

ArkUI JS 框架详细讲解

ArkUI JS UI 框架是专门为 JavaScript 开发工程师设计的一套跨设备、高性能的类 Web 应用开发框架,它支持声明式编程和跨设备多态 UI,采用组件化、数据驱动模式进行 界面开发,极大地提高了 UI 的开发效率和界面逻辑的复用性。

对于前端开发人员来讲,该框架与前端目前流行的 Vue 2.0 框架的使用方式基本相同, 熟悉 Vue.js 的前端开发人员基本可以零成本转到开发鸿蒙 JS 应用程序。

3.1 ArkUI JS 框架介绍

ArkUI JS应用开发框架提供了一套跨平台的类 Web 应用开发框架,通过 Toolkit 将开发者 编写的 HML、CSS 和 JS 文件编译并打包成 JS Bundle,解析运行 JS Bundle,生成 native UI View 组件树并进行渲染显示。通过支持第三方开发者使用声明式的 API 进行应用开发,以数据驱动 视图变化,避免大量的视图操作,大大降低应用开发难度,提升开发者开发体验,如图 3.1 所示。



图 3.1 鸿蒙 ArkUI JS 框架

3.1.1 ArkUI JS 框架的特征

鸿蒙 ArkUI JS 框架的特征如下。

1. 声明式编程

鸿蒙 JS UI 框架采用类 HTML 和 CSS 声明式编程语言作为页面布局和页面样式的开发语言,页面业务逻辑则支持 ECMAScript 规范的 JavaScript 语言。JS UI 框架提供的声明式编程,可以让开发者避免编写 UI 状态切换的代码,视图配置信息更加直观。

2. 跨设备

开发框架架构上支持 UI 跨设备显示能力,运行时可自动映射到不同的设备类型,开发 者无感知,从而降低了开发者多设备适配的成本。

3. 高性能

开发框架包含了许多核心的控件,如列表、图片和各类容器组件等,针对声明式语法进行了渲染流程的优化。

3.1.2 ArkUI JS 架构介绍

ArkUI JS UI 应用框架使用 JavaScript 语言实现了一套简单的数据劫持框架,通过数据劫持实现了界面上的组件与数据的分离,实现数据驱动式界面开发,如图 3.2 所示。

Application	JS FA			
Framework	MVVM	Component	API	Router Event
			JS Engine	
Engine		Animation	Focus	EventManager
	Pipelin	Layout	Theme	Image Provider
		Renderer	Channel	External Texture
	Porting	Layer(Surface/Even	t/Vsync/Life Cycle/.)
		OS		

图 3.2 ArkUI JS 架构图

ArkUI JS 前端组件采用类似 Vue. js 2.0 框架设计模式,如图 3.3 所示。



图 3.3 MVVM 模式图

3.1.3 ArkUI JS 运行流程

ArkUI JS 框架运行流程如图 3.4 所示。



图 3.4 ArkUI JS 运行原理图

3.2 创建一个 ArkUI JS 项目

在对 HarmonyOS 应用程序有了一个初步认知之后,我们使用 DevEco Studio 来创建 一个项目,把项目运行起来,先从整体上来了解一下 HarmonyOS 项目的整体结构及开发工 具的基本使用。

环境和工具配置好后,就可以创建一个项目了。先创建一个项目,从整体上了解一下 HarmonyOS应用的整体框架。我们在一个布局里放置一个文本框,用于显示一个数字,再 添加一个按钮,每次单击按钮让文本框中的数字加1。通过这样一个小程序简单地演示工 具的使用和项目的基本框架。

3.2.1 新建 ArkUI JavaScript 项目

Ճ Create HarmonyOS Project Choose your ability template Device Ţ Ô ĝ 0 Phone Tablet Car τv Wearable Lite Wearable art Vision Template Empty Feature Ability(Java) About Feature Ability(Java) Empty Feature Ability(JS) Business Card Ability(Java) A. Help Cancel Previous Next

选择 File→New→New Project 进行项目创建,会弹出如图 3.5 所示的窗口。

图 3.5 通过模板创建鸿蒙项目

上面的创建项目窗口可分为两块,其中 Device 表示目前支持的设备。设备列表中从左 到右依次为手机、平板、车机、智慧屏、穿戴设备和轻型穿戴设备。

因为笔者使用 JavaScript 来开发,所以选择第 1 个,即 Empty Feature Ability(JS),单击 Next 按钮进入下个页面,如图 3.6 所示。

配置项目名、包名、使用的 SDK 版本及项目的保存路径后,单击 Finish 按钮即可,创建 完成后项目会自动构建,构建成功后项目的整体结构如图 3.7 所示。

Configure your project	
wint Name	
Demo1	
tackage Name	
com.charjedu.ptgamebook.demo1	
Save Location	
D:\HmGameBook\Demo1	c

