

高等院校计算机应用系列教材

鸿蒙HarmonyOS NEXT 开发实践教程

苏莹 陈颖 刘云芳 主 编
祝潇威 任凯 李鹏 副主编

清华大学出版社
北 京

内 容 简 介

本书以ArkTS声明式开发为核心，系统讲解鸿蒙(HarmonyOS)应用开发全流程，构建从零基础入门到工程化落地的完整知识体系。本书共9章，内容层层递进，分为基础铺垫、核心技术、框架与实战三个阶段。开篇详细介绍鸿蒙系统、开发环境构建与工程管理，深入方舟UI框架，夯实ArkTS语法与页面布局基础；随后讲解常用与复杂UI组件，使读者掌握界面设计与交互逻辑；再深入讲解组件导航、Ability Kit应用框架，使读者掌握应用架构与生命周期管理；最后介绍ArkData数据管理、Network Kit网络服务套件，帮助读者实现全链路开发能力。

本书紧密对标鸿蒙最新技术标准，兼顾高校教学与企业实战，适合作为计算机、软件工程等相关专业教材，也可作为鸿蒙开发工程师的入门实战手册。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。举报：010-62782989，beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

鸿蒙HarmonyOS NEXT开发实践教程 / 苏莹, 陈颖,
刘云芳主编. -- 北京: 清华大学出版社, 2026. 6.
(高等院校计算机应用系列教材). -- ISBN 978-7-302-
-71907-6

I . TN929.53

中国国家版本馆CIP数据核字第2026NJ1067号

责任编辑: 刘金喜

封面设计: 高娟妮

版式设计: 思创景点

责任校对: 成凤进

责任印制: 沈 露

出版发行: 清华大学出版社

网 址: <https://www.tup.com.cn>, <https://www.wqxuetang.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-83470000 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 北京同文印刷有限责任公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 20 字 数: 517 千字

版 次: 2026 年 8 月第 1 版 印 次: 2026 年 8 月第 1 次印刷

定 价: 78.00 元

产品编号: 114179-01

当数字技术的浪潮席卷全球，操作系统作为信息技术生态的“根”，正迎来历史性的变革机遇。鸿蒙 HarmonyOS NEXT 作为面向全场景智慧时代的分布式操作系统，以其“一次开发，多端部署”的核心优势，彻底打破了设备间的技术壁垒，为开发者构建了一个覆盖手机、平板、智能穿戴、智慧家居等全场景的统一开发生态。在这一背景下，掌握鸿蒙开发技术不仅是顺应技术发展趋势的选择，更是未来开发者核心竞争力的重要组成部分。

编写本书的初衷，源于我们在鸿蒙生态教学及交流中遇到的痛点：许多开发者在面对新兴的分布式技术架构时，常因缺乏系统性的入门指南而望而却步；部分资料要么过于侧重理论讲解，缺少实战支撑，要么聚焦高级特性，忽略了基础能力的构建。为此，我们依据鸿蒙 HarmonyOS NEXT 的 latest 技术规范，结合一线项目开发经验，打造了这本“理论 + 实践”双轮驱动的入门教程，力求让零基础开发者也能快速上手，逐步成长为合格的鸿蒙应用开发者。

本书的内容体系经过精心设计，严格遵循“循序渐进、由浅入深”的认知规律，共分为9个核心章节，形成完整的技术学习闭环。

基础铺垫阶段(第1章和第2章)：作为鸿蒙开发的入门基础模块，本阶段详细讲解鸿蒙系统的特性与开发生态，逐步引导读者完成DevEco Studio环境搭建、首个ArkTS应用创建与工程管理，同时深入方舟UI框架，详解ArkTS声明式范式与线性、层叠、弹性、网格布局，帮助读者快速掌握开发工具与界面搭建基础，筑牢后续学习根基。

核心技术阶段(第3~5章)：本阶段聚焦UI开发的核心能力，在常用布局基础上，进阶讲解栅格、列表、轮播等复杂布局，全面覆盖文本、图片、按钮等基础组件，以及单选框、进度条、弹窗等复杂交互组件，帮助读者系统掌握鸿蒙应用界面的设计规范、交互逻辑梳理方法，以及多端适配的核心技巧。

框架与实战阶段(第6~9章)：本阶段围绕工程化全链路开发展开，重点介绍组件导航、页面路由等应用架构能力，深入剖析Ability Kit套件下的Stage应用模型，帮助读者掌握应用生命周期管理；同时覆盖ArkData数据管理、Network Kit网络服务套件相关内容，实现数据持久化与前后端交互开发，帮助读者完成从界面搭建到完整应用落地的全栈能力构建。

在编写过程中，我们始终坚持三大原则：一是实用性，每章均配有“习题”，帮助读者及时巩固知识点，所有示例代码均可直接运行，避免“纸上谈兵”；二是易读性，尽量避免晦涩的技术术语，对复杂概念采用“类比 + 图解”的方式解释，降低学习门槛；三是时效性，内容严格对标鸿蒙 HarmonyOS NEXT 的 latest 版本，确保技术点的准确性与前沿性。

本书适合三类读者：一是零基础的编程爱好者，希望以鸿蒙开发为切入点进入 IT 行业；二是有其他开发经验(如 Android、iOS、前端)的开发者，希望快速转型鸿蒙生态；三是高校计算机相关专业的学生，需要系统学习分布式操作系统的应用开发。无论你属于哪一类，只要跟随本

书的节奏，逐步完成每个章节的学习与实践，都能掌握鸿蒙应用开发的核心能力。

鸿蒙生态的建设不是一蹴而就的，需要每一位开发者的参与和贡献。我们希望本书能成为你进入鸿蒙世界的“第一把钥匙”，帮助你开启全场景智慧开发的新征程。在学习过程中，若遇到技术问题，建议结合鸿蒙官方文档与开发者社区的资源，通过“学练结合”的方式深化理解。

为便于教学，本书提供了教学课件、教学大纲、授课教案和案例代码，读者可通过扫描下方二维码获取相应资源。



教学课件



教学大纲



授课教案



案例代码

最后，感谢所有为鸿蒙生态建设付出努力的技术专家与开发者，也感谢你选择本书作为学习伙伴。愿你在鸿蒙开发的道路上不断成长，创造出更多优秀的全场景智慧应用！

服务邮箱：476371891@qq.com，wkservice@vip.163.com。

编 者
2026年1月

第1章 鸿蒙系统介绍	1		
1.1 初识鸿蒙系统	1		
1.1.1 鸿蒙系统介绍	1		
1.1.2 开发环境介绍	3		
1.2 鸿蒙开发环境构建	5		
1.2.1 DevEco Studio介绍	5		
1.2.2 DevEco Studio常见系统安装	7		
1.2.3 创建和运行Hello World	11		
1.3 工程管理	13		
1.3.1 工程介绍	13		
1.3.2 工程目录结构	15		
1.3.3 工程模板介绍	17		
1.3.4 创建一个新的工程	17		
1.3.5 模块管理	19		
1.3.6 构建第一个ArkTS应用(Stage模型)	25		
习题1	32		
第2章 方舟UI框架之常用布局	34		
2.1 方舟UI框架	34		
2.1.1 方舟UI框架简介	34		
2.1.2 UI开发(ArkTS声明式开发范式)概述	35		
2.2 开发布局	39		
2.2.1 布局概述	39		
2.2.2 线性布局	42		
2.2.3 层叠布局	60		
2.2.4 弹性布局	64		
2.2.5 相对布局	76		
习题2	87		
第3章 ArkUI之复杂布局	88		
3.1 栅格布局	88		
3.2 媒体查询	101		
3.3 创建列表	105		
3.4 创建网格	122		
3.5 创建轮播	128		
3.6 选项卡	133		
习题3	137		
第4章 常用UI组件	138		
4.1 文本	138		
4.1.1 文本显示(Text/Span)	138		
4.1.2 文本输入(TextInput/TextArea)	145		
4.1.3 富文本(RichEditor)	150		
4.1.4 图标小符号(SymbolGlyph/SymbolSpan)	151		
4.2 图片	155		
4.3 按钮	166		
习题4	172		
第5章 UI复杂组件应用	173		
5.1 单选框	173		
5.2 切换按钮	175		
5.3 进度条	178		
5.4 视频播放	181		
5.5 弹窗	185		
5.5.1 即时反馈(Toast)	186		
5.5.2 气泡提示(Popup)	188		
5.5.3 弹出框(Dialog)	191		
5.5.4 菜单控制(Menu)	200		
习题5	203		
第6章 组件导航和页面路由	204		
6.1 组件导航和页面路由概述	204		

6.2	组件导航(Navigation)	206
6.2.1	组件导航的使用方法	206
6.2.2	导航转场	222
6.3	页面路由(@ohos.router)	227
习题6		234

第7章

应用框架之Ability Kit(程序框架服务)

235

7.1	Ability Kit	235
7.1.1	Ability Kit简介	235
7.1.2	应用模型	236
7.2	Stage模型开发指导	237
7.2.1	Stage模型开发概述	237
7.2.2	应用配置文件概述	239
7.3	Stage模型应用组件	244
7.3.1	应用/组件级配置	244
7.3.2	UIAbility组件	246
7.3.3	AbilityStage组件容器	260
7.3.4	应用上下文Context	261
7.3.5	信息传递载体Want	263
7.3.6	组件启动规则(Stage模型)	266
习题7		267

第8章

ArkData(方舟数据管理)

269

8.1	ArkData简介	269
8.2	标准化数据定义	271
8.2.1	标准化数据定义概述	271
8.2.2	标准化数据类型	272
8.2.3	标准化数据结构	277
8.3	应用数据持久化	278
8.3.1	应用数据持久化概述	278
8.3.2	通过用户首选项实现数据持久化	279
8.3.3	通过键值型数据库实现数据持久化	282
8.3.4	通过关系型数据库实现数据持久化	286
习题8		292

第9章

Network Kit(网络服务套件)

293

9.1	Network Kit简介	293
9.2	Network Kit数据传输能力	294
9.2.1	HTTP数据请求	294
9.2.2	WebSocket连接	299
9.2.3	Socket连接	300
习题9		311

参考文献

312

第 1 章

鸿蒙系统介绍

本章学习目标

- 初识鸿蒙系统；
- 理解鸿蒙开发环境构建；
- 掌握工程管理方法；
- 掌握第一个Hello World实例。

1.1 初识鸿蒙系统

1.1.1 鸿蒙系统介绍

HarmonyOS(鸿蒙系统), 是华为技术有限公司(以下简称华为)打造的一款面向全场景的分布式操作系统。它的问世, 为智能设备的互联互通开辟了全新的道路, 有力地推动了万物互联时代的发展进程。

HarmonyOS的研发之路可追溯到2012年, 彼时华为前瞻性地开启了自有操作系统的规划, 并在芬兰赫尔辛基广纳技术人才, 为后续的技术攻坚储备力量。2016年5月, 华为消费者BG软件部正式立项研发分布式操作系统1.0版本, 经过一年的艰苦努力, 该版本成功完成研发, 同期鸿蒙内核1.0也顺利通过技术验证, 研发团队便马不停蹄地开展2.0版本的立项研发。2018年, 自研分布式操作系统项目正式落地, 并于8月24日申请“华为鸿蒙”商标。

2019年8月9日, 在美国技术封锁的严峻形势下, 华为在开发者大会上重磅发布HarmonyOS 1.0, 其首个落地产品为荣耀智慧屏, 鸿蒙系统迈出了商业化应用的第一步。2020年12月, HarmonyOS 2.0手机开发者Beta版本发布, 为智能手机全面适配HarmonyOS 2.0筑牢根基。2021年, HarmonyOS迎来爆发式发展: 5月18日智能硬件生态品牌升级, 5月25日EMUI官博更名为HarmonyOS, 6月2日HarmonyOS 2.0正式发布并推出“百机焕新计划”, 全新特性不断涌现, 用户数量迅速攀升。2022年, HarmonyOS 3.0开发者Beta版开启公测, 同年7月正式发布, 万物

互联成为显著标签，“鸿蒙世界”概念首次被提出，进一步拓展了系统的生态版图。2023年8月4日，HarmonyOS 4.0发布，具备元服务、分布式万物互联等特性，同时HarmonyOS NEXT预览版推出，标志着鸿蒙原生应用全面启动。2024年，鸿蒙生态持续壮大，众多企业积极参与，相关产品和服务不断推陈出新。

1. 系统特点

1) 分布式架构优势

HarmonyOS采用先进的分布式架构，具备强大的跨设备协同能力。在这种架构下，手机、平板、智能电视、智能手表、智能家居等不同设备能够如同一个整体般协同工作。例如，用户可以在手机上编辑文档，随后无缝切换到平板上继续操作；也可以利用手机的算力与智能电视的大屏显示能力，实现流畅的视频编辑与大屏展示。开发者基于该架构，能够更便捷地开发出跨终端的分布式应用，极大地丰富了应用的使用场景，提升了用户体验。

2) 微内核设计亮点

其微内核设计是一大核心亮点。相较于传统的宏内核，微内核仅保留了操作系统最基础、最关键的功能，如进程与线程管理、内存管理等，将其他功能以服务的形式运行在用户态。这种设计显著减少了内核的攻击面，大幅提升了系统的安全性与稳定性。当某个服务出现故障时，不会轻易导致整个系统崩溃，只会影响该服务本身，保障了系统的持续运行。同时，微内核的模块化设计使得系统更加灵活，便于根据不同设备的需求进行定制化开发，无论是资源有限的智能手表，还是功能强大的智能手机，都能适配得恰到好处。

