# uniapp39道面试题 (https://github.com/minsion/)

### 

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

#### 试题回答参考思路:

uniApp是一款基于Vue.js的跨平台开发框架,可以用于开发iOS、Android和Web应用程序。它使用了简单但强大的API,使得开发人员可以轻松地创建跨平台应用程序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/QQ/钉钉/淘宝)、快应用等多个平台。

uni-app官网是: https://uniapp.dcloud.io/

选择uni-app的原因:

uni-app在开发者数量、案例、跨端抹平度、扩展灵活性、性能体验、周边生态、学习成本、开发成本等8大关键指标上拥有更强的优势。

### 🖿 面试题 2. 简述uniApp相对于其他跨平台框架的优点是什么?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题 ▶

#### 试题回答参考思路:

相对于其他跨平台框架, uniApp有以下优点:

- 1 轻量级: uniApp比其他跨平台框架更加轻量级,因此它可以更快地加载和运行,从而提高应用程序的性能。
- 2 完美的支持: uniApp具有完美的平台支持,可以运行于iOS、Android和Web平台,且兼容主流的浏览器和操作系统。
- 3 强大的UI组件: uniApp拥有强大的UI组件库,包括丰富的原生组件和自定义组件,可以方便地创建精美的用户界面。
- 4 易于学习和使用: uniApp基于Vue.js框架, 所以对于已经熟悉Vue.js开发的开发者来说, 学习和使用uniApp是非常容易的。

# 🛅 面试题 3. 简述uniApp适用于哪些场景?

推荐指数: ★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

- (1) 多平台应用:如果您需要在iOS、Android和Web平台上构建应用,uniApp就是一个非常好的选择。
- (2)独立APP:如果您需要构建一个独立的移动应用程序,uniApp也是一个非常不错的选 择。
- (3) 基础应用:如果您需要构建一个基于模板的应用程序,uniApp也可以很适用。

### 🛅 面试题 4. 解释uniApp有哪些限制?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

#### 试题回答参考思路:

uniApp有一些限制,包括:

(1) 性能:虽然uniApp的性能比大多数跨平台框架都好,但与原生开发相比仍有一定差距。

- (2) 自定义组件: uniApp不支持所有的自定义组件,有一些组件无法在uniApp中使用。
- (3) 功能:虽然uniApp提供了很多原生的系统组件和API,但它还是无法完全满足所有的功能需求。

### 🛅 面试题 5. 简述uniApp并不支持某些平台上的高级特性?

推荐指数: ★★ 试题难度: 初级 试题类型: 原理题▶

# 试题回答参考思路:

Uni-app并不支持某些平台上的高级特性,以下是一些例子:

3D渲染: Uni-app并不支持在部分平台上的3D渲染功能,在实现3D场景渲染时需要进行组件或扩展封装。

桌面应用: Uni-app没有对桌面应用提供官方支持。

VR和AR技术支持: 在VR和AR技术领域,可能因为硬件和软件的限制,在部分平台上不被支持。某些平台上无法使用原生的 AR/VR SDK。

某些平台上无法使用原生的 NFC 功能。

某些平台上无法直接使用原生的硬件加速和高级图形渲染技术。

虽然Uni-app不能支持所有技术功能,但在WebApp跨端开发领域上已经拥有了很高的能力,同时也在不断扩展其生态和功能的支持

### 🖿 面试题 6. 简述uniApp怎么实现多端开发的?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

Uniapp是一个跨平台的开发框架,允许开发人员使用一套代码来构建多个平台的应用程序,包括iOS、Android、H5和小程序等。

以下是Uniapp实现多端开发的一些特点和方法:

#### 跨平台组件:

Uniapp提供了一组跨平台的组件和API,可以使用相同的语法和样式在不同平台上构建应用程序。例如,可以使用组件在H5和小程序中渲染相同的用户列表。

#### 条件编译:

Uniapp支持条件编译指令,允许开发人员编写特定平台的代码。例如,可以只在小程序中使用wx.showToast() API,并在其他平台中使用其他API。

#### 小程序转换:

Uniapp通过将代码转换为小程序原生代码来支持小程序平台。开发人员可以使用小程序特定的API和组件,并在跨平台应用程序中共享逻辑代码。Uniapp已经实现了H5、小程序和App平台之间的代码转换。

#### 平台差异性:

虽然Uniapp支持多种平台,但不同平台之间仍然存在一些差异。可能需要编写特定平台的 代码来解决这些问题。例如,在小程序中使用背景音乐播放时,需要注意循环播放时的限 制。

### 🛅 面试题 7. 简述uniApp如何本地存储数据?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

使用uni.setStorageSync()来进行本地存储数据,并使用uni.getStorageSync()来读取本地存储的数据。例如:

uni.setStorageSync('key', 'value');
const data = uni.getStorageSync('key');

