

如何分包加载？分包加载的优势在哪？

- 分包的介绍

分包加载其实就是把小程序划分成不同的子包，然后在用户使用的时候才进行加载，就是我们口中常说的按需加载。每个分包一般会有一个主包，主包会在小程序启动时首先下载，其他分包则会在进入对应页面时下载对应包的资源。

目前小程序分包大小有以下限制：

- 整个小程序所有分包大小不超过 8M
- 单个分包/主包大小不能超过 2M

- 分包的使用

通过在app.json中设置subpackages字段声明项目的分包结构，这里我们看下官网的例子：

```
{
  "pages": [
    "pages/index",
    "pages/logs"
  ],
  "subpackages": [
    {
      "root": "packageA",
      "pages": [
        "pages/cat",
        "pages/dog"
      ]
    }, {
      "root": "packageB",
      "name": "pack2",
      "pages": [
        "pages/apple",
        "pages/banana"
      ]
    }
  ]
}
```

可以看到这里除了主包里的pages页面外，subpackages中也配置了几个包，里面也有对应的页面。subpackages具体配置可以看看下图：

字段	类型	说明
root	String	分包根目录
name	String	分包别名，分包预下载时可以使用
pages	StringArray	分包页面路径，相对与分包根目录
independent	Boolean	分包是否是独立分包

这里root根目录指的是指定根目录的文件夹，来看下目录结构就知道了

```
├─ app.js
├─ app.json
├─ app.wxss
├─ packageA
│   └─ pages
│       ├── cat
│       └─ dog
├─ packageB
│   └─ pages
│       ├── apple
│       └─ banana
├─ pages
│   ├── index
│   └─ logs
└─ utils
```

上面指定的root中packageA和packageB就是packageA和packageB的路径

这里使用时需要注意几点，如下：

- subpackages配置路径外的目录会被打包到主包（app）中
- 主包（app）可以有自己的pages（即subpackages外的 pages 字段）
- subpackage 的根目录不能是另外一个 subpackages 内的子目录
- tabBar 页面必须在主包（app）内
- 分包与分包之间不可以引用其js文件、template以及资源（如packageA无法require packageB JS 文件），但可以require app`、自己 package 内的 js文件、template以及资源

- 分包的优势

采用分包加载，对开发者而言，代码体积可以承载更大；而对用户而言，可以提高小程序的启动速度，大大提升体验。