3) 统一与弹性部署能力

HarmonyOS拥有卓越的统一与弹性部署能力，一套操作系统可适配多种设备类型。它能够依据设备的硬件资源情况，灵活调整系统的功能和性能，实现“一次开发，多端部署”。从资源受限的物联网设备，到具备强大算力和丰富功能的智能手机、平板电脑、个人计算机，HarmonyOS都能完美适配，为不同设备提供一致且流畅的用户体验，有效降低了开发者的开发成本，促进了生态系统的繁荣发展。

2. 性能表现

1) 确定时延引擎助力

确定时延引擎是HarmonyOS保障系统性能的关键技术之一。在任务执行前，该引擎会精准分配系统中任务执行的优先级与时限。例如，对于用户正在操作的应用界面响应任务，会赋予高优先级和极短的时限要求，确保应用能够快速响应用户的操作，如点击、滑动等，避免出现卡顿、延迟现象，极大地提升了用户交互的流畅性和及时性。

2) 高性能IPC提升效率

HarmonyOS的高性能进程间通信(IPC)机制极大地提高了进程间的数据传输效率。在多任务并行的场景下，不同进程之间需要频繁地进行数据交互，高性能IPC能够减少数据传输的开销和延迟，使得各个进程之间的协作更加高效。以同时运行多个应用程序为例，应用之间的数据共享和协同操作能够快速完成，保证了系统整体的流畅运行，避免因进程间通信不畅导致的系统性能下降。

3. 应用场景

1) 智能手机与平板电脑

在智能手机和平板电脑领域，HarmonyOS为用户带来了诸多创新体验。统一控制中心让

用户能够便捷地管理设备的各项功能，如网络连接、音量调节、屏幕亮度等。超级终端一触即连功能，可轻松实现与其他鸿蒙设备的连接与协同，如快速连接无线耳机、智能音箱等。万能卡片则以直观、便捷的方式展示重要信息和常用功能，用户无须打开应用即可快速获取所需内容，如天气预报、日历提醒、运动数据等，大大提升了用户操作的便捷性和效率。

2) 智能穿戴设备

对于智能穿戴设备，如智能手表、智能眼镜等，HarmonyOS提供了强大的支持。通过与手机及其他设备的协同，智能手表不仅能够实时监测用户的健康数据，如心率、睡眠质量、运动步数等，还能接收手机的通知信息，实现便捷的信息交互。在运动场景下，智能手表可与运动App协同，精准记录运动轨迹和运动数据，并在手机上生成详细的运动报告。智能眼镜则可利用HarmonyOS的分布式能力，与手机连接，实现导航信息的实时显示、语音通话等功能，为用户带来更加智能、便捷的穿戴体验。

3) 智能家居

HarmonyOS在智能家居领域发挥着核心枢纽的作用。用户可通过搭载HarmonyOS的智能终端，如手机、智能音箱等，轻松控制家中的智能灯具、智能门锁、智能摄像头、智能空调等各类智能家电设备。借助分布式技术，不同品牌、不同类型的智能家居设备能够实现互联互通、协同工作。例如，用户回家前可通过手机远程控制智能空调提前调节室内温度，到家时智能门锁自动识别并解锁，同时智能灯具自动亮起，营造舒适的家居环境。此外，用户还能根据自己的生活习惯，设置各种智能场景模式，如睡眠模式、离家模式、观影模式等，实现家居设备的智能化、自动化控制，提升生活的舒适度和便捷性。

4) 智能汽车

在智能汽车领域，HarmonyOS支持车载娱乐系统、导航系统、自动驾驶辅助系统等多个关键部分。它能够为驾驶者提供流畅、智能的驾驶体验。例如，通过与手机的无缝连接，实现手机应用在车载大屏上的投屏显示，方便驾驶者操作常用应用，如音乐播放、导航软件等。同时，HarmonyOS还能与车辆的传感器、控制系统深度集成，为自动驾驶辅助系统提供稳定、高效的运行环境，实时处理车辆行驶过程中的各种数据，保障驾驶安全。此外，基于HarmonyOS的车机系统还能与智能家居系统联动，用户在车内即可远程控制家中的智能设备，真正实现了出行与家居生活的无缝衔接。

1.1.2 开发环境介绍

HarmonyOS面向开发者发布的所有版本清单及对应文档如表1-1所示。

表 1-1 版本清单及对应文档

HarmonyOS开发者版本	DevEco Studio版本	API版本	对应的SDK发布类型 (releaseType)	文档版本
HarmonyOS 5.0.5(17)	DevEco Studio 5.0.5 Beta1	5.0.5(17)	Beta1	从5.0.3(15)版本起， HarmonyOS开发者 文档版本合一，一 套文档覆盖所有已 发布的API版本
HarmonyOS 5.0.4(16)	DevEco Studio 5.0.4 Release	5.0.4(16)	Release	
HarmonyOS 5.0.3(15)	DevEco Studio 5.0.3 Beta1 / Beta2 /Release	5.0.3(15)	Beta1 / Beta2 /Release	

(续表)

HarmonyOS开发者版本	DevEco Studio版本	API版本	对应的SDK发布类型 (releaseType)	文档版本
HarmonyOS 5.0.2(14)	DevEco Studio 5.0.2 Beta1 /Release	5.0.2(14)	Beta1 / Release	5.0.2(14)开发者文档
HarmonyOS 5.0.1(13)	DevEco Studio 5.0.1 Beta3 /Release	5.0.1(13)	Beta3 / Release	5.0.1(13)开发者文档
HarmonyOS 5.0.0(12)	DevEco Studio 5.0.0 Release	5.0.0(12)	Release	5.0.0(12)开发者文档
HarmonyOS NEXT	DevEco Studio HarmonyOS NEXT Beta1 /Release	5.0.0(12)	Release	HarmonyOS 3.1/4.0 开发者文档
	DevEco Studio HarmonyOS NEXT Developer Beta1/2/3/5/6	5.0.0(12)	Beta1/2/3/5/6	
HarmonyOS 3.1/4.0	DevEco Studio 3.1 Release	(9)	Release	

1. 版本概览

HarmonyOS 5.0.0 Release继承自HarmonyOS NEXT Release，隶属于HarmonyOS NEXT，标志着HarmonyOS NEXT经过概念阶段、开发者体验阶段、消费者体验阶段的打磨后，正式面向消费者发布。

该版本是以API 12 Release为核心的HarmonyOS全套开发套件(含SDK及开发工具DevEco Studio)。开发者可基于Release状态的开发套件进行应用开发并正式上架华为应用市场。已经基于HarmonyOS NEXT Release开发套件进行开发的应用可平滑切换至HarmonyOS 5.0.0 Release开发套件。

SDK在HarmonyOS NEXT Beta1发布时已达到API 12 Release状态，相比HarmonyOS NEXT Beta1和HarmonyOS NEXT Release，HarmonyOS 5.0.0 Release版本未改变原系统的开发能力和API接口，完全兼容原来的应用和开发代码。

HarmonyOS 5.0.0 Release完整配套信息如表1-2所示。

表 1-2 HarmonyOS 5.0.0 Release 完整配套信息

软件包	发布类型	版本号	Build Version	发布时间
系统	Release	HarmonyOS 5.0.0 Release	5.0.0.102	2024/10/22
DevEco Studio	Release	DevEco Studio 5.0.0 Release	5.0.3.910	2024/11/3
			5.0.3.906	2024/10/22
SDK	Release	HarmonyOS 5.0.0 Release SDK	基于OpenHarmony SDK ohos_sdk_public 5.0.0.71(API 12 Release)	2024/10/22

说明：

- 可通过设备的“设置 > 设备名称 > 软件版本”查询系统版本。建议在本页面检查版本更新，以及时升级至最新版本。

- DevEco Studio版本请从DevEco Studio界面菜单选择Help > About DevEco Studio进行查询。
- SDK内置在DevEco Studio中，安装DevEco Studio时会自动安装配套版本的SDK。具体版本请从DevEco Studio界面菜单选择Help > About HarmonyOS SDK进行查询。

2. DevEco Studio兼容性配套关系

DevEco Studio 5.0.3.910对应组件列表及其所支持API版本的兼容性规则如表1-3所示。

表 1-3 DevEco Studio 兼容性规则

组件	版本	说明
DevEco Studio	DevEco Studio 5.0.0 Release(5.0.3.910)	—
HarmonyOS SDK	HarmonyOS 5.0.0 Release	—
HarmonyOS Emulator	HarmonyOS 5.0.0 Release(5.0.3.805)	模拟器，当前支持手机(包括折叠屏)、平板
hvm	5.8.9	编译构建工具DevEco Hvm(以下简称Hvm)，适用于API 10及以上的工程
ohpm	5.0.8	OpenHarmony三方库的包管理工具
compatibleSdk	最低兼容版本4.0.0(10)	HarmonyOS应用/元服务兼容的最低版本
modelVersion	5.0.0	开发版本号

1.2 鸿蒙开发环境构建

1.2.1 DevEco Studio介绍

HUAWEI DevEco Studio(以下简称DevEco Studio)是基于IntelliJ IDEA Community开源版本打造的开发工具，为运行在HarmonyOS系统上的应用和元服务(以下简称应用/元服务)提供一站式的开发平台。

DevEco Studio作为一款开发工具，除了具有基本的代码开发、编译构建及调测等功能，还具有如下特点。

高效智能代码编辑：支持ArkTS、JS、C/C++等语言的代码高亮、代码智能补齐、代码错误检查、代码自动跳转、代码格式化、代码查找等功能，提升代码编写效率。

多端双向实时预览：支持UI界面代码的双向预览、实时预览、动态预览、组件预览，以及多端设备预览，便于快速查看代码运行效果。

多端设备模拟仿真：提供HarmonyOS本地模拟器，支持Phone等设备的模拟仿真，便捷获取调试环境。

DevEco Profiler性能调优：提供实时监控功能和场景化调优模板，便于进行全方位的设备资源监测，采集的数据覆盖多个维度，为开发者带来高效、直通代码行的调优体验。

1. 应用/元服务开发流程

使用DevEco Studio，只需按照图1-1所示的开发流程，即可轻松开发一个应用/元服务。



图 1-1 开发流程图

1) 开发准备

前往华为官网下载DevEco Studio安装程序，并按步骤安装开发工具。

DevEco Studio开发环境依赖于网络环境，需要连接网络才能确保工具正常使用。部分企业网络受限的情况下，需要配置代理信息。

2) 开发应用/元服务

DevEco Studio集成了Phone、Tablet、PC/2in1、Car等设备的典型场景模板，可以通过工程向导轻松地创建一个新的工程。

接下来需完成应用/元服务的UI定义、业务功能开发等编码工作，可以根据HarmonyOS应用开发概述来查看具体的开发过程，并通过API接口文档查阅需要调用的API接口。

在编码过程中，可以使用浏览器查看应用/元服务效果，浏览器支持实时预览、动态预览、双向预览等功能，能显著提升编码效率。

3) 运行、调试和测试应用/元服务

应用/元服务开发完成后，可以使用真机进行调试(需要申请调测证书进行签名)，支持单步调试、跨语言调试等调试手段，使得应用/元服务调试更加高效。

HarmonyOS应用/元服务开发完成后，在发布到应用/元服务市场前，还需要对应用进行测试，主要包含Instrument Test和Local Test两类测试，确保HarmonyOS应用/元服务纯净、安全，给用户带来更好的使用体验。

4) 发布应用/元服务

完成HarmonyOS应用/元服务的开发和测试后，需要将其发布至应用市场，以便应用市场对应用/元服务进行分发，普通消费者可以通过应用市场获取对应的HarmonyOS应用/元服务。需要注意的是，发布到华为应用市场的HarmonyOS应用/元服务，必须使用应用市场颁发的发布证书进行签名。

2. 文档声明

DevEco Studio使用指南配套DevEco Studio最新版本。若使用DevEco Studio其他版本，可能存在文档与产品功能界面及操作不一致的情况，请以实际功能界面为准。

1.2.2 DevEco Studio常见系统安装

1. 下载软件

请前往下载中心(网址: <https://developer.huawei.com/consumer/cn/download/deveco-studio>), 登录华为账号后下载DevEco Studio, 并根据下载中心页面工具完整性指导进行完整性校验, 具体版本如图1-2所示。

