

# 美团开源Logan Web：前端日志在Web端的实现

原创 孙懿 美团技术团队 昨天

点击关注“美团技术团队”，阅读更多技术干货



总第378篇

2019年 第56篇



2018 年，美团点评推出大前端日志系统—— Logan，并开源了 Android 与 iOS 端的 SDK。这次，我们又开源了在 Web 环境运行的 SDK、日志分析平台以及服务端代码，希望能够为社区内小伙伴们提供更加完整的大前端日志解决方案。

## 1. 前言

Logan 是美团点评推出的大前端日志系统，支持多端环境运行，可为客户端、Web、小程序等用户端环境提供前端日志的存储、收集、上报及分析能力，能够帮助开发人员快速定位并解决端上问题，便于及时响应用户反馈与排除异常。

- 2018 年 10 月，Logan 在社区开源了 Android 与 iOS 端的 SDK，实现了在客户端进行日志存储及上报代码的功能，引起用户端相关开发者的广泛关注。详细可参见博客文章：《Logan：美团点评的开源移动端基础日志库》。
- 2019 年 12 月 12 日，Logan 开源了在 Web 环境运行的 SDK、日志分析平台以及服务端代码，为开发者们提供了 Logan 大前端日志系统的一整套实现方案，进一步解决了多端环境中日志的存储与采集问题。

本文将围绕 Logan 在 Web 端的应用背景、技术实现、美团点评的实践、开源整体进展以及未来规划这几个方面展开介绍，以方便读者对 Logan 大前端日志系统有更加深入的了解。

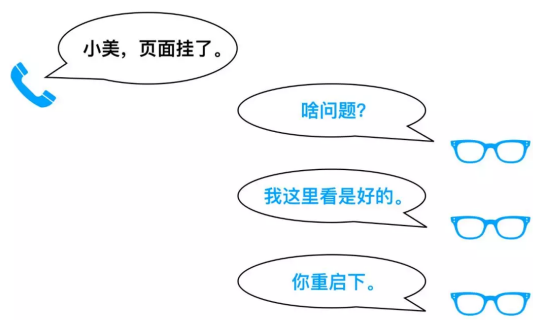
Logan 项目地址：

<https://github.com/Meituan-Dianping/Logan>

## 2. 背景

### 2.1 为何需要 Logan?

在 Logan 诞生前，用户端开发者在面对用户反馈页面功能异常时，最常说的灵魂三答是：



这三条回答分别对应着开发者在解决端上问题时的心路历程：

- “啥问题”：通过与用户沟通，获取异常发生前后过程的详细描述，尝试在开发者本地模拟，以期复现问题。
- “我这里看是好的”：问题没有复现，应该是用户端兼容性的 Bug。
- “你重启下”：想不出修复的办法，只能“死马当活马医”，碰碰运气。

对用户端的开发者来说，本地无法复现的问题好比“断了线的风筝”，让人无计可施。如果有办法获取到事发时完整的日志流以及用户环境的上下文信息，就能够帮助开发者快速了解并还原问题的发生现场，可以更有效地定位排查问题，让风筝最终被拉回到开发者手里。正是因为这样的迫切需要，Logan 大前端日志系统才应运而生。