#### 🛅 面试题 8. uniApp的生命周期描述 ?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题 ▶

### 试题回答参考思路:

### 1.应用生命周期

应用生命周期是指应用程序从启动到关闭的整个过程,包括应用程序的启动、前后台切换、 退出等。Uniapp提供了以下生命周期钩子函数:

onLaunch: 应用程序启动时触发, 仅在应用程序第一次启动时触发。

onShow: 应用程序进入前台时触发,可以获取到应用程序被打开的方式和场景值。

onHide: 应用程序进入后台时触发。

onError: 应用程序发生错误时触发,可以用来捕获和处理错误信息。

onUniNViewMessage: 监听来自nvue页面的消息。

#### 2.页面生命周期

页面生命周期是指页面从创建到销毁的整个过程,包括页面的创建、显示、隐藏和销毁等。 Uniapp提供了以下生命周期钩子函数:

onInit: 页面被初始化时触发,可以获取页面参数和数据。

onLoad:页面被加载时触发,可以进行数据初始化和网络请求等操作。onReady:页面渲染完成时触发,可以进行页面动画和交互等操作。onShow:页面被展示时触发,可以处理页面的显示效果和动画等操作。onHide:页面被隐藏时触发,可以处理页面的隐藏效果和动画等操作。onUnload:页面被销毁时触发,可以进行资源释放和清理等操作。

#### 3.组件生命周期

组件生命周期是指组件从创建到销毁的整个过程,包括组件的创建、更新、销毁等。 Uniapp提供了以下生命周期钩子函数:

beforeCreate: 组件实例被创建之前触发,此时组件的数据和方法都还没有初始化。

created:组件实例被创建之后触发,此时组件的数据和方法已经初始化。beforeMount:组件被渲染之前触发,此时组件还没有被渲染到页面上。

mounted: 组件被渲染之后触发,此时组件已经被渲染到页面上。

beforeUpdate: 组件数据更新之前触发,此时组件的数据还没有被更新。

updated: 组件数据更新之后触发,此时组件的数据已经被更新。

beforeDestroy:组件实例被销毁之前触发,此时组件的数据和方法还可以访问。destroyed:组件实例被销毁之后触发,此时组件的数据和方法已经无法访问。

## 🛅 面试题 9. 简述uniapp的配置文件、入口文件、主组件、页面管理?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

#### 试题回答参考思路:

pages.json

配置文件,全局页面路径配置,应用的状态栏、导航条、标题、窗口背景色设置等main.js

入口文件,主要作用是初始化vue实例、定义全局组件、使用需要的插件如 vuex,注意 uniapp无法使用vue-router,路由须在pages.json中进行配置。如果开发者坚持使用vue-router,可以在插件市场找到转换插件。

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。除此之外,应用生命周期仅可在App.vue中监听,在页面监听无效。

pages

页面管理部分用于存放页面或者组件

manifest.json

文件是应用的配置文件,用于指定应用的名称、图标、权限等。HBuilderX 创建的工程此文件在根目录,CLI 创建的工程此文件在 src 目录。

package.json

配置扩展。

# 🖹 面试题 10. 简述uniAPP 中常见的组件?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题▶

# 试题回答参考思路:

view: 视图容器。

icon:图标

text:文本组件。 button:按钮 image:图片 swiper:轮播图

### 🛅 面试题 11. 列举uniAPP 中常用的指令语句 ?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题 ▶

### 试题回答参考思路:

v-for: 循环渲染 (注意加: key)

v-if: 控制元素的删除添加 v-show: 控制元素的显示隐藏

v-model: 双向数据绑定 v-on: 事件绑定(简写@) v-bind: 属性绑定(简写: )

# 面试题 12. uniApp全局变量怎么定义,怎么获取?

推荐指数: ★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

在app.js中设置globalData设置,在需要的地方的js文件let app=getApp() app.globalData.数据

### 🛅 面试题 13. 简述uniApp 中的跳转方式?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

#### 试题回答参考思路:

#### 标签法:

跳转tabBar页面加open-type="switchTab"

#### 页面跳转

uni.navigateTo({url: "/路径?参数=参数值"}) 保留当前页面,只能打开非 tabBar 页面。uni.redirectTo({})关闭卸载当前页面,只能打开非 tabBar 页面tabBar跳转

uni.switchTab关闭所有非tabbar页面, 只能打开 tabBar 页面,不能url传参uni.reLaunch({})关闭卸载所有页面,可以打开任意页面

# 🖿 面试题 14. 简述uniApp项目页面之间传值?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题 ▶

#### 试题回答参考思路:

一、传单个或多个参数

方法: 直接在地址后面拼接

1、在将要跳转的页面上地址后面拼接要传递的参数,传递多个时用&符拼接 //任务列表页传递id 跳转到相应的任务详情页 gettaskList (id) {//传递多个参数时直接用&符拼接uni.navigateTo({url:'gettaskList?id=\${id}'})}

2、接收时,在页面的onload函数中接收传递过来的参数

// 任务详情页通过 onLoad 生命周期中接传递过来的参数 id onLoad(option){console.log('上一个页面传递过来的参数', option)console.log('id', option.id)console.log('item', option.item)// 接下来就是你需要该id做些什么,比如通过id查询到该详情等

二、传递对象

如需要传递的参数有很多时,由于uniapp跳转页面api 的 url 有长度限制,使用以下数据传说:

// item 为该列表的每一项的数据对象; encodeURIComponent 为uniapp 提供的api getTaskList(item) {uni.navigateTo({url: `getTaskList? item=\${encodeURIComponent(JSON.stringify(item))}`,}); }