Windows (64-bit)	Mac (X86)	Mac (ARM)
DevEco Studio 5.0.5 Beta1		
Build Version:5.0.13.100 2025/04/27 发布		
DevEco Studio for Windows 5.0.13.100(2.2GB) ↓ SHA-256 📄 PGP ↓ 版本说明		
DevEco Studio 5.0.4 Release		
Build Version:5.0.11.100 2025/03/30 发布		
DevEco Studio for Windows 5.0.11.100(2.2GB) ↓ SHA-256 📄 PGP ↓ 版本说明		
DevEco Studio 5.0.3 Release		
Build Version:5.0.9.300 2025/03/18 发布		
DevEco Studio for Windows 5.0.9.300(2.2GB) ↓ SHA-256 📄 PGP ↓ 版本说明		
DevEco Studio 5.0.3 Beta2		
Build Version:5.0.9.200 2025/03/07 发布		
DevEco Studio for Windows 5.0.9.200(2.2GB) ↓ SHA-256 📄 PGP ↓ 版本说明		
DevEco Studio 5.0.3 Beta1		
Build Version:5.0.9.100 2025/03/03 发布		
DevEco Studio for Windows 5.0.9.100(2.2GB) ↓ SHA-256 📄 PGP ↓ 版本说明		
DevEco Studio 5.0.2 Release		
Build Version:5.0.7.210 2025/02/17 发布		
DevEco Studio for Windows 5.0.7.210(2.2GB) ↓ SHA-256 📄 PGP ↓ 版本说明		
DevEco Studio 5.0.2 Release		
Build Version:5.0.7.200 2025/01/27 发布		
DevEco Studio for Windows 5.0.7.200(2.2GB) ↓ SHA-256 📄 PGP ↓ 版本说明		

图 1-2 DevEco Studio 版本一览

2. 安装DevEco Studio

DevEco Studio支持Windows和macOS系统, 下面将针对这两种操作系统的软件安装方式分别进行介绍。

1) Windows环境

(1) 运行环境要求。

为保证DevEco Studio正常运行, 建议计算机配置满足如下要求。

- 操作系统: Windows10 64位、Windows11 64位。
- 内存: 16GB及以上。
- 硬盘: 100GB及以上。
- 分辨率: 1280×800像素及以上。

(2) 安装过程。

① 下载完成后, 双击下载的“deveco-studio-xxxx.exe”, 进入DevEco Studio安装向导。如图1-3所示, 选择安装目录为C:\Program Files\Huawei\DevEco Studio, 也可以单击“浏览”按钮指定其他安装目录, 然后单击“下一步”按钮。



图 1-3 选择安装位置

② 按图1-4所示操作完成DevEco Studio设置，单击“下一步”按钮，直至安装完成。

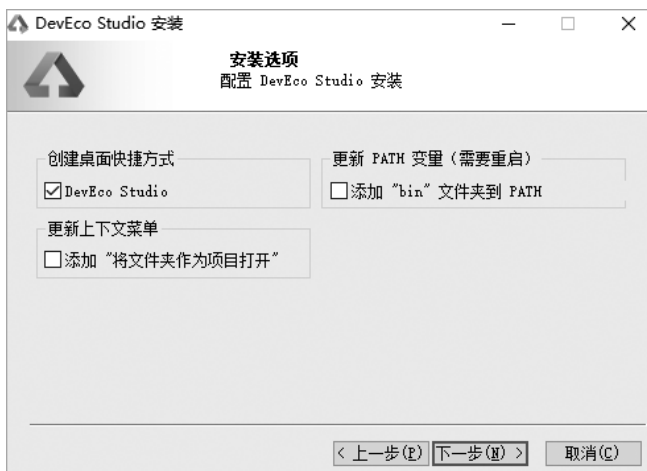


图 1-4 安装选项

③ 安装完成后，单击“完成”按钮完成安装，结果如图1-5所示。



图 1-5 安装完成

说明：

- DevEco Studio提供开箱即用的开发体验，将HarmonyOS SDK、Node.js、Hvigor、OHPM、模拟器平台等工具进行合一打包，简化了DevEco Studio的安裝配置流程。
- HarmonyOS SDK已嵌入DevEco Studio中，无须额外下载配置，可在DevEco Studio安装目录下的DevEco Studio\sdk中查看。如需进行OpenHarmony应用开发，可通过Settings > OpenHarmony SDK页签下下载OpenHarmony SDK。

2) macOS环境

(1) 运行环境要求。

为保证DevEco Studio正常运行，建议计算机配置满足如下要求。

- 操作系统：macOS(X86) 11/12/13/14、macOS(ARM) 12/13/14。
- 内存：8GB及以上。
- 硬盘：100GB及以上。
- 分辨率：1280×800像素及以上。

(2) 安装过程。

① 在安装界面中，将DevEco-Studio.app拖拽到Applications中，等待安装完成，如图1-6所示。

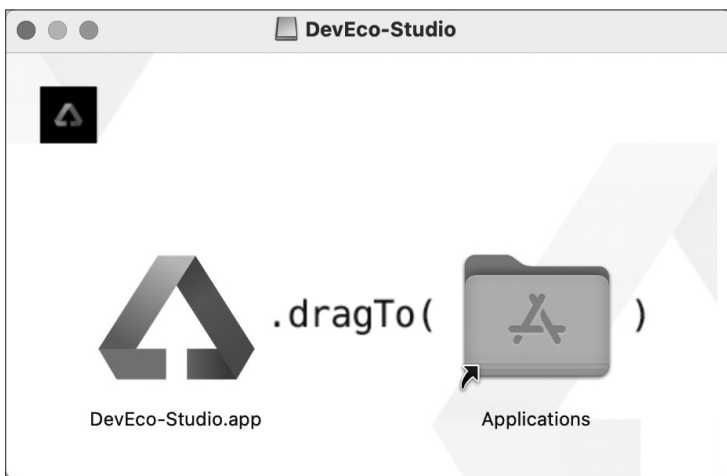


图 1-6 安装界面

② 安装完成后，请根据代理配置，检查和配置开发环境。

3. 诊断开发环境

为了获得良好的开发应用/元服务的体验，DevEco Studio提供了开发环境诊断功能，帮助用户识别开发环境是否完备。用户可以在图1-7所示的欢迎页面中单击Diagnose按钮进行诊断。如果用户已经打开了工程开发界面，也可以在菜单栏中单击Help > Diagnostic Tools > Diagnose Development Environment进行诊断。

4. 启用中文插件

(1) 单击File > Settings (macOS为DevEco Studio > Preferences) > Plugins，选择Installed页签，在搜索框中输入“Chinese”，搜索结果为中国语(简体)，在右侧单击Enable按钮，再单击OK按钮，如图1-8所示。

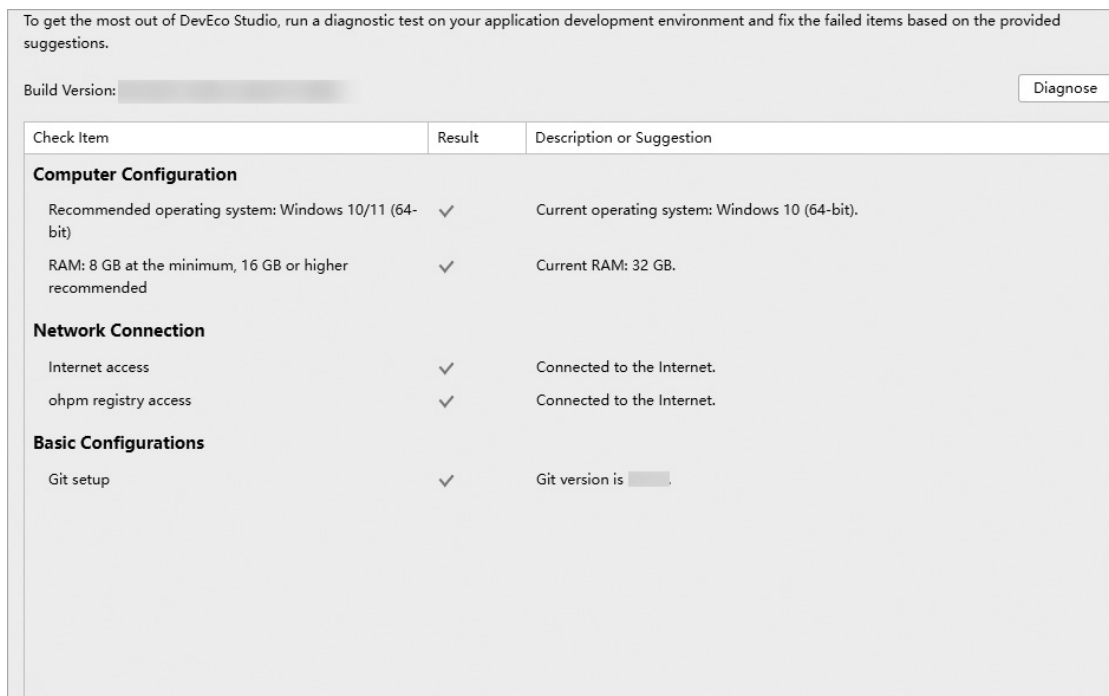


图 1-7 欢迎页面

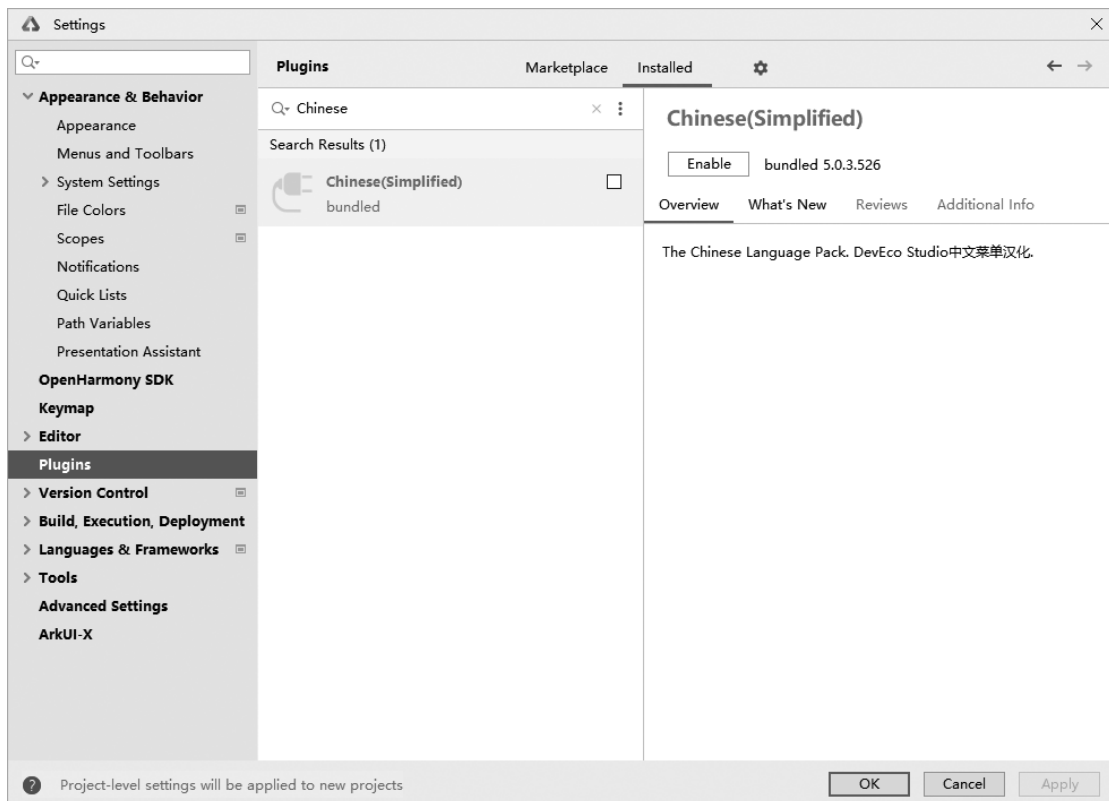


图 1-8 启用中文插件

(2) 在图1-9所示的弹窗中单击Restart按钮，重启DevEco Studio后即可生效。

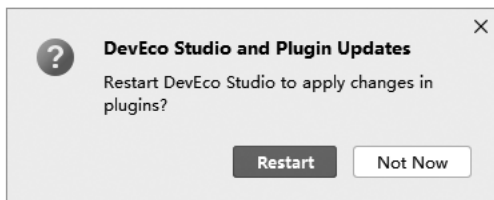


图 1-9 弹窗

1.2.3 创建和运行Hello World

DevEco Studio安装完成后，可以通过运行Hello World工程来验证环境设置是否正确。接下来以创建一个支持Phone设备的工程为例进行介绍。

1. 创建一个新工程

- (1) 打开DevEco Studio，在欢迎页面单击Create Project，创建一个新工程。
- (2) 根据工程创建向导，选择创建Application或Atomic Service。选择Empty Ability模板，然后单击Next按钮，如图1-10所示。



图 1-10 工程类型选择

- (3) 填写完工程相关信息后，单击Finish按钮，如图1-11所示。

工程创建完成后，DevEco Studio会自动进行工程的同步。

2. 运行Hello World

- (1) 将搭载HarmonyOS系统的真机与计算机连接。具体连接步骤及驱动安装要求，可访问华为开发者联盟官网搜索“运行应用/元服务”。温馨提示：真机调试前请务必先在IDE中登录华为账号并完成自动签名(Build > HAP/APP Signature)，否则无法安装成功。