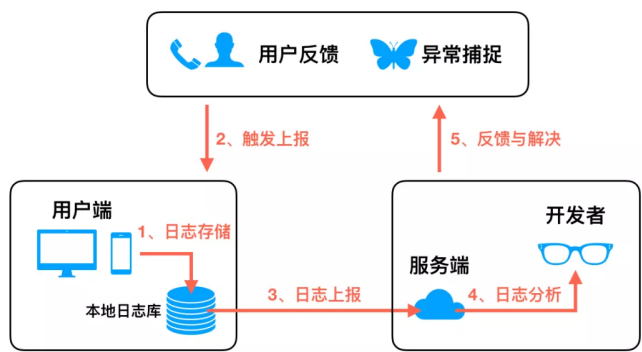
## 2.2 Logan 日志系统的策略与核心

### 2.2.1 Logan 在各端实行的通用策略

虽然用户端上完整的日志流及上下文环境信息更有助于开发者定位到问题，但对每个用户端的日志流都进行实时上报的话，也会出现如下问题：

- 巨大开销：耗用户流量，占用企业带宽与服务器存储资源。
- 大海捞针：在海量日志中可能只有极少部分的日志能够帮助复现问题。

因此 Logan 在各端上实行的通用策略，是本地日志存储结合触发上报的模式，如流程图所示：



- 平时在用户端脚本执行过程中产生的日志会落地到本地的存储容器中。
- 当遇到用户反馈或者端上异常被捕获时，Logan 以特定机制触发本地日志的上报。

- c. 本地日志流将在用户端上传，由服务端收集并解析，最后上传至云端存储。
- d. 由 Logan 统一的日志分析平台向开发者提供日志数据的可视化展示。
- e. 开发者利用 Logan 日志排查定位并解决问题后，向用户反馈或者排除异常。

### 2.2.2 Logan 的三大核心

上文所阐述的 Logan 通用策略中的工作流程也决定了 Logan 日志系统拥有三大核心：

- 用户端 SDK（客户端版、Web 版及小程序版）：负责存储与上报端上日志。
- 服务器端：负责接收、解析、整合与分析日志。
- 日志分析平台：提供日志的查询与数据可视化展示。

## 2.3 Logan 在 Web 端面临的问题

在 Web 环境中若要实现端上日志存储及上报需要解决三大难点：

- 如何存储？需要解决 Web 本地大体积日志流的存取。
- 如何保障日志安全？在本地已存储的日志需要有数据安全保障。
- 如何上报？需要有效的机制触发日志的上报。

## 2.4 Logan Web 做了什么？

Logan Web 是 Logan 在 Web 端的存储及上报实现方案，利用现有的前端技术加以优化与整合，有效地解决了 Web 端面临的三大难点。我们将存储与上报的实现封装在 Web 端的 SDK 内，开发者只需在页面脚本中引入该 SDK，便可直接使用 Logan 在 Web 端上的日志安全存储与上报能力。

下面将重点围绕存储方案、数据安全及上报机制这三点，具体阐述 Logan Web 目前的技术实现。

## 3. 技术实现

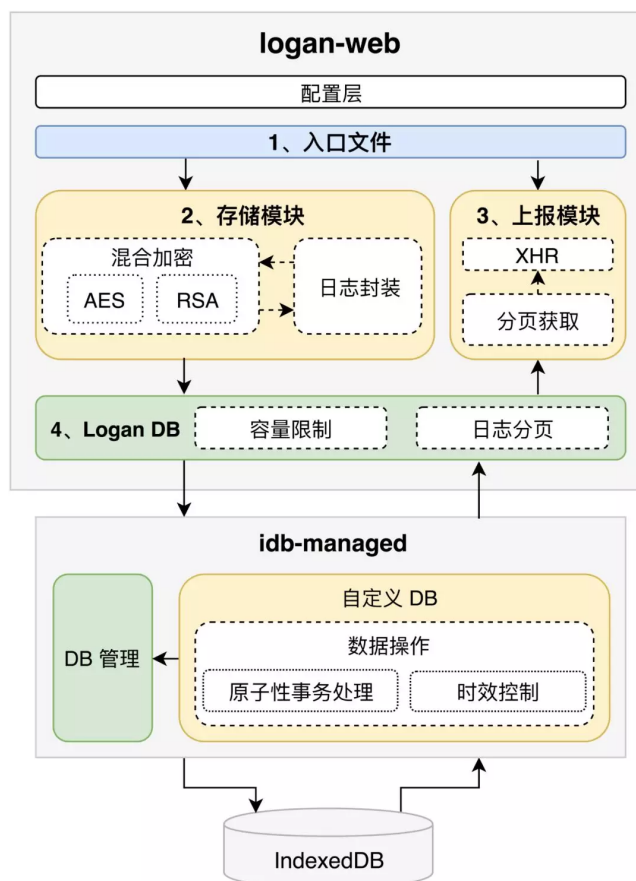
Logan Web 在底层利用了现有前端技术来实现大体积日志的安全存储：

- 存储方面：利用浏览器的 IndexedDB 作为本地日志的大容量存储容器。
- 日志安全方面：使用混合加密模式确保本地已存的隐私日志数据不会被破解。

Logan 为 Web 开发者提供了 logan-web 这个 SDK 包，其内部主要分为存储与上报两大核心模块，底层依赖了 IndexedDB 存储与加密组件。开发者可在页面脚本中引入并加载该 SDK 来调用日志存储或上报接口。

