

HttpDNS SDK 功能设计规范

1. 核心解析功能 (Core Resolution)

SDK 的基本职责是将域名转化为 IP 列表。

- **同步解析 (Sync Resolve)**: 直接发起网络请求并等待结果, 适用于对实时性要求极高且允许阻塞的情况(通常不建议在 UI 线程使用)。
- **异步解析 (Async Resolve)**: 非阻塞请求, 通过回调或 Future/Promise 返回结果。
- **批量解析 (Batch Resolve)**: 一次性请求多个域名的 IP, 减少 HTTP 握手开销(如在 App 启动阶段)。
- **多记录类型支持**: 除了 A 记录(IPv4), 还应支持 AAAA(IPv6) 以及自动根据当前网络环境返回 A 或 AAAA。

2. 缓存管理模块 (Intelligent Caching)

缓存是降低延迟的关键, SDK 必须维护一套内存级及持久化级的缓存系统。

- **TTL 自动过期**: 遵循服务端返回的 TTL(生存时间)。当 TTL 过期时, 自动清理或触发异步更新。
- **持久化存储**: 将解析结果加密保存到本地(如 SQLite 或 SharedPreferences), 以便 App 重启后能立即使用历史解析结果(“冷启动优化”)。
- **软过期策略**: 当缓存过期时, 优先返回过期数据给业务层, 同时 SDK 后台发起异步请求更新, 确保业务访问不间断。
- **LRU 淘汰**: 设置缓存上限, 根据最近最少使用原则清理陈旧数据。

3. 网络感知与自适应 (Network Awareness)

移动网络环境复杂, SDK 需要根据环境变化动态调整。

- **网络切换监听 (Network Change)**: 当设备从 WiFi 切换到 4G/5G, 或更换了 WiFi 接入点时, 必须立即清空缓存并重新解析。因为不同网络环境下, 最优接入点(IP)通常不同。
- **IP 优选 (IP Ranking/Racing)**: 如果一个域名对应多个 IP, SDK 可以在后台发起轻量级的探测(如 TCP Connect 耗时测试), 将响应最快的 IP 排在前面。
- **IPv6 优先策略**: 探测当前网络是否支持 IPv6, 若支持则优先下发 AAAA 记录。

4. 容错与降级策略 (Failover & Degradation)

确保在极端情况下业务不中断。

- **多服务 IP 轮询**: SDK 内置多个 HttpDNS 服务节点地址 (Anycast IP 或备份 IP)。若主节点连接失败, 自动重试备份节点。
- **自动降级 (Degradate to System DNS)**:
 - 当 HttpDNS 请求超时。
 - 当 HttpDNS 服务端返回 5xx 错误。
 - 当连续多次 HTTPS 握手失败。

SDK 应立即切换到系统原生解析逻辑 (InetAddress.getByName)。

- **域名过滤 (Blacklist)**: 支持设置黑名单, 特定域名 (如私有云、本地局域网域名) 强制走系统 DNS。

5. 安全防护 (Security)

- **请求签名 (API Authentication)**: 基于 Secret 算法生成 MD5 或 HMAC 签名, 防止 API 被盗用。
- **HTTPS 传输**: 支持 HTTPS 访问 HttpDNS 服务器, 解决 SNI 握手及 Host 校验问题。
- **数据加密**: 本地持久化存储的解析记录应加密, 防止设备被 root 后数据泄露。

6. 业务辅助工具 (Business Tools)

- **预解析 (Pre-fetch)**: 支持在 App 启动时, 由开发者配置一组核心域名进行静默解析, 避免用户第一次访问时的等待。
- **SNI 助手 (SNI Helper)**: 提供工具方法帮助开发者在 IP 直连模式下正确填充 TLS 握手的 server_name。
- **自定义 Header 注入**: 自动为网络请求注入正确的 Host Header。

7. 监控与日志 (Monitoring)

- **解析统计**: 记录各域名的解析成功率、耗时、缓存命中率等数据, 定期上报。
- **错误分析**: 详细记录失败原因 (超时、签名错误、网络不可用等), 方便后端调优。