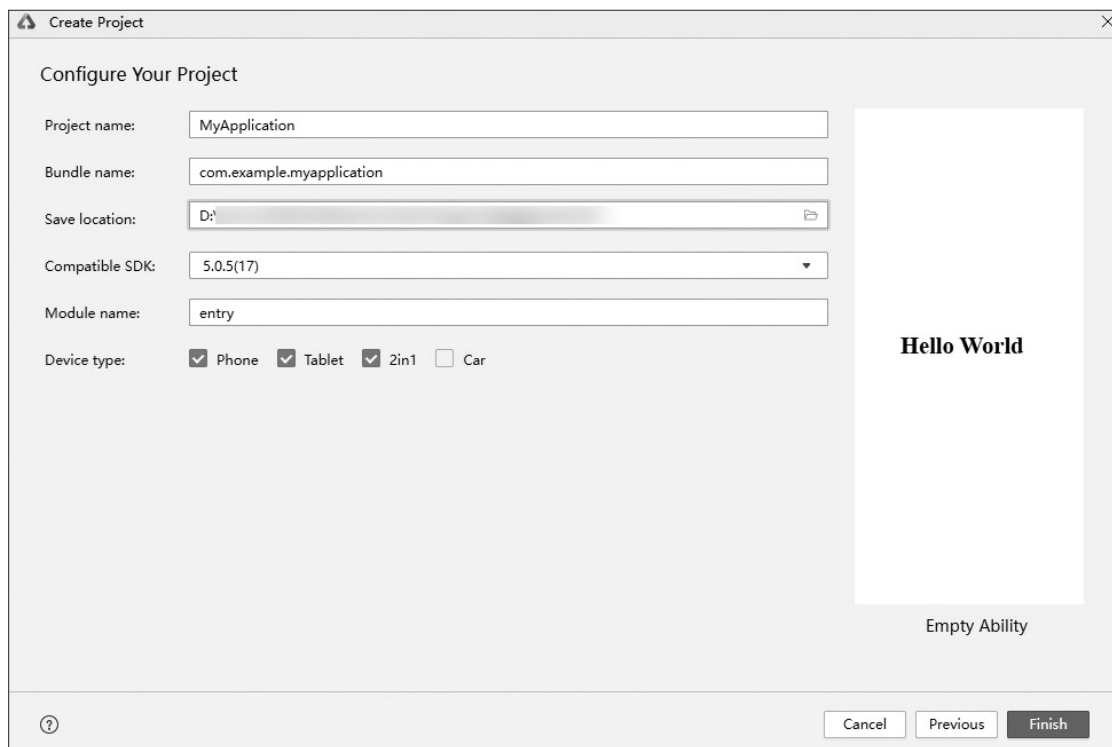


图 1-11 工程信息填写

(2) 单击File > Project Structure... > Project > Signing Configs，勾选Support HarmonyOS和Automatically generate signature复选框，单击界面提示的Sign In，使用华为账号登录。等待自动签名完成后，单击OK按钮即可，如图1-12所示。

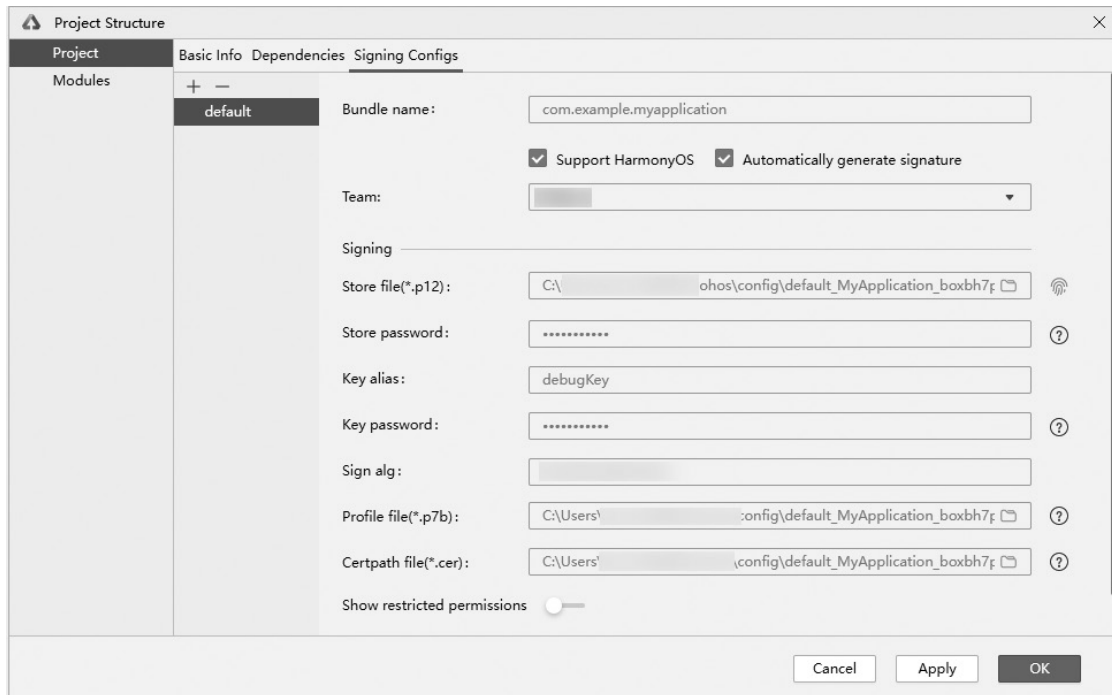


图 1-12 工程运行配置

(3) 单击编辑窗口右上角工具栏中的  按钮运行应用。运行结果如图1-13所示。

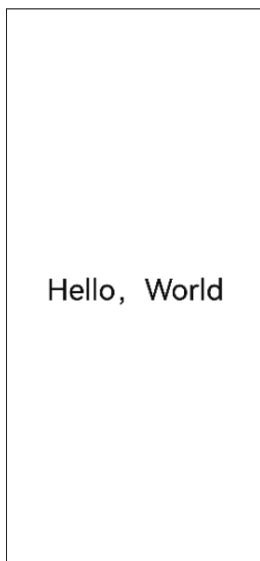


图 1-13 运行结果

1.3 工程管理

1.3.1 工程介绍

1. App包结构

在进行应用/元服务开发前，开发者应该掌握应用/元服务的逻辑结构。

应用/元服务的发布形态为App Pack(Application Package)，它由一个或多个HAP(Harmony Ability Package)包及描述App Pack属性的pack.info文件组成。

一个HAP在工程目录中对应一个Module，它由代码、资源、三方库及应用/元服务配置文件组成。HAP可以分为Entry和Feature两种类型。

- **Entry**：应用的主模块，作为应用的入口，提供了应用的基础功能。
- **Feature**：应用的动态特性模块，作为应用能力的扩展，可以根据用户需求和设备类型进行选择性安装。

基于Stage模型和FA模型开发的应用，其应用程序包结构并不相同。

Stage模型应用程序包结构如图1-14所示。

FA模型应用程序包结构如图1-15所示。

2. 切换工程视图

DevEco Studio工程目录结构提供工程视图和Ohos视图两种展示形式。工程视图展示工程中实际的文件结构；Ohos视图会隐藏编码过程中不常用的文件，并将常用文件进行重组展示，方便开发者查询或定位需编辑的模块或文件。

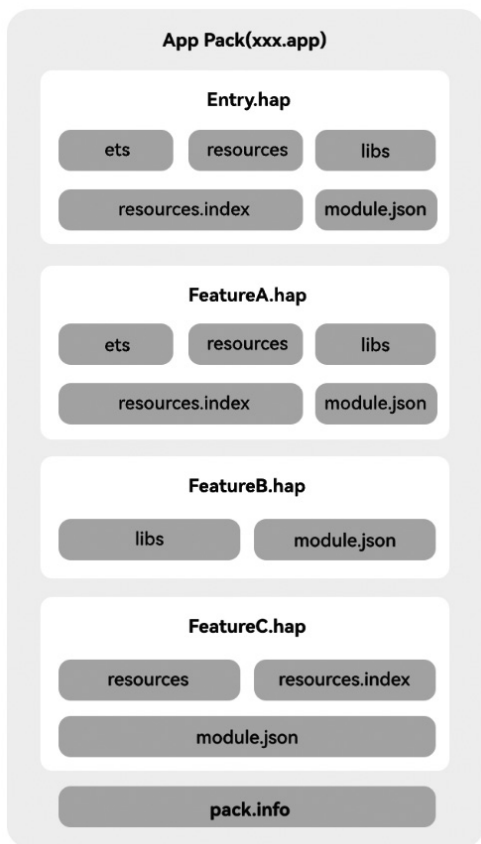


图 1-14 Stage 模型应用程序包结构

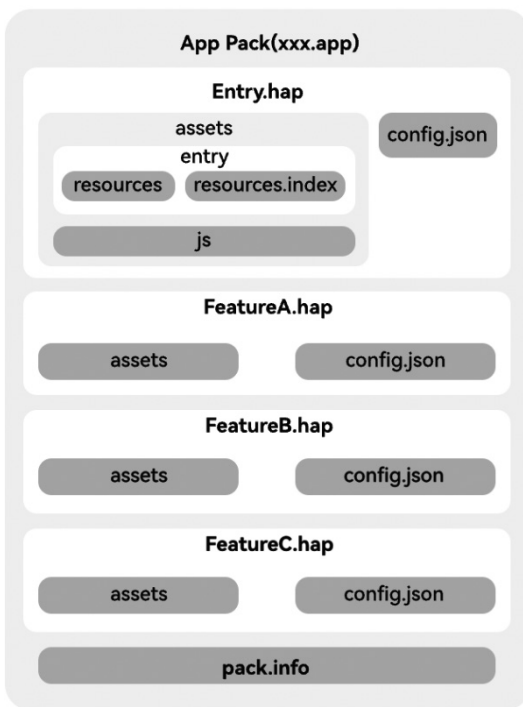


图 1-15 FA 模型应用程序包结构

工程创建或打开后，默认显示工程视图。若要切换为Ohos视图，则可单击Project > Ohos进行切换，如图1-16所示。

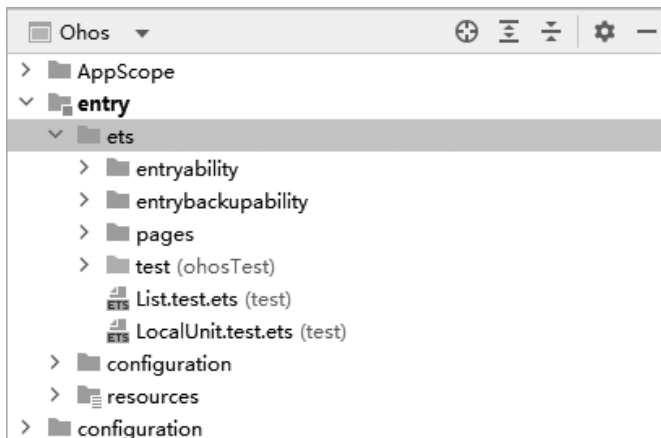


图 1-16 视图切换

1.3.2 工程目录结构

1. ArkTS工程目录结构(Stage模型)

ArkTS Stage模型支持API Version 10及以上版本，其工程目录结构如图1-17所示。

相关包作用及信息如下。

- AppScope > app.json5: 应用的全局配置信息。
- entry: 应用/元服务模块，编译构建后生成一个HAP。
 - src > main > ets: 用于存放ArkTS源码。
 - src > main > ets > entryability: 应用/元服务的入口。
 - src > main > ets > pages: 应用/元服务包含的页面。
 - src > main > resources: 用于存放应用/元服务模块所使用的资源文件，如图形、多媒体、字符串、布局文件等，具体说明如表1-4所示。

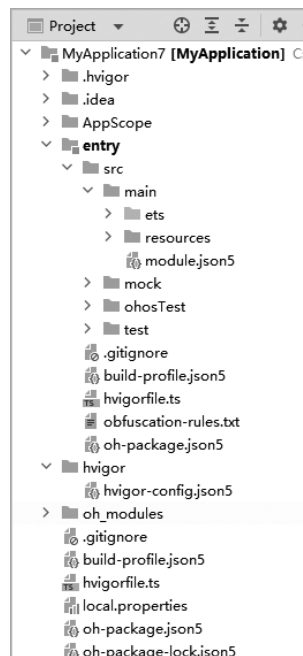


图 1-17 ArkTS 工程目录结构

表 1-4 资源目录及文件说明

资源目录	资源文件说明
base>element	包括字符串、整型数、颜色、样式等资源的json文件。每个资源均由json格式进行定义，例如： boolean.json: 布尔型 color.json: 颜色 float.json: 浮点型 intarray.json: 整型数组 integer.json: 整型 pattern.json: 样式 plural.json: 复数形式 strarray.json: 字符串数组 string.json: 字符串值
base>media	多媒体文件，如图形、视频、音频等，支持的文件格式包括.png、.gif、.mp3、.mp4等
rawfile	用于存储任意格式的原始资源文件。rawfile不会根据设备的状态去匹配不同的资源，需要指定文件路径和文件名进行引用

- src > main > module.json5: Stage模型模块配置文件，主要包含HAP的配置信息、应用在具体设备上的配置信息，以及应用的全局配置信息。
- build-profile.json5: 当前的模块信息、编译信息配置项，包括buildOption、targets配置等。
- hvmfile.ts: 模块级编译构建任务脚本。

- oh-package.json5: 描述三方包的包名、版本、入口文件(类型声明文件)和依赖项等信息。
- oh_modules: 用于存放三方库依赖信息, 包含应用/元服务所依赖的第三方库文件。
- build-profile.json5: 应用级配置信息, 包括签名、产品配置等。
- hvmigorfile.ts: 应用级编译构建任务脚本。
- oh-package.json5: 描述全局配置, 如依赖覆盖(overrides)、依赖关系重写(overrideDependencyMap)和参数化配置(parameterFile)等。

