

前言

本文档根据 fitness-2023 运动健康 app 的开发经验所写。

华为手表开发分为鸿蒙端和后端两个部分。鸿蒙端开发类似于安卓和 web 前端的开发，用来呈现界面和完成用户交互，所使用的框架是基于 vue 的，因此有相关开发经验的会很容易上手。后端用来支持服务，本项目后端基于 Nodejs 开发，轻便简洁。

鸿蒙端开发

一、开发环境

DevEco

专为华为应用开发的编译器，仿照 JetBtrain 设计，直接百度进官网下载最新版本即可。下载 DevEco 后，软件会继续要求下载相应的 SDK 和 API，建议全部下载，因为不同设备的开发所需要的 SDK 和 API 版本可能是不一样的。

Nodejs

DevEco 开发需要特定版本的 node 环境，一般低于 Nodejs 的官网现行版本。如果本机没有 node 环境，那么跟着提示一起下载即可；如果有，那么软件可能会因版本不一致问题提示你更换 node 环境。同时，node 环境也用于后端开发。

二、上手开发

<https://developer.huawei.com/consumer/cn/training/course/video/101612430951144217>

以上链接是手表开发入门视频，帮助了解基本情况。如果有网页开发或安卓开发的经验，那么就可以迅速上手。如果没有相应基础，即使把视频中的 demo 都实现一遍也很难继续深入。强烈建议先学习 JavaScript 或者 vue 的相关知识。以下是 JavaScript 教程：

<https://zh.javascript.info/>

<https://pan.baidu.com/s/1RD9FHSaEVfHhbrAjMdTwYA> 提取码: v4le

以上是一个实战视频，开发一个简单聊天助手 APP。本文档作者因为有网页开发的经验，因此没有看，新手可以照此学习，但要注意设备。手表前端开发，属于客户端开发的门类，对设备类型十分敏感。

三、调试和部署

调试

上文提到的 previewer 调试，是编译器自带的调试器，从 View->Tool Windows->Previewer 打开，方便快捷，用来快速查看界面效果和交互效果，但是局限性很多，例如键盘输入、网

络通信这些，只能通过真机调试。

虚拟机可以忽略不计，这东西太慢了。

然后就是真机调试，在官方文档中可以找到在真机上调试的文档，写的很乱，但应该是能看明白的。对于手表而言，USB 调试，本文亲测不太好用，因此一律使用 IP 调试。基本步骤就是，电脑和手表连到同一热点，手表打开开发者选项一系列操作，电脑连接手表的 IP，连上之后，打开 File->Project Structure->Signing Configs，自动申请临时证书，再 run，就能在真机上运行了。

部署

本地部署，就和真机调试是一个步骤。

上传到应用商城，需要在华为官网申请正式的证书将项目打包，上传验证，具体步骤在官方文档里也都有。由于本文档开发者不知道因为什么原因始终无法验证合法，因此放弃了上传到应用商城。

四、注意事项

文档

尽管华为官方开发文档，目前而言质量还有待提高，但是在开发过程中仍然需要时常翻阅。直接在下载 DevEco 的那个官网上就能找到文档。在查阅文档的过程中，你可能可以看到搜索内容对应的版本、权限、实现代码、实现效果等内容，但大多数是手机端开发的效果，对手表开发仍不太友好。

版本

版本是个大问题。华为官方给出的版本也时不时处在变化之中。例如上轮手表开发使用 DevEco 2，API 7 开发 `wearable`（华为 watch3 所属的设备类型）应用，但在本轮开发使用的 DevEco 3 中，`wearable` 仅由 API 6 支持。于是，本文档作者在 API 7 中错误地选择了 `lite wearable`（与 `wearable` 是不同的设备类型，顾名思义轻量级穿戴设备开发不支持很多接口），走了很多弯路。

除了 API 的版本，各种接口、组件的版本也很重要，这点在每篇文档的开头都会提及。例如，本轮开发使用 `fetch` 进行网络通信，但是在开发将近结束时，官方文档更新，API 6 以上不再支持 `fetch`，该用 `http`。目前，`fetch` 试用稳定，还没有完全被废弃的迹象，而因为 `http` 无法在 `previewer` 模式下测试，因此本文没有替换该接口。还有一些组件的方法或者属性，在官方文档中都会注明所需的 API 版本，在此不多赘述。