### 3.1 Logan Web 整体技术架构

以下是 Logan Web 的整体架构示意图：



- logan-web 提供了一个入口文件，它将在日志存储方法或者日志上报方法被触发时，异步地获取存储或上报模块。
- 存储模块中会优先处理日志内容的加密及包装，再执行后续的分页存储流程。
- 上报模块会分页读取指定天的日志数据，并行上报至接收日志的服务端，进行后续的日志解析、解密、整合及分析。
- 这两大核心模块都会使用 Logan DB 模块封装的日志存取逻辑，该模块还会控制本地日志数据的存储容量以及日志分页。

对 IndexedDB API 的调用被封装在独立于 logan-web 的 idb-managed 包中，该包主要解决在使用 IndexedDB 进行本地存储时遇到的技术挑战。

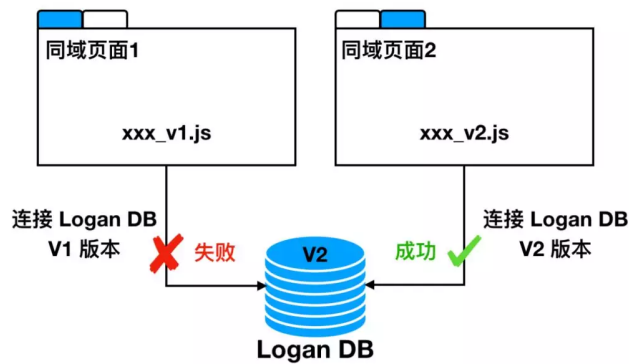
## 3.2 本地存储方案：idb-managed

### 3.2.1 原生 IndexedDB API 的局限

IndexedDB 支持大容量存储，并且其读取操作是异步化的，非常适合作为 Logan Web 的本地存储容器。但 IndexedDB API 在使用上会遇到如下几个问题：

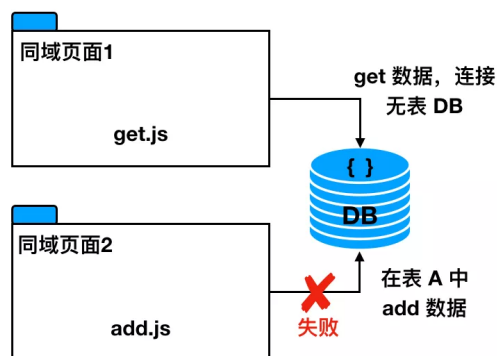
#### (1) DB 版本升级问题

本地 DB 依靠版本的升级来更新库表结构，当本地该 DB 的版本升级后，尝试连接低版本 DB 的操作将失败。



## (2) 获取 DB 数据前需要设置 DB 版本及库表信息

获取 DB 数据前需要首先连接 DB，如果连接时没有设置恰当的库表信息，下一次存储时依然以同版本建立连接，IndexedDB 则不再更新该 DB 的库表结构，最终会因为存储数据不匹配表结构而导致存储失败。



## (3) IndexedDB 不提供数据的时效设置与过期数据清理

IndexedDB 默认数据是持久化落地的，尽管它的可用容量远大于 LocalStorage，但在像 Logan Web 这样需要随时间不断更新本地数据的使用场景下，就需要一套随时间迭代的数据更新机制来“除旧存新”。

## (4) 原生 IndexedDB API 不提供多表间的原子性增删操作

原生的 IndexedDB API 只提供了单条数据的添加，以及单表内的数据批量删除操作，并不直接提供 API 对多表的数据进行添加或者删除的原子性操作（要么全部生效，要么全不生效）。Logan DB 的库表结构涉及到两张表数据间的联动，需要多表间原子性增删来保证日志数据在两张表间保持一致。