接收时

// 同样onLoad 生命周期中进行接收, decodeURIComponent 为uniapp 提供的api onLoad(option) {const item

JSON.parse(decodeURIComponent(option.item));console.log('上一个页面传递过来的参数对象', item );// 接下来就是你需要该对象数据做些什么,当然这里你可以直接赋值给data绑定的数据this.objData = item;

注意: 传递数据的时候必须使用 JSON.stringify 将其转为 JSON 字符串, 然后接收的时候也必须使用 JSON.parse 来进行解析! 操作数组也是一样的, 因为数组也是对象

#### 🖿 面试题 15. 简述Uniapp 中组件的创建以及使用和传参方式?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题▶

#### 试题回答参考思路:

在uni-app中,可以通过创建一个后缀名为vue的文件,即创建一个组件成功,其他组件可 以将该组件通过impot的方式导入,在通过components进行注册即可

传参方法:父传子,子传父,全局,事件总线

# 🛅 面试题 16. 简述UniApp中如何发送HTTP请求?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

发送HTTP请求是许多应用程序的常见任务。在UniApp中,您可以使用uni.request API来

```
发送HTTP请求。以下是一个发送HTTP GET请求的示例代码:
uni.request({
url: 'https://jsonplaceholder.typicode.com/users',
method: 'GET',
success: function (res) {
console.log(res.data);
},
fail: function (err) {
console.error(err);
}
});
以上代码使用uni.request API发送了一个HTTP GET请求,并在成功响应时打印响应数
据,而在失败时打印错误信息。
要发送HTTP POST请求,您可以使用类似的代码:
uni.request({
url: 'https://jsonplaceholder.typicode.com/users',
method: 'POST',
data: {
name: 'John Doe',
phone: '555-555-5555',
email: 'johndoe@example.com'
},
success: function (res) {
console.log(res.data);
},
fail: function (err) {
console.error(err);
}
});
```

以上代码使用uni.request API向服务器发送了一个HTTP POST请求,并在成功响应时打印响应数据,而在失败时打印错误信息。

### 🛅 面试题 17. 简述如何使用UniApp中的组件?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

## 试题回答参考思路:

在UniApp中,您可以使用像Vue.js一样的模板语法来编写组件。以下是一个简单的UniApp组件示例代码:

以上代码定义了一个名为"my-component"的UniApp组件,该组件具有两个属性: "title"和"description"。在示例代码中,"title"是必需的属性,而"description"是可选的。如果未提供"description"属性,则该属性的默认值为"No description"。

您可以在其他UniApp页面中使用这个组件,以下是一个使用示例代码:

以上代码使用my-component组件并向它传递了"title"和"description"属性。注意,使用组件时必须在父组件中导入组件,并将其注册为局部组件。

# 🖿 面试题 18. 简述uniApp中获取DOM元素方法?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

# 试题回答参考思路:

一、通过选择器获取DOM元素

在uniapp中,最常用的获取DOM元素的方法就是通过选择器进行获取

二、通过事件回调函数获取DOM元素

在开发中,我们通常需要在用户触发某个事件之后,获取相应的DOM元素。这时,通过事件 回调函数来获取DOM元素就很方便了

三、通过ref属性获取DOM元素

在uniapp中,我们也可以通过为DOM元素设置ref属性来获取该元素

四、通过uni-app提供的组件方式来获取元素 uni-app中也提供了一种获取DOM元素的方式,那就是通过提供的组件进行获取

# 🖹 面试题 19. 简述uniApp form表单的过程?

推荐指数: ★★ 试题难度: 初级 试题类型: 原理题▶

```
试题回答参考思路:
一、提交表单到后端
在uniapp中提交表单到后端,常见的做法是使用uni.request方法进行网络请求
// js代码
Page({
submitForm: function(e) {
uni.request({
url: 'https://example.com/login',
method: 'POST',
data: e.detail.value,
success: (res) => {
console.log(res.data)
}
})
}
})
```

二、设置请求头

在实际项目开发中,我们可能需要设置请求头,比如传递用户认证信息、设置Content-Type等。这时候我们可以通过在uni.request的header参数中设置请求头

```
Page({
    submitForm: function(e) {
    uni.request({
    url: 'https://example.com/api',
    method: 'POST',
    data: e.detail.value,
    header: {'Content-Type': 'application/json'},
    success: (res) => {
    console.log(res.data)
    }
})
}
```

# 🛅 面试题 20. 简述uniApp中刷新页面的实现?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

#### 试题回答参考思路:

一、使用onPullDownRefresh实现页面刷新

在Uniapp中,我们可以使用onPullDownRefresh函数实现下拉刷新功能。当用户下拉页面时,页面就会触发onPullDownRefresh函数,我们可以在这个函数中写入刷新代码,实现

```
页面的刷新
export default {
onPullDownRefresh(){
//执行刷新操作
}
}
```