配置文件与权限：

一些接口是需要权限的，权限一般在官方文档里都有写，照着官方文档，在配置文件里权限部分添加即可。例如 `fetch` 接口，需要开启 `INTETNET` 权限，如果没有开启，在 `previewer` 模式下调试是不会发现问题的，因为编译器会通过电脑的互联网访问到后端，而忽略了真实的手表/手机上的情况。

除了权限，配置文件里还有 `app` 名称、路由等设置。

开发

开发遇到的问题基本源于对 JS/Vue 的生命周期、响应原理等等基础性的知识理解不深。

本文档作者理解也不是很深。就举一个例子说明，手表真机很容易熄屏，每次重新亮屏，对应的是代码中的 `onshow()` 周期，在这个周期内，必须关闭并重启类似 `setInterval` 之类的触发器，否则在实际应用中，屏幕一旦熄灭，触发器对应的功能就会戛然而止。而且，这类问题，也无法通过 `previewer` 调试被探知。如何解决该问题呢？本文尝试让手表不熄屏，但是查阅官方文档后，相应的接口不支持手表；于是只好修改触发器的逻辑，以匹配熄屏问题。

后端开发（基于 Nodejs）

一、开发环境

Nodejs

上文提到，此处不再赘述。

nodemon

一个非必要但实用的命令行工具，因为在一般情况下，使用 `npm` 启动 `nodejs` 服务，调试之后，需要重启，很麻烦。使用 `nodemon` 命令启动服务，则为热启动，代码变则服务变。但要注意，涉及数据库操作的时候，要分外小心，以免对数据库进行误操作。

Visual studio code

万能轻便开发工具。

MongoDB

非关系型数据库。

MongoDB Compass

MongoDB 官方提供的 GUI 工具，可以很方便的查看修改数据。本地开发调试以及真实场景中连接远程服务器中的数据库，都很实用。

二、上手开发

Nodejs+MongoDB 开发

Nodejs 写起来挺简单的，基本遵循 JavaScript 写法。与 MongoDB 的联动也不难，如果学习过数据库，那么掌握基本的数据库连接和增删改查操作，那么也就足够完成开发任务了。以下链接帮助学习 Nodejs 与 MongoDB 的基础开发：

https://www.bilibili.com/video/BV1Gz4y1X7Uq?spm_id_from=333.999.0.0&vd_source=e3ef940eb6c895ae8e05433a9ad26c2a

其中，很多内容都不需要看，比如 redis 这些，只需要聚焦于 MongodDB 的增删改查操作即可。另外，电脑上本地下载的 MongoDB 数据库仅仅是方便本地调试，实际部署的时候，要在真正的服务器上配置好 MongoDB 环境。

Koa2 框架

实际上，Nodejs 开发框架可以选择 express 或者 koa2，本文依照上轮开发路径，选择了 koa2 框架。以下是学习链接：

<https://www.jspang.com/article/61>

https://www.bilibili.com/video/BV1JE411q7qr?spm_id_from=333.999.0.0&vd_source=e3ef940eb6c895ae8e05433a9ad26c2a

这类实战类项目的学习视频，可以不用全部看完，只需要看前面基础的部分即可，后面涉及具体特定任务的实现可以选择性学习。如果想要使用 express 框架，B 站直接搜索即可。

三、部署服务

部署服务有两种方式。

方式一：部署在云服务器上

1. 准备一台云服务器，本文使用的是阿里云服务器。

2. 参考链接：

https://www.bilibili.com/video/BV1my4y1a7xN?p=1&vd_source=5e2172352eac9cefe238dc46bc81e6bc

<https://www.yuque.com/uuyv8m/linux/im67s65a49k3yk1m>

方式二：部署在开发者自己的电脑上

开发使用的电脑也可以作为服务器，只需要使用内网穿透。有免费的，也有付费的。免费的 frp 配置方法，百度即可，略微复杂。付费的，某些网站会提供该服务，通过分配本地端口一个公网域名，实现内网穿透，价格比租服务器便宜。

但是内网穿透要求开发者的电脑实时待机，实际应用时还是有风险的。

四、注意事项

在部署云服务器时，要注意服务器的 centos 版本，如果版本过高，可能导致游戏额动态链接库的缺失或者不匹配，因此要尽量按教程的版本来配置。

MongoDB 在 6.0 以上版本不支持 mongo.exe，而是有另外的 shell 工具，建议在 MongoDB 官网仔细查看用法。

配置 MongoDB.conf 时，一定要配置 bind_ip: 0.0.0.0，否则外部很可能访问不了服务器上的数据库。