
          }
        },
        fail:(data,code)=>{
          console.log("fail:" + JSON.stringify(data))
        }
      })
    },
    goDetail:function(e){
      uni.navigateTo({
        url:'/pages/detail/detail?id='+e.id
      })
    }
  }
}

```


详情页实现

后端:

```
/**
 * 根据id查询新闻详细信息
 * @param id
 * @return
 */
@GetMapping("/detail/{id}")
public R detail(@PathVariable(value = "id") Integer id){
    News news = newsService.getById(id);
    Map<String,Object> resultMap=new HashMap<>();
    resultMap.put("news",news);
    return R.ok(resultMap);
}
```

加下点击事件:

```
1 <template>
2   <view class="uni-list">
3     <block v-for="(value,index) in listData" :key="index">
4       <view class="uni-list-cell" @click="goDetail(value)">
5         <view class="uni-media-list">
6           <image class="uni-media-list-logo" :src="'http://localhost/image/'+value" />
7           <view class="uni-media-list-body">
8             <view class="uni-media-list-text-top">{{value.title}}</view>
9             <view class="uni-media-list-text-bottom">
10               <text>{{value.author}}</text>
11               <text>{{value.releaseDate}}</text>
12             </view>
13           </view>
14         </view>
15       </view>
16     </block>
```

跳转方法:

```
goDetail:function(e){
    uni.navigateTo({
        url: '/pages/detail/detail?id='+e.id
    })
}
```

样式:

```
.banner {
    height: 360rpx;
    overflow: hidden;
    position: relative;
    background-color: #ccc;
    .banner-img {
        width: 100%;
    }

    .banner-title {
```

```

        max-height: 84rpx;
        overflow: hidden;
        position: absolute;
        left: 30rpx;
        bottom: 30rpx;
        width: 90%;
        font-size: 32rpx;
        font-weight: 400;
        line-height: 42rpx;
        color: white;
        z-index: 11;
    }
}

.article-meta {
    padding: 20rpx 40rpx;
    display: flex;
    flex-direction: row;
    justify-content: flex-start;
    color: gray;

    .article-text {
        font-size: 26rpx;
        line-height: 50rpx;
        margin: 0 20rpx;
    }

    .article-author,
    .article-time {
        font-size: 30rpx;
    }
}

.article-content {
    padding: 0 30rpx;
    overflow: hidden;
    font-size: 30rpx;
    margin-bottom: 30rpx;
}

```

```

<view>
  <view class="banner">
    <image class="banner-img"
:src="'http://localhost/image/'+news.coverImage"></image>
    <view class="banner-title">{{news.title}}</view>
  </view>
  <view class="article-meta">
    <text class="article-author">{{news.author}}</text>
    <text class="article-text">发表于</text>
    <text class="article-time">{{news.releaseDate}}</text>
  </view>
  <view class="article-content">
    <rich-text :nodes="news.content"></rich-text>
  </view>
</view>

```

```

export default{
  data(){
    return {
      news:{}
    }
  },
  onLoad(event){
    const id=event.id || 0;
    console.log("id="+id)
    this.getDetail(id);
  },
  methods:{
    getDetail(id){
      if(id>0){
        uni.request({
          url:"http://localhost/news/detail/"+id,
          success:(data)=>{
            let result=data.data;
            this.news=result.news;
            uni.setNavigationBarTitle({
              title:result.news.title
            })
          },
          fail:(data,code)=>{
            console.log("fail:" + JSON.stringify(data))
          }
        })
      }
    }
  }
}

```