### 二、使用scroll-view组件实现页面刷新

除了下拉刷新之外,我们还可以使用scroll-view组件实现页面的刷新。scroll-view组件是Uniapp中非常常用的一个组件,可以让页面实现滚动效果。我们只需要在scroll-view组件中设置一个scroll-top属性,然后在刷新时改变这个属性的值即可实现页面的刷新

```
export default {
  data() {
  return {
    scrollTop: 0
  }
  },
  methods: {
  refreshing() {
    //执行刷新操作
  this.scrollTop = Math.random()
  }
  }
}
```

# 三、使用下拉框组件实现页面刷新

除了scroll-view组件之外,我们还可以使用下拉框组件实现页面的刷新。下拉框组件是Uniapp中非常常用的一个组件,可以让我们在页面中添加一个下拉框,在下拉框中选择某个选项后,页面就会刷新

#### 四、使用v-if指令实现页面刷新

除了上面介绍的几种方法之外,我们还可以使用v-if指令实现页面的刷新。v-if指令是Vue框架中常用的一个指令,可以让我们根据变量的值来决定页面中的某个元素是否显示。我们只需要在刷新时改变这个变量的值,就能够实现页面的刷新了

#### 五、使用setInterval函数实现页面自动刷新

在有些情况下,我们需要让页面自动刷新,以便获取最新的数据。这时,我们可以使用 setInterval函数实现页面的自动刷新。setInterval函数是JavaScript中常用的一个函数, 可以让我们重复执行某个函数

### 🛅 面试题 21. 简述uniAppim是什么?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题 ▶

uniappim是一款基于UniApp开发的即时通讯组件,能够帮助开发者快速搭建IM聊天场景。通过集成uniappim,开发者可以实现发送文本、语音、图片、视频等多种消息类型,同时还支持语音和视频通话功能。

uniappim提供了丰富的API接口和底层SDK支持,可以满足各种IM场景的开发需求。同时, uniappim还提供了完善的接入文档和示例代码,方便开发者进行快速接入和使用。

下面是uniappim的一些核心功能和特点:

支持多种消息类型,包括文本、语音、图片、视频等,且可扩展 支持语音和视频通话功能,可与腾讯云实现深度集成 提供了完善的API接口和底层SDK支持,易于开发和集成 可与UniApp框架完美融合,具备高度灵活性和可扩展性

### 🖿 面试题 22. 简述uniAPP 打包 iOS 的基础的方法流程?

推荐指数: ★★★★★ 试题难度: 高难 试题类型: 原理题 ▶

### 试题回答参考思路:

#### 一、环境准备

在进行 iOS 打包前,需要安装 Xcode 和一些相关的工具和环境。 1. 安装 Xcode Xcode 是 Apple 官方提供的开发工具,可以用来开发 macOS 和 iOS 等应用程序,同时也是打包 iOS 应用程序的必要工具。在 App Store 中下载安装即可。 2. 安装 CocoaPods CocoaPods 是一个 Swift 和 Objective—C 的依赖管理器,它负责从中央仓库中下载并管理开源代码库。在终端中输入以下指令进行安装:

sudo gem install cocoapods

3. 安装 Node.js 和 HBuilderX Node.js 是一个 JavaScript 运行环境,可以在服务器端运行 JavaScript 代码,而 HBuilderX 则是基于 Eclipse 开发的一款跨平台前端开发工具,在安装 Node.js 后,可以在官网下载 HBuilderX。

### 二、生成 iOS 项目

在进行 iOS 打包前,需要将 Uniapp 项目转换成 iOS 项目。 1. 生成 iOS 项目 在 HBuilderX 中,选择 ``网站 -> 生成移动应用``, 在弹出的对话框中选择 iOS, 点击 ``生成``按钮即可。 2. 安装插件 在生成 iOS 项目后,需要安装一些插件才能完成 iOS 打包的操作,插件的安装可以通过 CocoaPods 进行,具体步骤如下: - 进入 iOS 项目所在的根目录; - 在终端中输入 ``pod init`` 命令,生成 Podfile 文件; - 将以下代码替换到 Podfile 文件中:

platform:ios, '9.0'

target '应用名称' do pod 'AFNetworking', '~> 3.0' # 依赖库 end

- 在终端中输入 ``pod install`` 命令, 进行插件的安装。

## 三、打包 iOS 项目

在完成插件的安装后,就可以进行 iOS 打包了。 1. 配置应用程序标识符 在 Xcode 中,点

击 ``File -> Project Settings``, 在弹出的配置面板中, 选择 ``Targets -> 应用名称 ``, 在 ``General -> Identity`` 选项卡下, 填写应用程序标识符。 2. 配置签名文件 在 Xcode 中, 选择 ``Targets -> 应用名称 -> General -> Signing & Capabilities`` 选项卡下, 勾选 ``Automatically manage signing`` 选项, 然后选择对应的开发者账号和签名文件即可。 3. 编译项目 在 Xcode 中, 点击 ``Product -> Archive``, 进行项目编译。 4. 导出应用程序 将编译成功的应用程序导出,选择 ``Distribute App``, 选择对应的发布方式, 然后填写相关的信息即可导出应用程序。

#### 四、调试 iOS 应用

在完成 iOS 打包后,可能会出现应用出现闪退的情况。为了解决这个问题,可以通过以下步骤进行调试: 1. 查看日志 在 iOS 设备上,可以通过 Xcode 连接设备,然后在 ``Devices`` 窗口中选择对应的设备,查看设备日志信息和应用程序日志信息。 2. 检查权限 在 iOS 应用程序中,权限问题可能导致应用闪退。可以在应用程序中添加权限的检查逻辑,确保应用程序能够正常访问设备资源。