2. C++工程目录结构(Stage模型)

C++ Stage模型支持API Version 10以上版本, 支持使用ArkTS+C++进行开发, 其工程目录结构如图1-18所示。

- entry: 应用模块, 编译构建后生成一个HAP。
 - src > main > cpp > types: 用于存放C++的API接口描述文件。
 - src > main > cpp > types > libentry > index.d.ts: 描述C++ API接口行为, 如接口名、入参、返回参数等。
 - src > main > cpp > types > libentry > oh-package.json5: 配置.so三方包声明文件的入口及包名。
 - src > main > cpp > CMakeLists.txt: CMake配置文件, 提供CMake构建脚本。
 - src > main > cpp > napi_init.cpp: 定义C++ API接口的文件。
 - src > main > ets: 用于存放ArkTS源码。
 - src > main > resources: 用于存放应用所使用的资源文件, 如图形、多媒体、字符串、布局文件等, 其作用同表1-4一致。
 - src > main > module.json5: Stage模块配置文件, 主要包含HAP的配置信息、应用在具体设备上的配置信息, 以及应用的全局配置信息。
 - build-profile.json5: 当前的模块信息、编译信息配置项, 包括buildOption、targets配置等。
 - hvmigorfile.ts: 模块级编译构建任务脚本。
- build-profile.json5: 应用级配置信息, 包括签名、产品配置等。
- hvmigorfile.ts: 应用级编译构建任务脚本。

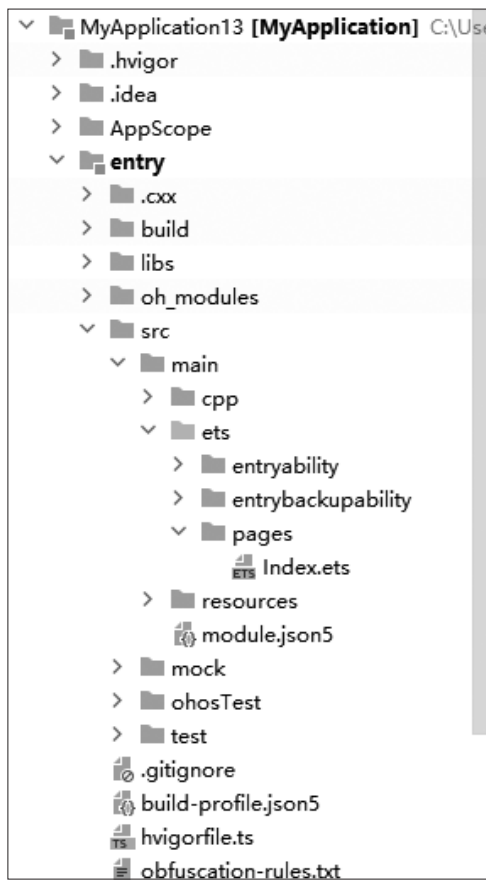


图 1-18 C++ 工程目录结构

1.3.3 工程模板介绍

DevEco Studio支持多品类应用/元服务开发，预置丰富的工程模板(见图1-19)，有助于开发者根据工程向导轻松创建适配各类设备的工程，并自动生成对应的代码和资源模板。同时，DevEco Studio还提供了多种编程语言供开发者进行应用/元服务开发，包括ArkTS、JS和C/C++。



图 1-19 工程模板

工程模板说明如表1-5所示。

表 1-5 工程模板说明

模板名称	说明
Empty Ability	用于Phone、Tablet、2in1、Car设备的模板，展示基础的Hello World功能
Native C++	用于Phone、Tablet、2in1、Car设备的模板，作为应用调用C++代码的示例工程，界面显示“Hello World”
[CloudDev]Empty Ability	端云一体化开发通用模板
[Lite]Empty Ability	用于Lite Wearable设备的模板，展示基础的Hello World功能。开发者可基于此模板修改设备类型及RuntimeOS，进行小型嵌入式设备开发
Flexible Layout Ability	用于创建跨设备应用开发的三层工程结构模板。三层工程结构包含common(公共能力层)、features(基础特性层)、products(产品定制层)
Embeddable Ability	用于开发支持被其他应用嵌入式运行的元服务的工程模板

1.3.4 创建一个新的工程

在开发应用/元服务前，需先根据工程创建向导创建一个新的工程，工具会自动生成对应的代码和资源模板。

说明：

在运行DevEco Studio工程时，建议每个运行窗口预留2GB以上的可用内存空间。

1. 创建HarmonyOS工程

(1) 通过以下两种方式打开工程创建向导界面。

- 如果当前未打开任何工程，则可以在DevEco Studio的欢迎页面中单击Create Project创建一个新工程。
- 如果已经打开了工程，则可以在菜单栏中单击File > New > Create Project创建一个新工程。

(2) 根据工程创建向导，选择创建Application或Atomic Service，再选择需要的Ability工程模板，然后单击Next按钮，如图1-20所示。

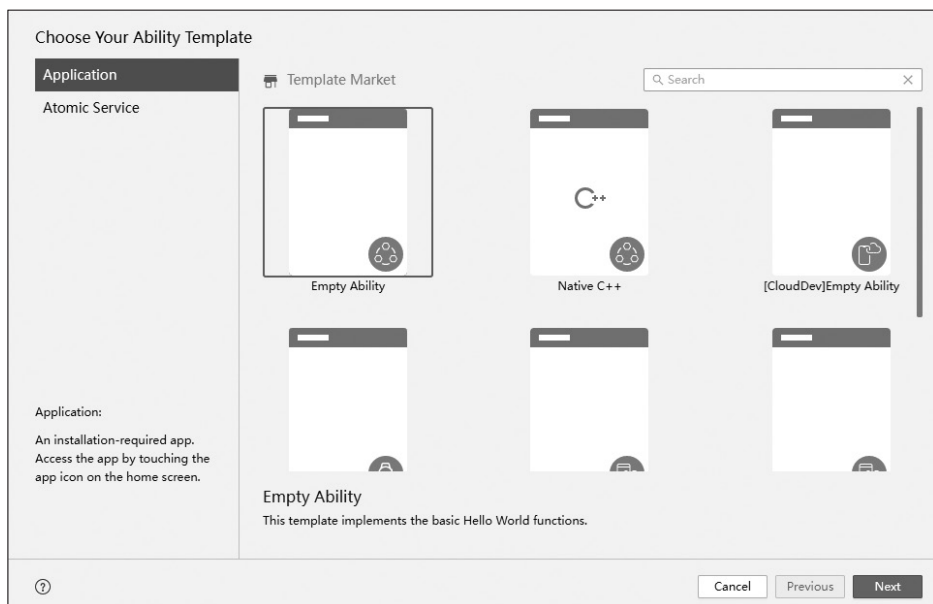


图 1-20 工程创建向导

说明：

- 从API 11版本开始支持Atomic Service元服务工程开发。
- Atomic Service元服务工程暂不支持Native开发。

(3) 在工程配置页面，需根据向导配置工程的基本信息，如图1-21所示。

- **Project name:** 工程的名称，可以自定义，由大小写字母、数字和下划线组成。
- **Bundle name:** 应用的包名，用于标识应用的唯一性。

说明：

应用包名需满足以下要求。

- 必须为以点号(.)分隔的字符串，且至少包含三段，每段中仅允许使用英文字母、数字、下划线(_)，如com.example.myapplication。
- 首段以英文字母开头，非首段以数字或英文字母开头，每段以数字或英文字母结尾，如com.01example.myapplication。

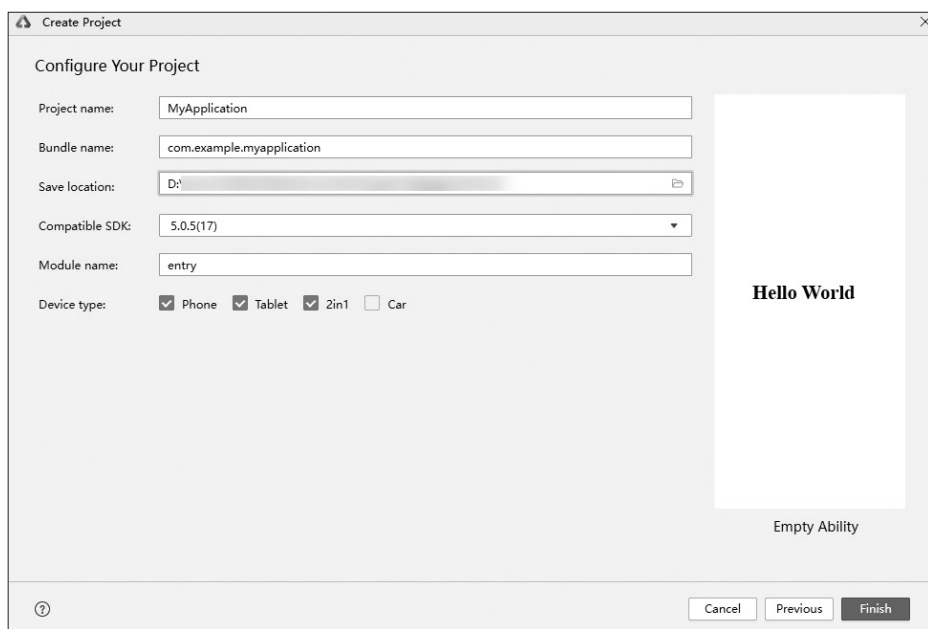


图 1-21 工程配置

- 不允许多个点号(.)连续出现，如com.example..myapplication。
- 长度需为7~128个字符。

Save location: 工程文件本地存储路径，由大小写字母、数字和下划线等字符组成，不可包含中文字符。

Compatible SDK: 兼容的最低API Version。

Module name: 模块的名称。

Device type: 该工程模板支持的设备类型。

(4) 单击Finish按钮，工具会自动生成示例代码和相关资源，等待工程创建完成。

1.3.5 模块管理

Module是应用/元服务的基本功能单元，包含源代码、资源文件、第三方库及应用/元服务配置文件，每个Module都可以独立编译和运行。一个应用/元服务通常会包含一个或多个Module，因此，可以在工程中创建多个Module。Module分为Ability和Library两大类型，其中Ability型包含entry、feature两种子类型，Library型包含har、shared两种子类型。

1. 在工程中添加Module

(1) 创建新的Module。可通过以下三种方法在工程中创建新的Module。

方法1：将鼠标移到工程目录顶部，右击，选择New > Module...，即可创建新的Module，此时该Module将生成在工程根目录下。

方法2：选中工程目录中的任意文件，然后在菜单栏中选择File > New > Module...，即可创建新的Module，此时该Module将生成在工程根目录下。

方法3：先在工程根目录下创建一个新的目录，再在该目录下右击，选择New > Module...，即可创建新的Module，此时该Module将生成在该目录下，方便开发者对模块进行分类管理。

说明：

当前暂不支持在AppScope、hvmigor、oh_modules、build，以及以点开头的目录(如.hvmigor、.idea)下通过右击的方式创建Module。