### 3.2.2 idb-managed 的解决手段

为了统一解决以上在使用 IndexedDB 时面临的困扰，我们额外封装了 idb-managed 包，该包随着 Logan Web 一起开源。

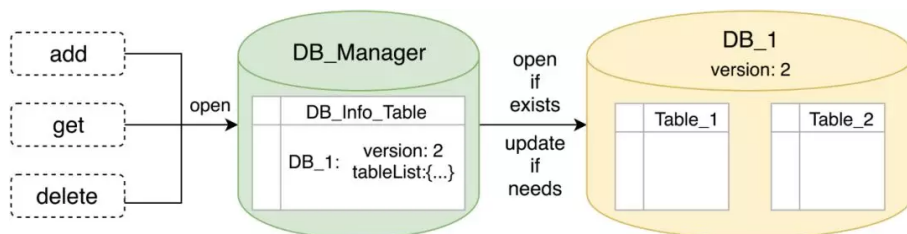
idb-managed 中分别针对：

- 问题 1 与 2：提出并实现了 DB Manager 机制来解决 DB 版本升级与数据获取的困境。

- 问题 3：封装了对存储数据的时效设置、过期处理逻辑。
- 问题 4：利用事务回滚方法实现在多表内的原子性增删操作。

### (1) DB Manager 机制

idb-managed 中内建了一个 DB Manager 来管理当前所有本域下的 DB 版本与库表结构信息，其实 DB Manager 自身就是一个版本号固化的 DB，它本身不存在升级问题。建立连接示意图如下：



当对一个新 DB 建立连接时，会将脚本中设置的 DB 版本及库表信息注册到 DB Manager 中并把数据存储下来。下一次如果有该 DB 的低版本尝试连接时，DB Manager 会用当前已注册的库表信息连接并打开 DB，由此避免了 DB 升级而导致连接低版本失败的问题。

同时，因为有 DB Manager 来统一管理本地的 DB 信息，所以从 DB 获取数据可以无需知晓 DB 库表结构，只需要 DB 名和表名即可。当本地库表不存在时，DB Manager 会阻止对该 DB 的连接，直接返回空数据，避免了错误的库表结构污染本地 DB。

### (2) 数据时效控制

idb-managed 会为每张表建立一个到期时间索引，开发者可对单条数据、单个表或者单个库设置一个持久化时间限制，在数据存储时 idb-managed 会根据这些限制及优先级顺序（单条 > 单表 > 单库）计算该条数据的到期时间，并与数据一起保存下来，过期的数据会在该表下一次添加数据时先行删除。

### (3) 多表间原子性增删操作

IndexedDB 中数据的增删操作全部建立在事务（IDBTransaction）基础之上，idb-managed 封装了多表批量数据的增删操作接口，并将这些操作包裹在一次事务内。如果在一次事务中发生异常，idb-managed 将执行本次事务的回滚，从而保证这批操作具有原子性。