### 🛅 面试题 23. 简述uniAPP 弹框的组件?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题 ▶

# 试题回答参考思路:

uniapp中提供了3种不同类型的弹框组件: alert、confirm、prompt。

alert: 只有一个OK按钮,点击按钮关闭弹框。

confirm:包含确定、取消两个按钮,点击确定按钮执行对应的回调函数并关闭弹框,点击取消按钮关闭弹框。

prompt:包含输入框、确定、取消三个按钮,点击确定按钮执行对应的回调函数并关闭弹框,点击取消按钮关闭弹框

### 🖿 面试题 24. 简述uniApp异步数据流解决方案: uniapppromise?

推荐指数: ★★★★ 试题难度: 高难 试题类型: 原理题 ▶

# 试题回答参考思路:

uni-apppromise是一个解决uni-app异步数据流问题的轻量级解决方案。该方案简洁易用,能够快速解决uni-app中异步数据流处理过程中的困扰。本文将从多个方面对uniapppromise进行详细的阐述,包括其核心思想、使用方法、优缺点等。

#### 一、核心思想

uni-apppromise的核心思想是基于ES6的Promise对象,将异步数据流中的回调地狱问题进行解决,使得异步调用更为简洁,可读性更高。

### 二、使用方法

uni-apppromise 的使用方法十分简单。首先,需要在uni-app项目中引入uniapppromise.js文件。

#### 1. then方法

使用then方法可以对一个异步操作进行处理。then方法接收一个成功回调和一个失败回调, 当异步操作执行成功时,会执行成功回调;当异步操作执行失败时,会执行失败回调。

import uniApromise from '@/utils/uniapppromise.js'

```
uni.login({
success(res) {
//登录成功,使用uniapppromise处理异步流程
uniApromise.resolve(res.code)
.then(code => uni.request({ url: 'http://example.com/login', data: { code } }))
.then(res => {
console.log('登录成功', res)
})
.catch(err => {
console.log('登录失败', err)
})
}
```

2. all方法

使用all方法可以对多个异步操作进行处理。all方法接收一个数组作为参数,数组中包含多个异步操作的返回值。当所有的异步操作执行成功时,会执行成功回调;当有任意一个异步操作执行失败时,会执行失败回调。

import uniApromise from '@/utils/uniapppromise.js'

```
Promise.all([getData1(), getData2(), getData3()])
.then(results => {
  console.log('所有异步操作都执行成功', results)
})
.catch(err => {
  console.log('有异步操作执行失败', err)
})
三、优缺点
1. 优点
```

uni-apppromise解决了异步数据流回调地狱问题,使得异步代码更加简洁易懂。同时, uni-apppromise基于ES6 Promise对象,可以放心使用,不需要考虑兼容性问题。

#### 2. 缺点

uni-apppromise虽然很好的解决了异步数据流问题,但是也存在一些缺点。首先,由于uni-apppromise是基于ES6的Promise对象,所以对于没有完全支持ES6的浏览器不能使用。其次,uni-apppromise不支持取消异步操作。

### 四、总结

通过对uni-apppromise的分析,我们可以看出它的几个关键的特点。uni-apppromise最大的优点就是解决了异步数据流回调地狱问题,使得异步代码更加简洁易懂。同时,由于基

于ES6的Promise对象,开发者可以放心使用。当然,它也有一些缺点,比如不支持所有的浏览器,不支持取消异步操作。总体而言,对于uni-app开发者而言,uni-apppromise还是一个十分值得尝试的解决方案。

### 🖹 面试题 25. uni中如何为不同的平台设置不同的代码?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题 ▶

### 试题回答参考思路:

#### 使用条件注释

条件编译是用特殊的注释作为标记,在编译时根据这些特殊的注释,将注释里面的代码编译 到不同平台。

#### 条件注释的作用和使用方法

uni-app 已将常用的组件、JS API 封装到框架中,开发者按照 uni-app 规范开发即可保证多平台兼容,大部分业务均可直接满足。

但每个平台有自己的一些特性,因此会存在一些无法跨平台的情况。

大量写 if else、会造成代码执行性能低下和管理混乱。

编译到不同的工程后二次修改,会让后续升级变的很麻烦。

在 C 语言中,通过 #ifdef、#ifndef 的方式,为 windows、mac 等不同 os 编译不同的 代码。 uni-app 参考这个思路,为 uni-app 提供了条件编译手段,在一个工程里优雅的完 成了平台个性化实现