(2) 在New Project Module界面中，选择需要创建的模板，单击Next按钮，如图1-22所示。

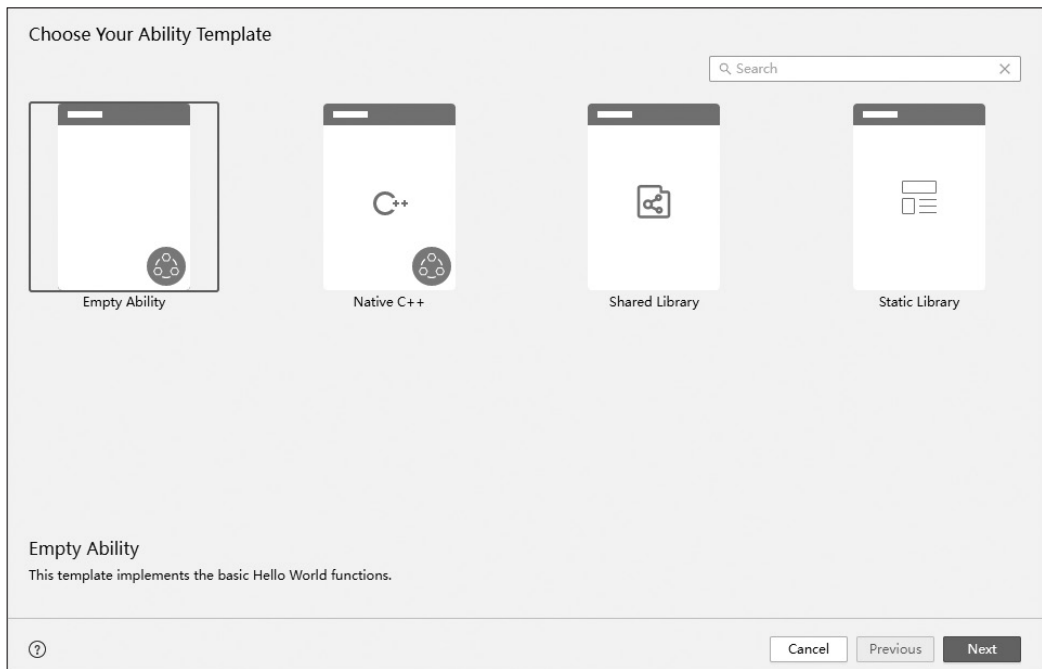


图 1-22 New Project Module 界面

(3) 在Module配置页面，设置新建Module的基本信息，然后单击Next按钮，如图1-23所示。

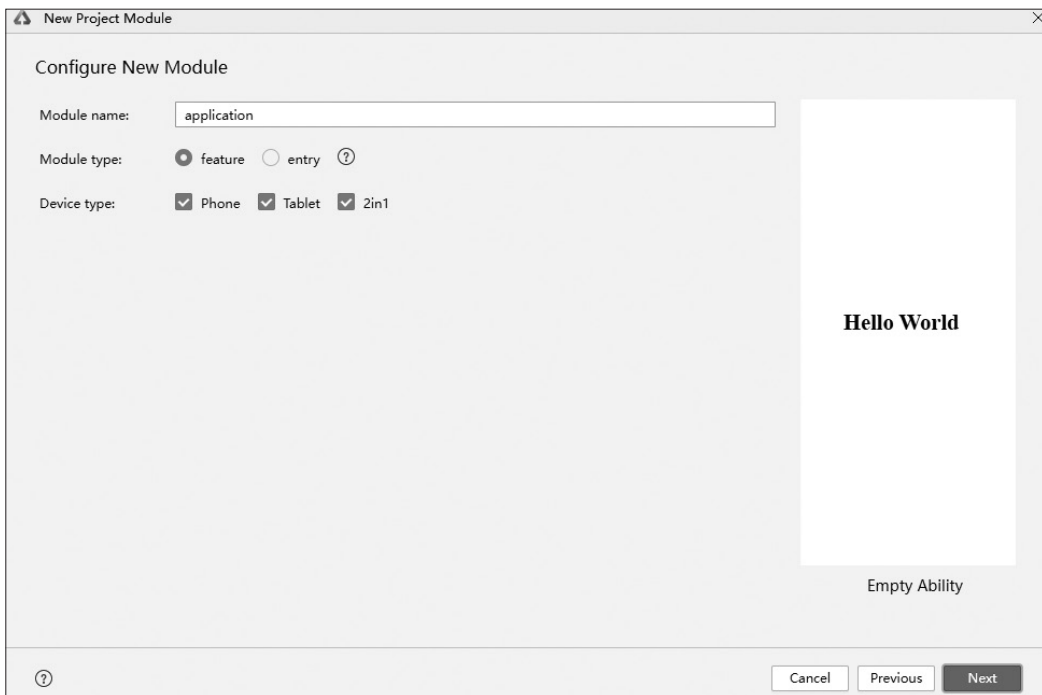


图 1-23 Module 配置页面

- **Module name:** 新建模块的名称，不可与工程名称相同。
- **Module type:** 仅Ability模板包含该配置项，可以选择feature或entry类型。

说明：

- 同一工程中，通过新建Module的方式仅支持创建一个entry模块。若需新建entry类型模块，可在module.json5文件中修改相应Module下的type字段。
- 如果同一类型的设备已经存在entry模块，新建entry模块后，还需要配置distroFilter分发规则。
- **Device type:** 选择模块适配的设备类型。如果新建模块的Module type为feature，则只能选择该工程原有的设备类型；如果Module type为entry，则可以选择该Module支持的其他设备类型。
- **Enable native:** 仅Library模板包含该配置项，将创建一个可以调用C/C++的共享包。

(4) 若该Module的模板类型为Ability，还需要设置新建Ability的Ability name和Exported参数。Exported参数用于设置该Ability是否可以被其他应用/元服务所调用(FA模型下为Visible参数)。

- 勾选(true)：可以被其他应用/元服务调用。
- 不勾选(false)：不可以被其他应用/元服务调用。

(5) 单击Finish按钮，等待Module创建完成后，可以在工程目录中查看和编辑新建的Module。工程中所有模块的信息可以在build-profile.json5的module字段中进行配置。

2. 导入Module

HarmonyOS工程支持导入其他HarmonyOS模块，但仅支持同模型间导入：FA模型的模块可导入至FA模型，Stage模型的模块可导入至Stage模型。不支持FA模型的模块导入至Stage模型或Stage模型的模块导入至FA模型。

说明：

DevEco Studio支持引用当前工程目录之外的Module(即其他工程下的Module)。除通过Import Module方式导入模块外，还可在build-profile.json5文件的srcPath字段中配置工程外Module的相对路径实现导入。通过srcPath方式导入工程，仅引用Module相关信息，不会将Module代码完全复制至本地。

(1) 在菜单栏中单击File > New > Import... > Import Module，如图1-24所示。

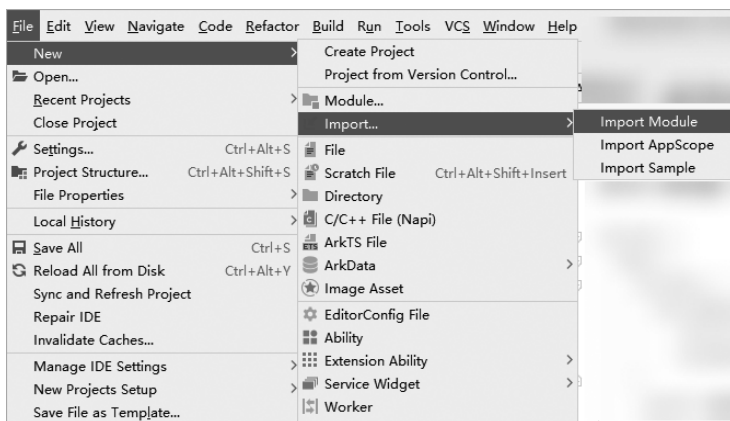


图 1-24 其他 HarmonyOS 模块导入

(2) 选择导入的模块。在指定路径下选择导入的模块，单击OK按钮，如图1-25所示。导入的模块可以为文件夹，也可以为zip格式。

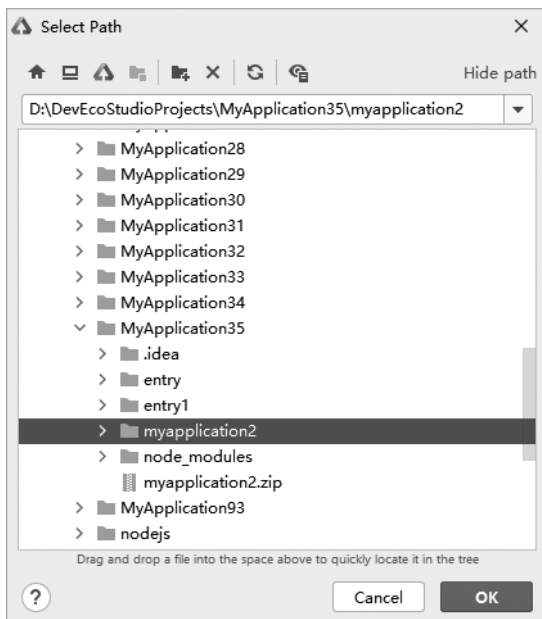


图 1-25 模块导入

3. 删除Module

在工程目录中选中要删除的模块，右击，选择Delete选项，并在弹出的对话框中单击Delete按钮。

4. 在模块中添加Ability

Ability是应用/元服务所具备的能力的抽象，一个Module可以包含一个或多个Ability。应用/元服务先后提供了两种应用模型。

- FA(Feature Ability)模型：API 7开始支持的模型，目前已不再作为主推模型。
- Stage模型：API 9新增的模型，是目前主推且会长期演进的模型。在该模型中引入了AbilityStage、WindowStage等类作为应用组件和Window窗口的“舞台”，因此这种应用模型称为Stage模型。

Stage模型包含两种Ability组件类型。

- UIAbility组件：包含UI界面，具备UI展示能力，主要用于与用户交互。
- ExtensionAbility组件：具备特定场景的扩展能力，可满足更多的使用场景。

1) 在Stage模型中添加UIAbility

(1) 选中对应的模块，右击，选择New > Ability。

(2) 设置Ability名称，并确定是否在设备主屏幕上显示该功能的启动图标，单击Finish按钮完成Ability创建，如图1-26所示。

2) 在模块中添加Extension Ability

(1) 在工程中选中对应的模块，右击，选择New > Extension Ability，选择不同的场景类型。当前仅Application工程支持创建Extension Ability，如图1-27所示。

- EmbeddedUIExtensionAbility：提供跨进程界面嵌入能力。

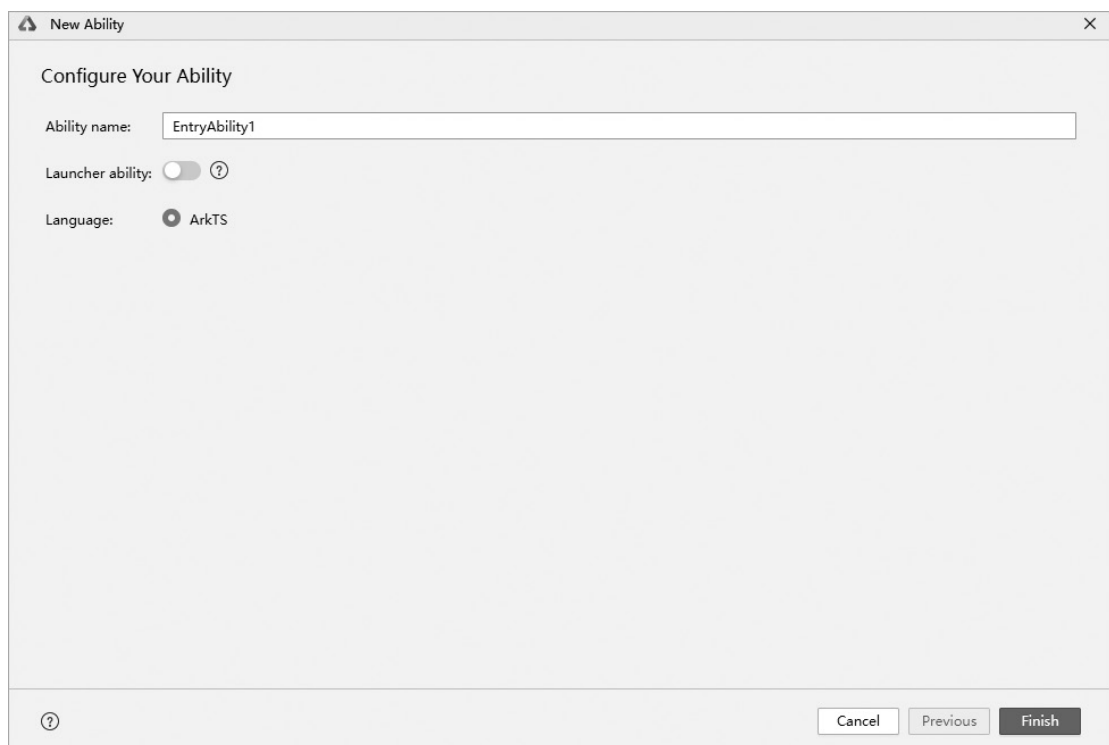


图 1-26 创建 Ability

- BackupAbility: 用于实现应用数据的备份与恢复。
- WorkScheduler: 提供延迟任务的回调执行能力。
- RemoteNotificationAbility: 提供场景化消息数据获取和生命周期销毁回调的通知能力。

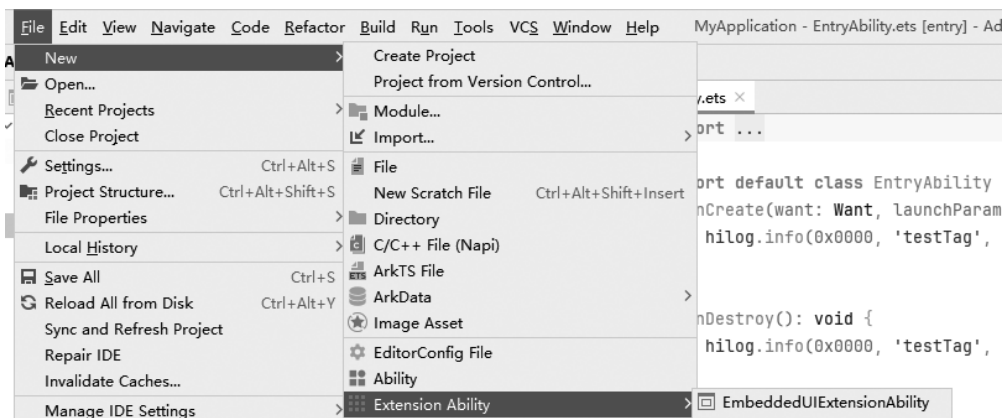


图 1-27 添加 Extension Ability

(2) 设置Ability名称，单击Finish按钮完成Extension Ability的创建，如图1-28所示。

5. 添加Page

在基于ArkTS语言的工程中，支持添加Page。Page是表示应用/元服务的一个页面。应用/元服务可以设计为多个功能页面，每个页面进行单独的文件管理，并通过路由API实现页面间的调度管理，以实现应用内的功能解耦。在基于ArkTS语言的工程中添加Page后，会在pages文件夹下生成一个新的ets文件。