## 3.3 日志加密方案：混合加密

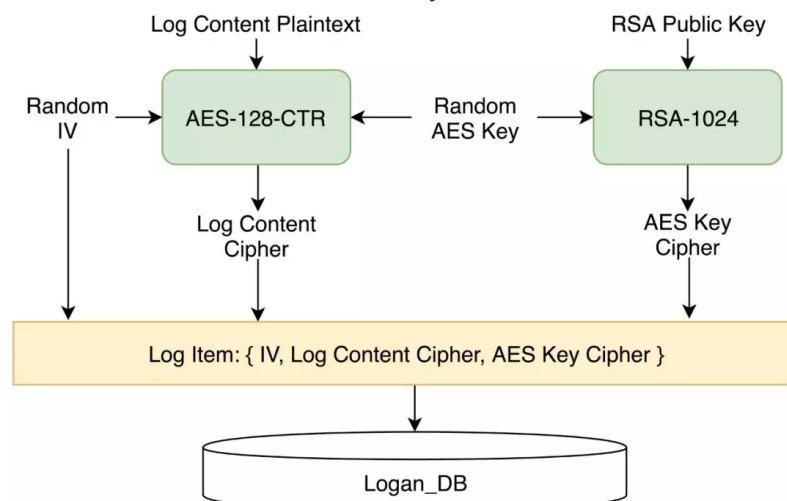
混合加密方式吸取了对称加密与非对称加密各自的优势：

- 对称加密：保证对长内容加密的效率。
- 非对称加密：保证对称密钥的安全性。



Logan Web 选择了 AES-128-CTR 结合 RSA-1024 的混合加密模式。在存储每条具有私密性的日志前都会经历以下加密流程：

- 准备对称密钥与初始向量：随机产生 AES 对称密钥 AES Key 及初始向量 IV。
- 对称加密：使用 AES Key 及 IV 对日志明文进行 AES-CTR 对称加密，得到日志密文。
- 非对称加密 AES Key：使用 RSA 公钥对 AES Key 进行非对称加密，得到 AES Key 密文。该 RSA 公钥与服务器端的私钥是成套的，只有该私钥可以解开该 AES Key 密文，从而解开日志密文。
- 包装数据：将以上初始向量、日志密文与 AES Key 密文包装成一条日志对象，随后存储落地。



### 3.4 上报的触发机制

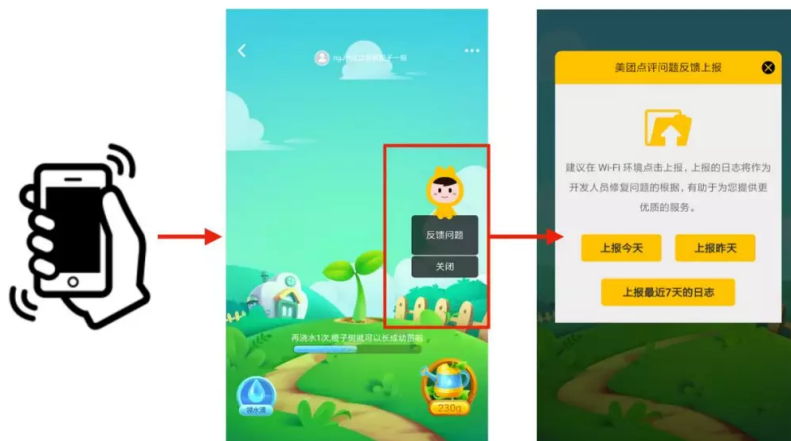
用户端的日志上报触发机制一般分为两大类：

- 用户主动触发。优点是上报的日志能够对应到用户反馈的个案；缺点是存在交互上的用户教育成本，同时依赖用户反馈的异常处理流程，过于滞后。
- 代码层面触发。优点是用户无需感知上报流程；缺点是可能存在大量“无帮助”的上报日志，需要对触发条件做好频率控制。

Logan Web 在两类方式上提供了配置与接口，供 SDK 使用方自行选择与扩展，例如：

#### (1) 用户主动触发

- PC 端：Logan Web 在美团点评内实践时，为业务方提供了 DOM 元素配置项，用于 Logan Web 绑定触发上报的钩子函数。
- H5 手机端：Logan Web 可扩展内置设备摇动检测，利用浏览器的 DeviceMotionEvent 事件监听，当用户“摇一摇”时，引导上报流程的美团卡通形象会出现在页面侧栏，如下图所示。
- 通用：Logan Web 在美团点评内扩展提供了接口供业务方显示或者隐藏引导上报流程的侧栏。



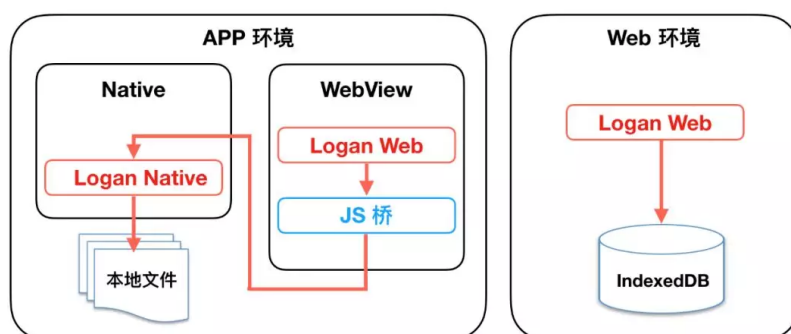
## (2) 代码层面触发

考虑到用户主动触发方式存在的弊端，Logan Web 提供了上报接口供使用者在代码层面调用。另外 Logan Web 也正在迭代实现内部的代码触发上报逻辑，提供在 Web 端内异常发生等特定场景下主动上报日志的能力。