#### 条件注释定义

条件编译是用特殊的注释作为标记,在编译时根据这些特殊的注释,将注释里面的代码编译 到不同平台。

#### 条件注释的语法

写法: 以 #ifdef 或 #ifndef 加 %PLATFORM% 开头, 以 #endif 结尾。

#ifdef: if defined 仅在某平台存在

#ifndef: if not defined 除了某平台均存在

%PLATFORM%: 平台名称

#### 🖿 面试题 26. uniApp跨域问题怎么解决?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题 ▶

### 试题回答参考思路:

#### 一、uniapp跨域问题的原因

跨域指的是浏览器在向服务器发出请求时,如果当前页的协议、主机名或端口与服务器不同,就会产生跨域问题。在Web开发中,由于安全策略的存在,浏览器只允许向同源服务器

发出请求,而同源是指该服务器的协议、主机名和端口都与当前网页完全一致。 Uniapp框架是基于Vue.js进行封装的,而Vue.js有自己的跨域解决方案。但是,由于 Uniapp是一个跨平台开发框架, 所以Uniapp项目有着许多特殊的情况, 可能会使Vue.js的 跨域方案无法完全覆盖。 二、uniapp跨域问题的解决方法 在uni-config.json配置文件中进行跨域设置。 在Uniapp框架中,可以在项目的全局配置文件uni-config.json中设置跨域。具体方法是在 该文件中的"networkTimeout"字段下添加"request"字段并配置一个代理地址。 比如: { "networkTimeout": { "request": 30000, "downloadFile": 10000, "uploadFile": 10000, "connectSocket": 5000, "uploadTask": 10000, "downloadTask": 10000 } "proxy": { "/api": { "target": "https://www.example.com", "changeOrigin": true, "secure": false, "pathRewrite": { "^/api": "" } } } 上述配置中,"/api"是指这个代理地址的前缀,"target"是指被代理的地址。 "changeOrigin"字段用于控制请求头中的host是否使用被代理地址,"secure"字段用于控 制是否使用https协议,"pathRewrite"字段用于控制代理时路径的重写规则。 uni.request的header中添加'Access-Control-Allow-Origin'字段 Uniapp框架自带的网络请求API是uni.request。可以通过设置其请求头部信息来解决跨域 问题。具体方法是在请求头部信息中添加"Access-Control-Allow-Origin"字段。 例如: uni.request( { url: 'https://www.example.com/getdata', method: 'GET', header: { 'content-type': 'application/json', 'Access-Control-Allow-Origin': '\*' }

success: (res) => {
console.log(res);

}

```
fail: (err) => {
console.log(err);
}
}
);
上述代码中,"Access-Control-Allow-Origin"字段的值为"",表示允许所有域名访问该接口。如果想要指定具体的域名进行访问,就需要将""替换成具体的域名。
```

## 🖹 面试题 27. 简述uniApp 的配置文件部分?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

```
試题回答参考思路:
uniCloud ―― 云空间目录
components ―― 组件目录
comp-a.vue ―― 可复用的a组件
hybrid ―― App端存放本地html文件的目录
platforms ―― 存放各平台专用页面的目录
pages ―― 业务页面文件存放的目录
static ―― 存放本地静态资源
uni_modules ―― 存放【uni_module】规范的插件
wxcomponents ―― 存放小程序组件的目录
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
mainfest.json ―― 配置应用名称,appid,logo,版本等打包信息
page.json ―― 配置页面路由,导航条,选项卡等页面类信息
uni.scss ―― uni-app内置的常用样式变量
```

#### 🖿 面试题 28. 简述uni–app 的编译器是如何特定编译的?

推荐指数: ★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

在 web、app平台,将.vue 文件 编译成 js 代码,小程序则拆分shengchengwx,l,wxss,is等

如果涉及uts代码,安卓编译为kotlin代码,ios 编译成swift 代码。

vue2版本的编译器基于 webpack 实现, vue3通过 vite 实现, 性能更快。

同时也支持条件编译,可以指定代码至编译到特定的终端平台

### 🛅 面试题 29. uniApp 在非h5端上运行为什么要在架构上分为逻辑层和视图层?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题 ▶

### 试题回答参考思路:

主要原因是性能。web端都运行在webview里,js运算和界面渲染会抢资源导致卡顿,而小程序和app,逻辑层都独立为了单独的 js 引擎,渲染层仍然是webvbiew(app也支持原生渲染)。所以在小程序和app上不支持window,dom等API(app可以在渲染层操作window,dom)

### 🛅 面试题 30. 简述uniApp的手机端用户体验如何?

推荐指数: ★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

使用uni-app开发的微信小程序,和直接开发微信小程序相比性能没有明显差别,因为uni-app输出到微信时也是编译为wxml格式。uni-app编译到微信时用的是mpvue框架,这个是业内广泛使用的成熟框架。

uni-app打包成App后的体验和微信小程序一样好,在某些场景下更好;微信小程序的 Hybrid应用框架是业内体验上的标杆,实践证明这种体验足以承载一线互联网开发商获得上 亿用户。

uni-app内置预载机制,加载新页面速度极快,可实现无白屏极速渲染。

uni-app在App端还支持weex原生渲染(nvue),左右拖动长列表等复杂场景依然可实现高流畅性。

对比其他跨平台方案:

对于Hybrid方案, uni-app比普通基于webview的Hybrid方案体验更好,包括比DCloud之前的mui体验更好。

对比纯原生渲染的方案,体验差不多,但易用性和生态完整度上uni-app明显胜出(uni-app自身功能组件丰富,并且小程序的周边丰富生态都可以用于跨平台开发)。

就当前环境看, uni-app是功能、生态完整的跨平台方案里体验最好的。

当前的手机硬件越来越好,和以往大不相同。就像486微机时代,网页应用很难发展,肯定是原生开发的天下。但随着终端性能的提升,原生开发的必要性越来越小,加上流量获取的便利性,web开发成为主流。