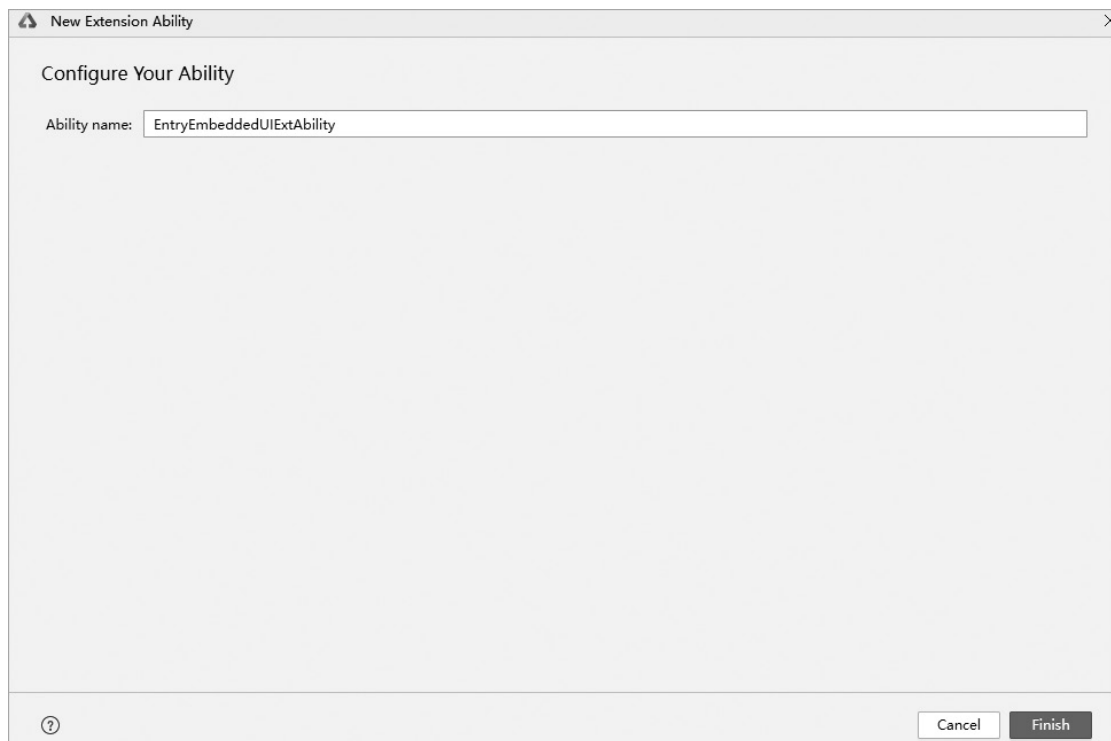


图 1-28 设置 Ability 名称

(1) 在Stage工程中选中ets文件夹下的pages，右击，选择New > Page，如图1-29所示，提供的Page类型如下。

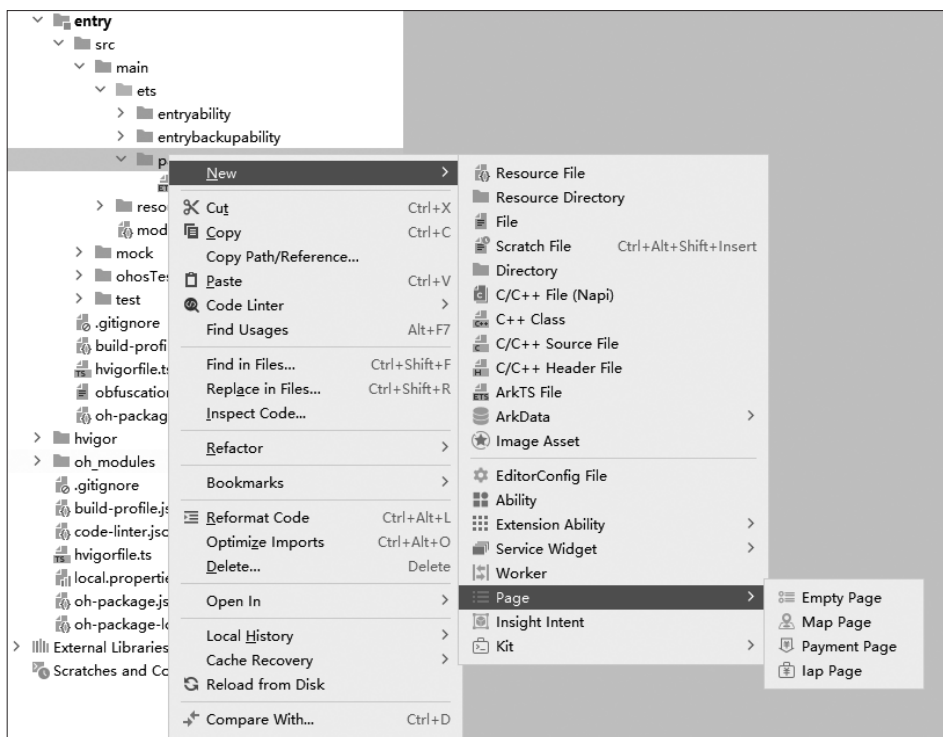


图 1-29 Page 类型

- Empty Page: 创建一个普通页面，展示基础的Hello World功能。
- Map Page: 创建一个地图页面，展示地图视图功能，当前仅支持在Phone设备中使用。
- Payment Page: 创建一个支付页面，可以实现点击按钮调起支付弹窗，当前仅支持在Phone设备中使用。
- Iap Page: IAP Kit场景化模板，可快速生成应用内购、虚拟数字商品支付相关代码。

说明：

API 10工程中仅支持创建Page，展示基础的Hello World功能；若要使用场景化Page模板，则需将工程切换为API 11及以上后进行开发。

(2) 输入Page name(由大小写字母、数字和下划线组成)，单击Finish按钮完成Page的添加，如图1-30所示。

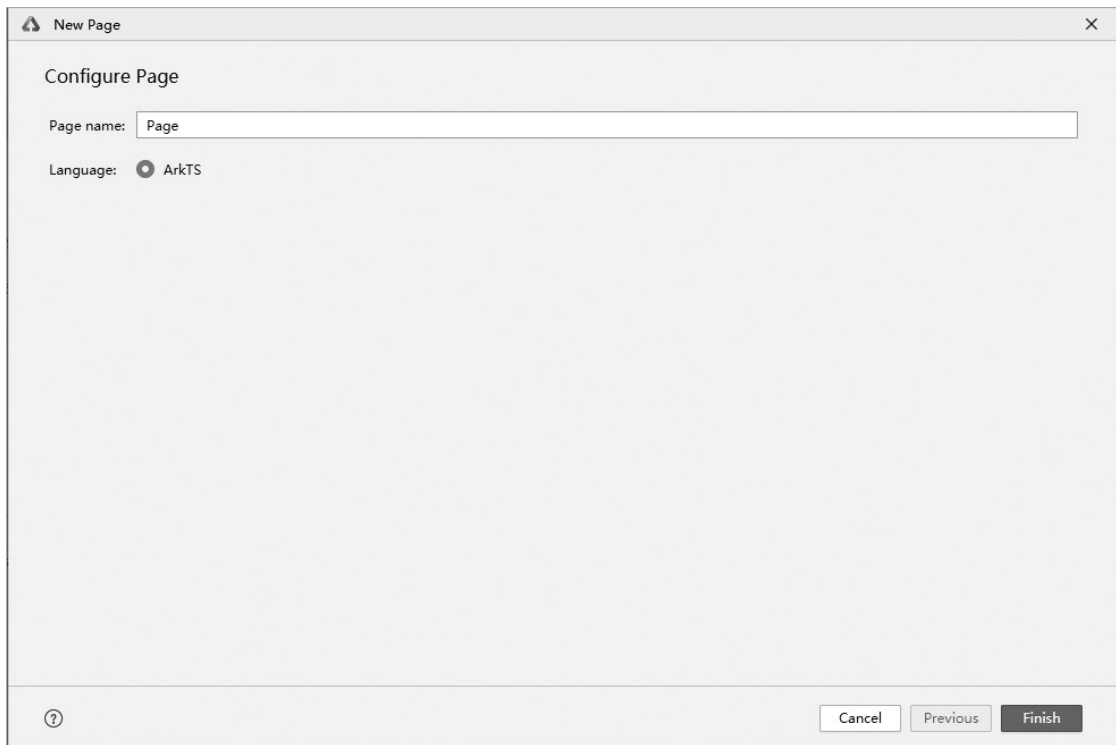


图 1-30 输入 Page name

1.3.6 构建第一个ArkTS应用(Stage模型)

1. 开发准备

通过构建一个简单的具有页面跳转/返回功能的应用(见图1-31)，快速了解工程目录的主要文件，熟悉HarmonyOS应用开发流程。

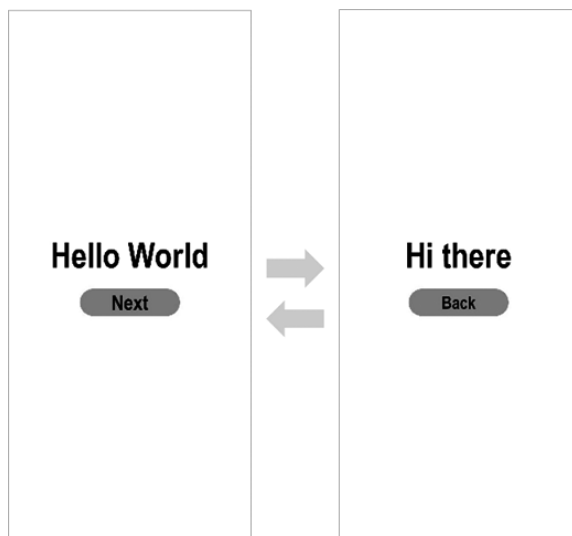


图 1-31 应用效果

在开始之前，需要了解以下基本内容：UI框架的简单说明、应用模型的基本概念。

1) UI框架

HarmonyOS提供了一套UI开发框架，即方舟开发框架(ArkUI框架)。方舟开发框架可为开发者提供应用UI开发所必需的能力，如多种组件、布局计算、动画效果、UI交互、图形绘制等。

方舟开发框架针对不同开发诉求和技术背景的开发者提供了两种开发范式，分别是基于ArkTS的声明式开发范式(简称“声明式开发范式”)和兼容JS的类Web开发范式(简称“类Web开发范式”)。表1-6是两种开发范式的简单对比。

表 1-6 两种开发范式对比

开发范式名称	语言生态	UI更新方式	适用场景	适用人群
声明式开发范式	ArkTS语言	数据驱动更新	复杂度高、团队合作要求较高的程序开发	移动系统应用开发人员、系统应用开发人员
类Web开发范式	JS语言	数据驱动更新	界面较为简单的脚本应用程序和卡片开发	Web前端开发人员

2) 应用模型

应用模型是HarmonyOS为开发者提供的应用程序所需能力的抽象提炼，它提供了应用程序必备的组件和运行机制。有了应用模型，开发者可以基于一套统一的模型进行应用开发，使应用开发更简单、高效。

随着系统的演进，HarmonyOS先后提供了FA模型和Stage模型。

2. 第一个ArkTS应用

1) 创建ArkTS工程

(1) 若首次打开DevEco Studio，则单击Create Project创建工程。如果已经打开了一个工程，则在菜单栏中单击File > New > Create Project，创建一个新工程，如图1-32所示。

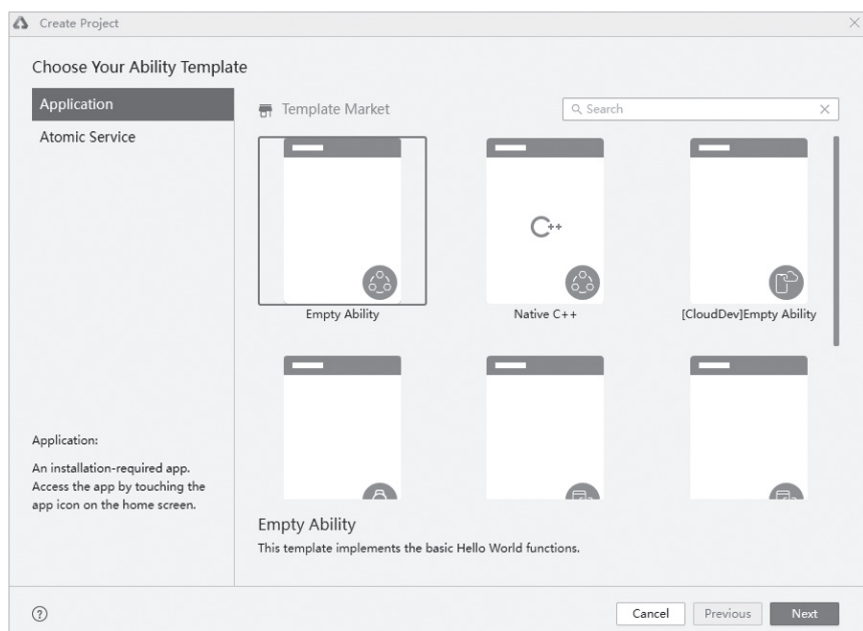


图 1-32 ArkTS 工程创建

(2) 选择Application应用开发(本文以应用开发为例, Atomic Service对应元服务开发), 选择模板Empty Ability, 单击Next按钮进行下一步配置。若开发者需要进行Native相关工程的发展, 则应选择Native C++模板。