图 3.6 配置项目信息



图 3.7 HarmonyOS 项目代码整体结构

3.2.2 编写界面布局

界面使用 div、text、button 3 个鸿蒙内置组件。div 表示垂直居中对齐,将 text 的字体 设置为 100px, button 默认为胶囊形状,如代码示例 3.1 所示。

```
代码示例 3.1 编写界面 HML Demo1/index.hml
<div class = "container">
<text class = "title">
{{num}}
</text>
<button @click = "updateNum">单击 + 1 </button >
</div >
```

设置页面的样式,如代码示例 3.2 所示,这里使用 FlexBox 布局,这种布局也是默认的 布局方式。

```
代码示例 3.2 编写界面样式 Demo1/index.css
```

```
.container {
    flex - direction: column;
    justify - content: center;
    align - items: center;
}
.title {
    font - size: 100px;
}
button {
    width:200px;
    height:60px;
    background - color: cadetblue;
}
```

3.2.3 编写界面逻辑代码

在 index. js 文件中编写页面逻辑代码。在 data 对象中定义一个 num 属性,定义一个 updateNum()方法。当调用 updateNum()方法后,设置 this. num++,实现 num 自增加 1, 如代码示例 3.3 所示。

代码示例 3.3	编写界面逻辑代码	Demo1/index.js
export default	: {	
data: {		
num:0		
},		
onInit()	{	

```
},
updateNum(){
    this.num++
}
```

现在可以单击 DevEco Studio 右边的 Preview 预览代码的效果,如图 3.8 所示。



图 3.8 Preview 预览代码的效果

3.2.4 通过模拟器预览效果

如果需要使用不同的模拟器进行项目测试,则首先需要登录华为开发者账号。单击 Tools→DevEco Login→Sign in,如图 3.9 所示。



图 3.9 单击 DevEco Login 登录

华为则	长号登录	
	● 请输入您的账号	
August 1	手机号/邮件地址/账号名	
1000		Ì
ALC: NOT THE OWNER OF THE OWNER OWNE OWNER OWNE	短信验证码登录	
	反臣	
著您使用华为手机,请进入"设置">"华为账号"扫码登录。	注册 忘记密码 遇到问	回题

单击 Sign in 按钮后,会自动弹出华为账号登录的页面,如图 3.10 所示。

图 3.10 登录华为开发者账号

登录成功后,需要允许 HUAWEI DevEco Studio 访问你的华为账号,如图 3.11 所示。



图 3.11 允许 DevEco Studio 访问华为账号

单击"允许"按钮后,会弹出选择虚拟设备的列表,选择需要的模拟器,单击启动按钮▶, 如图 3.12 所示。

			04				
Туре	Type	Device	Resolution	API	CPU/A8I	Status	Actions
Rhope	Ģ	τv	1920*1080	3	arm	ready	•
Q TV	Ģ	TV	1920*1060	4	arm	ready	•
 Tablet Wearable 	Q	Wearable	466*466	4	arm	ready	►
📾 Car	O	Wearable	466*466	3	arm	ready	•
		Car	1680*630	3	arm	ready	
	۵	P40	1080*2340	4	arm	ready	Þ
		MatePad Pro	1600*2560	4	arm	ready	•
🖨 Car	© #	Wearable Car P40	466*466 1680*630 1080*2340	3	arm arm arm	ready ready ready	1

图 3.12 选择模拟器

单击启动选择的模拟器后,模拟器的启动效果如图 3.13 所示。远程模拟器的使用时间 是一小时,如果超过一小时,则会自动释放模拟器,如果需要继续使用模拟器,则需要在设备 列表中重新启动模拟器。



图 3.13 P40 模拟器运行效果

启动模拟器后,单击菜单栏下面的启动按钮,如图 3.14 所示,单击启动按钮后,ide 便开始编译,编译成功后会自动打包项目并安装到模拟器上,这里如果有多个模拟器,则可以在已经连接的设备列表中选择需要使用的模拟器,然后单击 OK 按钮。

	🛔 entry	• • •	₿.	ŧ	
Select Deployment Target		/			×
Connected Devices					_
[127.0.0.1:18888] HUAWEI ANA-	AN00				
HUAWEI A	ANA-AN00				
Available Huawei Lite Devices					
🖗 Huawei Lite Wearable Simulator					
[O] Huawei Smart Vision Simulator					
Huawei Smart Vision Simulator					

等待编译成功并打包上传后,就可以在模拟器中预览上面页面的显示效果了,如 图 3.15 所示。

	P40
<div class="container"></div>	000:56:51
<text class="title"></text>	
{ { num } }	
	Demo1
<button @click="updateNum">単击+1</button>	
	5
	单击+1

图 3.15 启动编译并打包安装鸿蒙应用

图 3.14 启动编译后打包安装鸿蒙应用