

JS高级 -- Hybrid

1. hybrid 是什么，为何会用 hybrid?

- 文字解释
 - hybrid 即‘混合’，即前端和客户端混合开发
 - 需前端开发人员和客户端开发人员配合完成
 - 某些环节也可能涉及到 server 端
- 存在价值，为何会用 hybrid
 - 可快速迭代更新（无需APP审核），hybrid是纯前端代码，没有很高的操作权限
 - 体验流畅（和native的体验基本类似）
 - 减少开发和沟通成本，双端公用一套代码
- webview
 - 是app中的一个组件（app可以有webview,也可以没有）
 - 用于加载h5页面，即一个小型的浏览器内核
- file:// 协议
 - file协议 file:// 后面是文件的绝对路径
 - file协议：本地文件，快
 - http(s)协议：网络加载，慢
- hybrid 实现流程
 - 不是所有场景都适合使用hybrid
 - 使用native：体验要求极致，变化不频繁(如头条的首页)
 - 使用hybrid：体验要求高，变化频繁（如头条的新闻详情页）
 - 使用h5：体验无要求，不常用（如举报、反馈等页面）
 - 具体实现
 - 前端做好静态页面，将文件交给客户端
 - 客户端拿到前端静态页面，以文件形式存储在app中
 - 客户端在一个webview中使用file协议加载静态页面

2. 介绍一下 hybrid 更新和上线的流程?

- 分版本号，设置版本号
- 将静态文件压缩成zip包，上传到服务端
- 客户端每次启动，都去服务端检查版本号
- 如果服务端版本号比客户端版本号更新一些，就去下载新的zip包
- 下载之后解压包，然后将现有文件覆盖

3. hybrid 和 h5 的区别?

- 优点
 - 体验更好，跟native体验基本一致
 - 可快速迭代，无需app审核
- 缺点
 - 开发成本高，联调、测试、查bug都比较麻烦
 - 运维成本高
- 适用场景
 - hybrid:产品的稳定功能，体验要求高，迭代频繁,适合产品型页面

- h5:单次的运营活动（如xx红包）或不常用功能，适合运营型页面

4. 前端 JS 和客户端如何通讯？

- schema 协议
 - 前端和客户端通讯的标准
- 内置上线
 - 将封装的代码打包，内置到客户端，叫做 `invoke.js`
 - 客户端每次启动webview,都默认执行 `invoke.js`
 - 本地加载，免去网络加载的时间，更快
 - 本地加载，没有网络请求，黑客看不到 schema 协议，更安全