(3) 进入配置工程界面, Compatible SDK表示兼容的最低API Version, 此处以选择5.0.4(16)为例, 其他参数保持默认设置即可, 如图1-33所示。

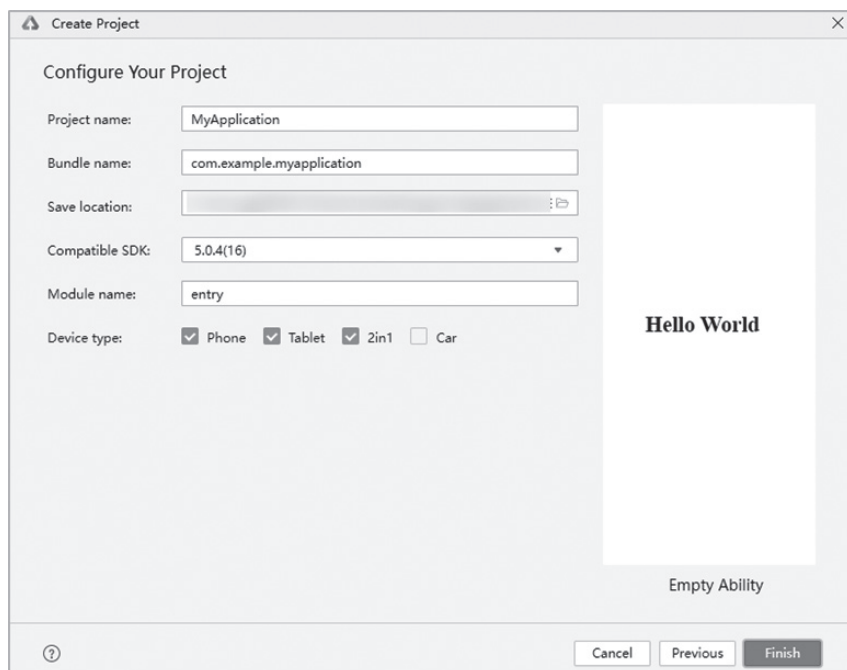


图 1-33 ArkTS 工程配置

(4) 单击Finish按钮, 工具会自动生成示例代码和相关资源, 等待工程创建完成。

2) 构建第一个页面

(1) 使用文本组件。

工程同步完成后，在Project窗口中单击entry > src > main > ets > pages，打开Index.ets文件，进行页面的编写。

针对本文中使用的文本/按钮来实现页面跳转/返回的应用场景，均使用Row和Column组件来组布局。对于更多复杂元素对齐的场景，可选择使用RelativeContainer组件进行布局。

Index.ets文件的示例如下。

```
// Index.ets
@Entry
@Component
struct Index {
  @State message: string = 'Hello World';
  build() {
    Row() {
      Column() {
        Text(this.message)
          .fontSize(50)
          .fontWeight(FontWeight.Bold)
      }
      .width('100%')
    }
    .height('100%')
  }
}
```

(2) 添加按钮。

在默认页面基础上添加一个Button组件，作为按钮响应用户点击，从而实现跳转到另一个页面。

Index.ets文件的示例如下。

```
// Index.ets
@Entry
@Component
struct Index {
  @State message: string = 'Hello World';
  build() {
    Row() {
      Column() {
        Text(this.message)
          .fontSize(50)
          .fontWeight(FontWeight.Bold)
        // 添加按钮，以响应用户点击
        Button() {
          Text('Next')
            .fontSize(30)
            .fontWeight(FontWeight.Bold)
        }
        .type(ButtonType.Capsule)
        .margin({
          top: 20
        })
        .backgroundColor('#0D9FFB')
        .width('40%')
        .height('5%')
      }
      .width('100%')
    }
  }
}
```

```

        .height('100%')
    }
}

```

(3) 预览效果。

在编辑窗口右上角的侧边工具栏中单击**Previewer**，打开预览器。第一个页面的效果如图1-34所示。

3) 构建第二个页面

(1) 创建第二个页面。

① 新建第二个页面文件。在Project窗口中单击entry > src > main > ets，右击pages文件夹，选择New > ArkTS File，命名为Second，按Enter键，可以看到文件目录结构如图1-35所示。

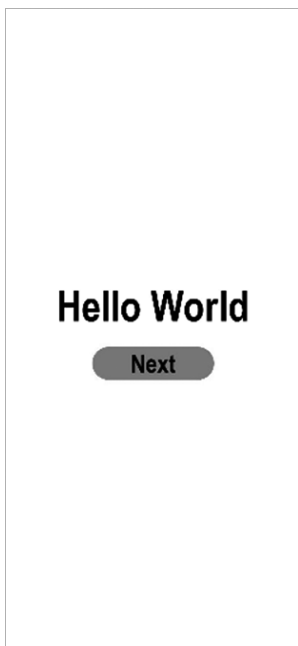


图 1-34 第一个页面的效果



图 1-35 文件目录结构

说明：

开发者也可以在右击pages文件夹时，选择New > Page > Empty Page，命名为Second，单击Finish按钮完成第二个页面的创建。若使用此种方式，则无须再进行下文中提到的第二个页面路由的手动配置。

② 配置第二个页面的路由。在Project窗口中单击entry > src > main > resources > base > profile，在main_pages.json文件中的“src”下配置第二个页面的路由“pages/Second”。示例如下。

```

{
  "src": [
    "pages/Index",
    "pages/Second"
  ]
}

```

(2) 添加文本及按钮。

参照第一个页面，在第二个页面添加Text组件、Button组件等，并设置其样式。Second.ets文件的示例如下。

```
// Second.ets
@Entry
@Component
struct Second {
    @State message: string = 'Hi there';
    build() {
        Row() {
            Column() {
                Text(this.message)
                    .fontSize(50)
                    .fontWeight(FontWeight.Bold)
                Button() {
                    Text('Back')
                        .fontSize(30)
                        .fontWeight(FontWeight.Bold)
                }
                .type(ButtonType.Capsule)
                .margin({
                    top: 20
                })
                .backgroundColor('#0D9FFB')
                .width('40%')
                .height('5%')
            }
            .width('100%')
        }
        .height('100%')
    }
}
```

4) 实现页面间的跳转

页面间的导航可以通过页面路由router来实现。页面路由router会根据页面url定位到目标页面，从而实现页面跳转。使用页面路由需导入router模块。

如果想要实现更好的转场动效，推荐使用Navigation组件。

(1) 从第一个页面跳转到第二个页面。

在第一个页面中，为跳转按钮绑定onClick事件，点击按钮时跳转到第二个页面。Index.ets文件的示例如下。

```
// Index.ets
// 导入页面路由模块
import { router } from '@kit.ArkUI';
import { BusinessError } from '@kit.BasicServicesKit';
@Entry
@Component
struct Index {
    @State message: string = 'Hello World';
    build() {
        Row() {
            Column() {
                Text(this.message)
                    .fontSize(50)
                    .fontWeight(FontWeight.Bold)
                // 添加按钮，以响应用户点击
                Button() {
                    Text('Next')
                        .fontSize(30)
                        .fontWeight(FontWeight.Bold)
                }
                .type(ButtonType.Capsule)
            }
        }
    }
}
```

```

        .margin({top: 20})
        .backgroundColor('#0D9FFB')
        .width('40%')
        .height('5%')
        // 为跳转按钮绑定onClick事件, 点击时跳转到第二个页面
        .onClick(() => {
            console.info(`Succeeded in clicking the 'Next' button.`)
            // 跳转到第二个页面
            router.pushUrl({ url: 'pages/Second' }).then(() => {
                console.info('Succeeded in jumping to the second page.')
            }).catch((err: BusinessError) => {
                console.error(`Failed to jump to the second page. Code is ${err.code},
                    message is ${err.message}`)
            })
        })
    }
    .width('100%')
}
.height('100%')
}
}

```

(2) 从第二个页面返回第一个页面。

在第二个页面中, 为返回按钮绑定onClick事件, 点击按钮时返回第一个页面。Second.ets文件的示例如下。

```

// Second.ets
// 导入页面路由模块
import { router } from '@kit.ArkUI';
import { BusinessError } from '@kit.BasicServicesKit';
@Entry
@Component
struct Second {
    @State message: string = 'Hi there';
    build() {
        Row() {
            Column() {
                Text(this.message)
                    .fontSize(50)
                    .fontWeight(FontWeight.Bold)
            }
            Button() {
                Text('Back')
                    .fontSize(30)
                    .fontWeight(FontWeight.Bold)
            }
        }
        .type(ButtonType.Capsule)
        .margin({top: 20})
        .backgroundColor('#0D9FFB')
        .width('40%')
        .height('5%')
        // 为返回按钮绑定onClick事件, 点击按钮时返回第一个页面
        .onClick(() => {
            console.info(`Succeeded in clicking the 'Back' button.`)
            try {
                // 返回第一个页面
                router.back()
                console.info('Succeeded in returning to the first page.')
            } catch (err) {
                let code = (err as BusinessError).code;
                let message = (err as BusinessError).message;
                console.error(`Failed to return to the first page. Code is ${code}, message
                    is ${message}`)
            }
        })
    }
}

```

```

    }
  })
}
.width('100%')
}
.height('100%')
}
}
}

```

(3) 预览效果。

打开Index.ets文件，单击预览器中的按钮进行刷新，效果如图1-31所示。

5) 使用真机运行应用

(1) 将搭载HarmonyOS系统的真机与计算机连接。

(2) 单击File > Project Structure... > Project > SigningConfigs，勾选Support HarmonyOS和Automatically generate signature复选框，单击界面提示的Sign In，使用华为账号登录。等待自动签名完成后，单击OK按钮即可，如图1-36所示。

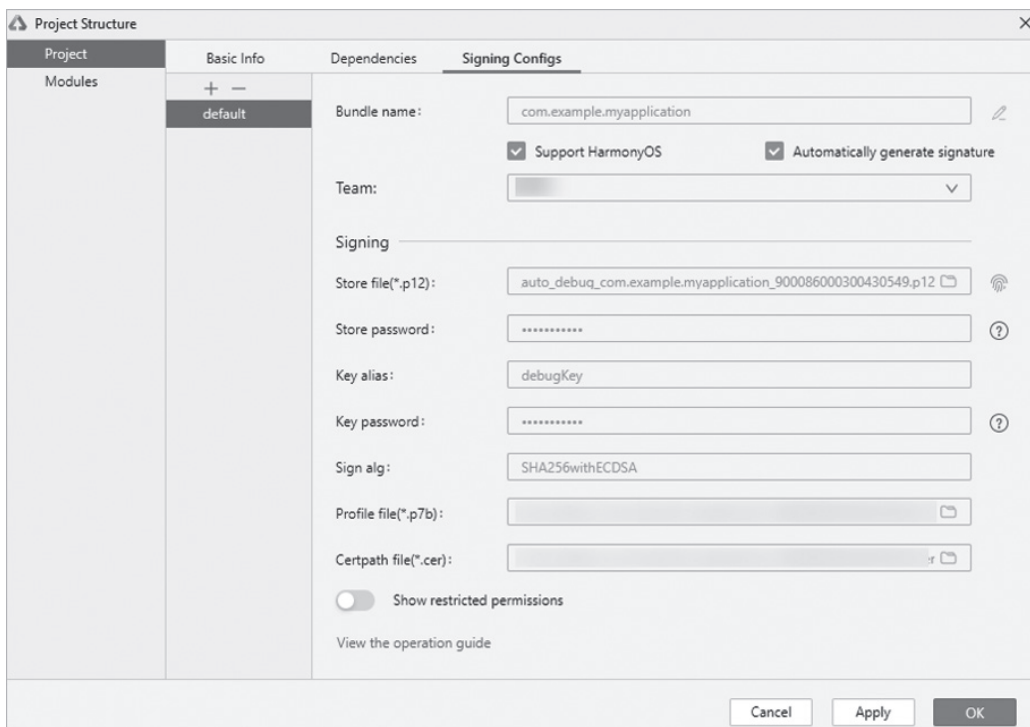


图 1-36 真机运行配置

(3) 单击编辑窗口右上角工具栏中的 ▶ 按钮运行应用，运行结果如图1-31所示。

习 题 1

- 下列选项中，不属于HarmonyOS的分布式架构优势的是()。
 - 跨设备协同能力强
 - 不同设备可作为整体协同工作
 - 能提升单个设备的运算速度
 - 便于开发者开发跨终端分布式应用

2. 以下关于HarmonyOS微内核设计的说法，错误的是()。
 - A. 仅保留操作系统最基础、关键的功能
 - B. 减少了内核的攻击面，提升了安全性与稳定性
 - C. 某个服务故障会导致整个系统崩溃
 - D. 模块化设计使系统便于定制化开发
3. DevEco Studio不支持()编程语言的应用开发。
 - A. ArkTS
 - B. Java
 - C. JS
 - D. C/C++
4. 请简述HarmonyOS的统一与弹性部署能力的含义。
5. 请简述在DevEco Studio中创建一个新的ArkTS工程的主要步骤。