# 🖿 面试题 31. uniApp 开发体验如何?支持现代前端开发流程吗?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

#### 试题回答参考思路:

uni-app 积极拥抱社区现有的现代开发流程,包括但不限于:

内置了webpack

NPM 包管理系统, 详见参考

es6+ 语法(发布时会自动编译为es5),详见参考

各种预处理器 (less、scss、stylus、typescript)

uni-app的官方ide: HBuilderX, 在vue、json、markdown、代码提示、操作效率上, 有非常明显的优势, 可帮助开发者大幅提高工作效率

uni-app同时也提供了cli方式,可使用其他开发工具开发,当然开发效率不如HBuilderX

### 🛅 面试题 32. 简述uni-app调试怎么做?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

#### 试题回答参考思路:

开发小程序时, 调试仍在微信开发者工具里

开发H5版时,在chrome里调试,推荐安装chrome的vue devtools插件。

开发App时, uni的部分(不是app特色部分),可以在chrome里调试,也可以在微信开发者工具里调试。

在App侧,支持真机运行,打log。不能审查元素和debug。

如果你在App侧使用nvue,也可以在weex在线调试器里审查元素

## 面试题 33. 简述uni-app用什么ui库?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题 ▶

## 试题回答参考思路:

首先要理解传统的h5下的vue库和传统的微信小程序ui库虽然也能在uni-app下使用,但他们无法跨6端。 想要完美跨端需要看uni-app生态下的资源。

uni-app内置组件直接用。

扩展组件是uni ui, 在组件的文档左侧。

更多ui库和模块,见插件市场: https://ext.dcloud.net.cn

基于vue的无dom库也支持,包括graceui、zanui-mpvue。这些ui可以在app、小程序、h5端均可使用。但zanui-mpvue不是纯flex的。 如果是操作dom和window的vue库,因为小程序不支持dom和window,比如elementui,无法在uni-app中使用。 基于微信小程序自定义组件的ui库也支持,比如vantui的微信小程序版。但性能不如基于vue的组件,平台覆盖方面也只支持微信小程序和App。

如果你在App侧使用nvue, 也支持weex ui

## 🖿 面试题 34. uni–app 生态开放性如何?能否直接利用现有前端社区资源?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

uni-app 提供了开放性的生态:

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, 参考

兼容微信小程序自定义组件,并且App侧通用,参考

支持 NPM 包管理系统,参考

支持 mpvue 项目及组件,参考

支持原生插件,见插件市场: https://ext.dcloud.net.cn

支持原生开发混写,可在原生工程里嵌入uni-app的sdk。

需要注意的是,uni-app 采用Vue.js方案,限制了dom操作,因此所有基于dom的前端库都无法直接使用;所幸的是目前基于微信小程序的库已经非常多了,这个生态也很完善,所有微信小程序的库在uni-app 里都能用,不止是在微信里能用,在App下也能用

# 🛅 面试题 35. uni-app 支持的手机版本最低到多少?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题 ▶

### 试题回答参考思路:

Android4.4、iOS8是官方会保障兼容的。 更低版本其实也能用,但官方发布时不会测试更低版本。

另外如果使用flex布局,更低版本是不支持的,只能使用传统css布局。

开发时尽量避免使用太新的css,可能在老款手机上产生兼容问题

# 🖿 面试题 36. 小程序的导航栏等UI相对单一,平台接口也较少,App端能自由定制这些UI吗?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题 ▶

### 试题回答参考思路:

可以

uni-app 发行到App端使用的是5+引擎,5+的所有API都可以在uni-app中可以使用;通过5+API,可以实现更为丰富的App能力,比如设置导航栏为透明渐变模式。

# 🖿 面试题 37. 已有小程序项目,如何迁移到uni–app?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题 ▶

# 试题回答参考思路:

使用【miniprogram-to-uniapp】可以将微信小程序项目转为 uni-app 项目(新版本 HBuilderX 工具已经支持各种小程序转换插件)

HBuilderX 插件地址: miniprogram-to-uniapp v2 - DCloud 插件市场

核心原理: 使用 Babel 获取AST(词法分析), 然后或使用 Babel 自带函数增删, 或正则分析替换等等操作

#### 支持的事件

- 1、支持微信、QQ、头条/抖音、支付宝/钉钉和百度等小程序转换到 uni-app 项目
- 2、支持有/无云开发的小程序项目转换为 uni-app 项目(cloudfunctions 目录将被忽略, uni-app 结合小程序云开发见:使用 uni-app 进行微信小程序云开发经验分享)
- 3、支持解析 TypeScript 小程序项目
- 4、支持解析使用 npm 模块的小程序项目
- 5、支持解析 include 标签
- 6、支持解析 template 标签
- 7、支持解析 Behavior 文件为 mixins 文件
- 8、支持 .js', .wxml 和 \*.wxss 文件进行相应转换, 并做了大量的优化
- 9、支持识别 App、Page、Component、VantComponent、Behavior 和纯 Javascript 文件的转换
- 10、修复变量名与函数重名的情况
- 11、合并使用 require 导入的 wxs 文件
- 12 setData() polyfill
- 13、搜索未在 data 声明, 而直接在 setData() 里使用的变量, 并修复
- 14、使用 jyf-parser 替换 wxParse(感谢网友"爱瑞巴勒康忙北鼻"的建议)
- 15、因 uni-app 会将所有非 static 目录的资源文件删除,因此将所有资源文件移入 static 目录,并修复所有能修复到的路径(目前 uni 编译时会将非 static 目录的文件复制一份到 static 目录,但并不完全,因此本功能仍保留)