## 4. Logan Web 在美团点评内的实践

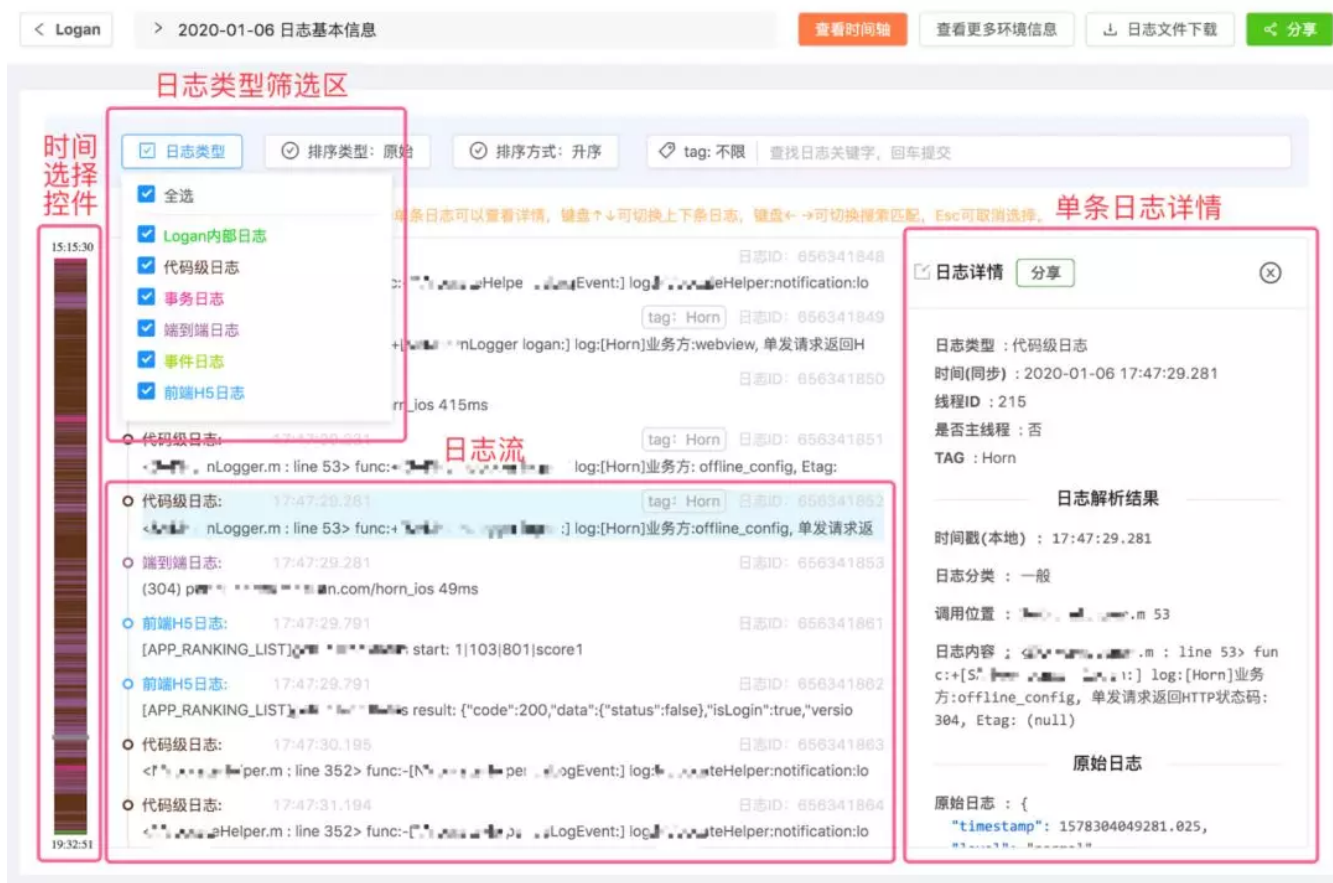
Logan Web 已在美团内部上线，并持续迭代了一年多的时间，覆盖公司很多主要的业务线。截止 2019 年 11 月，公司内使用 Logan Web 的页面 PV 量已达日均 1.5 亿余次，Web 上报日志量达日均 500 余次，约 97% 上报的 Web 日志在 Logan 日志分析平台上被公司研发同学搜索查阅。利用 Logan 日志系统，公司各研发团队已能够尽早获得完整的客户端日志流及环境信息，帮助业务及时排查问题并响应用户反馈。