#### 不支持的转换

- 1、不支持转换反编译后的小程序项目
- 2、不支持转换使用 uni-app 编译的小程序项目
- 3、不支持转换使用 redux 开发的小程序(代表为: 网易云信小程序 DEMO)
- 4、不支持转换使用 wxpage 开发的小程序(https://github.com/tvfe/wxpage)
- 5、不支持转换使用腾讯 omi 开发的小程序(https://github.com/Tencent/omi)
- 6、不支持转换小程序抽象节点 componentGenerics
- 7、不支持 component 里的 pageLifetimes 生命周期,请手动绕过
- 8、不支持使用 js 系统关键字作为函数或变量名(如 default、import、return、switch等)
- 9、不支持以 \\$ 开头的变量名称,如 Page( $\{data:\{sdata:\{name:"hello"\}\}\}\}$ ),刚好\\$data 是 vue 内置变量,so 不支持,需手动修复

10、不支持以动态绑定的函数		,	需手动修复
----------------	--	---	-------

11、更多,请参照 miniprogram to uniapp 工具答疑

### 使用方法

#### 第一步

win + R 在命令行里,运行【 npm install miniprogram-to-uniapp -g 】进行安装,因为这个包是工具,要求全局都能使用,所以需要 -g 进行全局安装

npm install miniprogram-to-uniapp -g

提示: 如果运行 npm 报错,请先安装 Node.js,下载地址: https://nodejs.org/zh-cn/

# 第二步

继续在命令行里,运行【wtu-V】,查看版本号,执行结果如下:

wtu -V

显示版本号,说明已经安装成功了(wtu -> 取自 wx to uni 之意,后面都用这个全局命令)

#### 第三步

在命令行里,输入【wtu-i "你的小程序项目路径"】

注意 -i 前面和后面都有空格

注意 -i 后面要使用双引号("")不能使用单引号(")

wtu -i ""

如:【wtu -i "D:\Desktop\wxmini\_demo\mini"】,回车后即可以在源项目同及目录得到一个后缀为\_uni的目录,即转换成功

#### 工具升级命令

因为工具更新比较频繁, 安装后, 可以使用如下命令进行升级

npm update miniprogram-to-uniapp -g

工具源码

仓库地址: https://github.com/zhangdaren/miniprogram-to-uniapp

### 🖿 面试题 38. App打包必须上传DCloud云端吗?代码会泄露吗?

推荐指数: ★★ 试题难度: 中级 试题类型: 原理题▶

### 试题回答参考思路:

代码可以云打包, 也可以离线打包。

即使使用云打包,DCloud也不会保留开发者的代码,云端打包完成后程序不会持久化存储相关文件。DCloud不愿也不敢沾惹这方面的麻烦。

之所以提供云打包,是方便不熟悉原生的前端工程师直接生成App安装包。包括没有mac电脑的工程师也可以打出iOS的包

#### 🛅 面试题 39. uniapp是不是基于vue研发?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题▶

### 试题回答参考思路:

Uniapp是一款基于Vue.js框架的跨平台开发工具,其特点是可以一次编写,多端发布,即只需要编写一套代码就可以发布到多个平台,包括微信小程序、支付宝小程序、H5、安卓App、IOS App等。

Uniapp所使用的语言是Vue.js,而Vue.js是当下非常流行的前端框架之一,其具有轻量、高效、易用等优点,广泛应用于各种Web应用开发中。Vue.js不仅可以进行Web开发,还可以使用Vue.js的跨平台搭建框架进行移动端应用的开发。

Uniapp结合Vue.js框架,为开发者提供了非常方便的开发工具和开发流程。开发者只需按照Vue.js的开发规范进行开发,即可轻松构建出运行在多个平台上的应用。同时,Uniapp还提供了丰富的组件和模板,可以大大简化开发的难度和速度。

Uniapp不仅是一款良好的Vue.js跨平台开发工具,还可以结合各种常见的前端框架进行使

用,比如React、AngularJS等。因此,Uniapp具有较强的可扩展性和适应性,可以适应不同开发者的需求。

除了其开发的方便性之外,Uniapp还具有快速上手和快速学习的特点。对于已经熟悉 Vue.js的前端开发者来说,上手Uniapp只需要很少的时间。对于初学者来说,Uniapp提供 了丰富的文档和教程,可以轻松入门。

总之,Uniapp的开发效率和跨平台方便性,使其成为了众多开发者和企业的首选。当下,随着跨平台应用的不断发展和普及,Uniapp也将越来越受到关注和青睐。