另外，在美团，Logan Web 除了提供上文介绍的技术实现之外，还利用美团内部的通用 JS 桥组件实现了与 Logan 客户端日志的打通，如下图流程所示：



这意味着在美团现有的 App 环境（如美团 App、点评 App 等）中运行的 H5 页面如果使用了 Logan Web，所记录的日志会利用 JS 桥传给 Logan 客户端，与客户端日志一起落地在 App 本地文件中。因此在美团的 App 环境内上报的日志流中可查看上下文连续的 Web 端日志与客户端日志，日志分析平台展示的某篇日志详情示意图如下：





## 5. Logan 开源进展与未来规划

在 Logan Web 开源前，我们在美团 Logan 开源技术交流群中进行了开发者需求调研，依据收集到的建议，Logan Web 这次开源版本将支持 TypeScript，同时提供了本地日志的加密策略选择。开源的代码及使用文档可在 Meituan-Dianping/Logan/WebSDK 仓库下查阅，开发者也可以在项目中直接通过 npm 包引入的方式引入 logan-web。同时 Logan Web 底层依赖的 idb-managed 也已在 GitHub 与 npm 仓库开源。

随着本次 Logan Web 同时开源的还有 Logan 服务端与 Logan 日志分析平台的实现，读者可一并在 Logan 代码仓库下找到相应的代码和使用文档。客户端 SDK 的实现博客可点击参考：《美团点评移动端基础日志库——Logan》。

目前 Logan 已开源了客户端 SDK、Web 端 SDK、服务端及日志分析平台，已经为社区开发者们提供了初步完善的整套 Logan 日志系统实现。在未来我们将继续优化扩展 Logan 能力，帮助开发者们在各端环境中，都能更快更早更方便地定位问题及排除异常。

模块	已开源	规划中
iOS 端 SDK	✓ (2018年10月)	
Android 端 SDK	✓ (2018年10月)	
Web 端 SDK	✓ (2019年12月)	
服务端	✓ (2019年12月)	
日志分析平台	✓ (2019年12月)	
动态化框架适配		✓ (2020年Q1)
小程序 SDK		✓ (2020年Q3)

## 6. 联系我们

本次开源只是 Logan 贡献社区的一小步，我们希望在未来 Logan 能够为社区提供更完整可靠的大前端日志服务，我们诚挚地欢迎开发者向我们提出宝贵的建议，与我们共建社区。您可以挑选以下方式向我们反馈建议和问题：

- 在 [github.com/Meituan-Dianping/Logan](https://github.com/Meituan-Dianping/Logan) 提交 PR 或者 Issue。
- 发送邮件至：  
edp.bfe.opensource@meituan.com。

## 7. 作者简介

孙懿，美团点评基础技术部前端技术中心资深工程师。

欢迎加入**美团 Logan 技术交流群**，跟作者零距离交流。如想进群，请加美美同学的微信（微信号：**MTDPtech03**），回复：**Logan**，美美会自动拉你进群。

----- END -----

## 招聘信息

美团点评基础技术部前端技术中心负责美团云平台运维领域的前端基础研发工作，包含前端监控、日志系统、长连通信、运维工具等基础建设。欢迎各路英雄扫码投递简历，我们满心期待您的加入。感兴趣的同学可将简历投递至：[tech@meituan.com](mailto:tech@meituan.com)（邮件标题注明：基础技术部前端技术中心）。

## 也许你还想看

Logan：美团点评的开源移动端基础日志库

Logan：美团点评移动端基础日志库揭秘

美团技术团队  
tech.meituan.com



CODE A BETTER LIFE  
一行代码 亿万生活



长按二维码关注我们

[阅读原文](#)

相关阅读

美团语音：如何从比特世界向原子世界进击？

菠萝财经 阅读 8922

---

[更多文章](#)