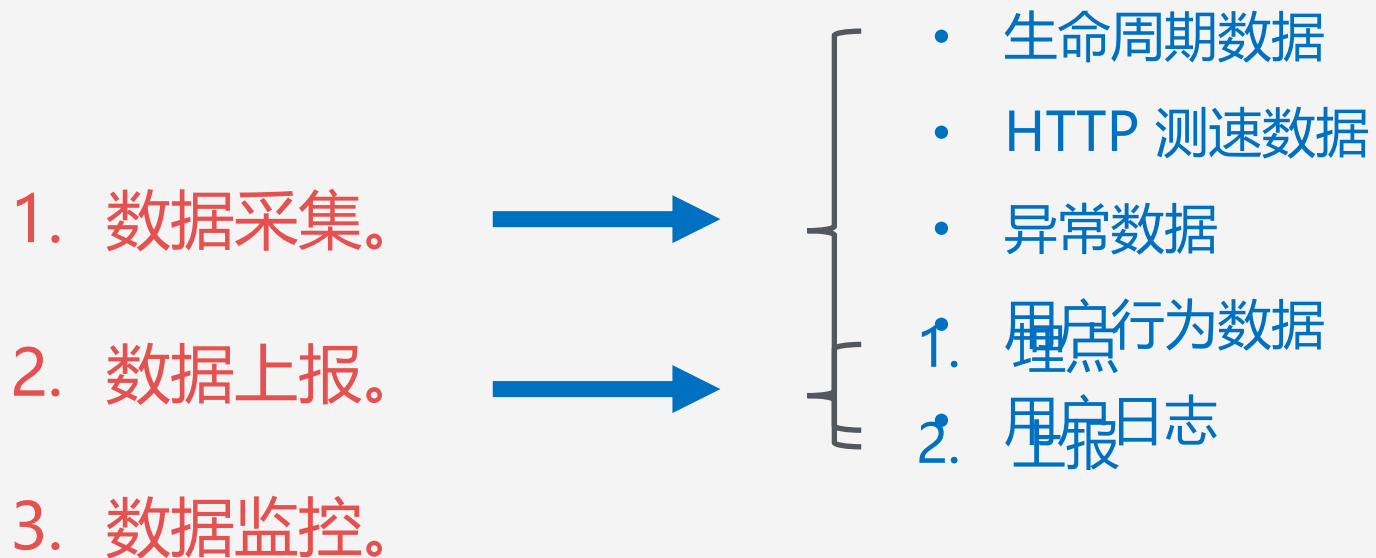


前端监控搭建之 数据上报

@被删

前端监控搭建主要做些什么？

一般来说，前端监控体系需要做的事情分三步：



1

前端数据埋点



前端常见的埋点方案

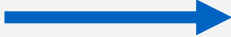
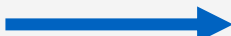
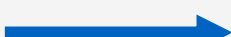
前端常见的埋点方案有三种：代码埋点、可视化埋点、无痕埋点。

-	代码埋点	可视化埋点	无痕埋点
使用方式	手动编码	可视化圈选	嵌入 SDK
自定义数据	可自定义	较难自定义	难以自定义
更新代价	需要版本更新	需要下发配置	不需要
使用成本	高	中	低
适用场景	高度自定义/难以自动化埋点场景 异常/边界埋点	UI 通用数据埋点 点击/长按埋点	通用页面生命周期数据埋点 页面打开/加载/跳转埋点

不管是哪种埋点方式，我们都需要对埋点数据进行标准化处理。

标准化埋点数据

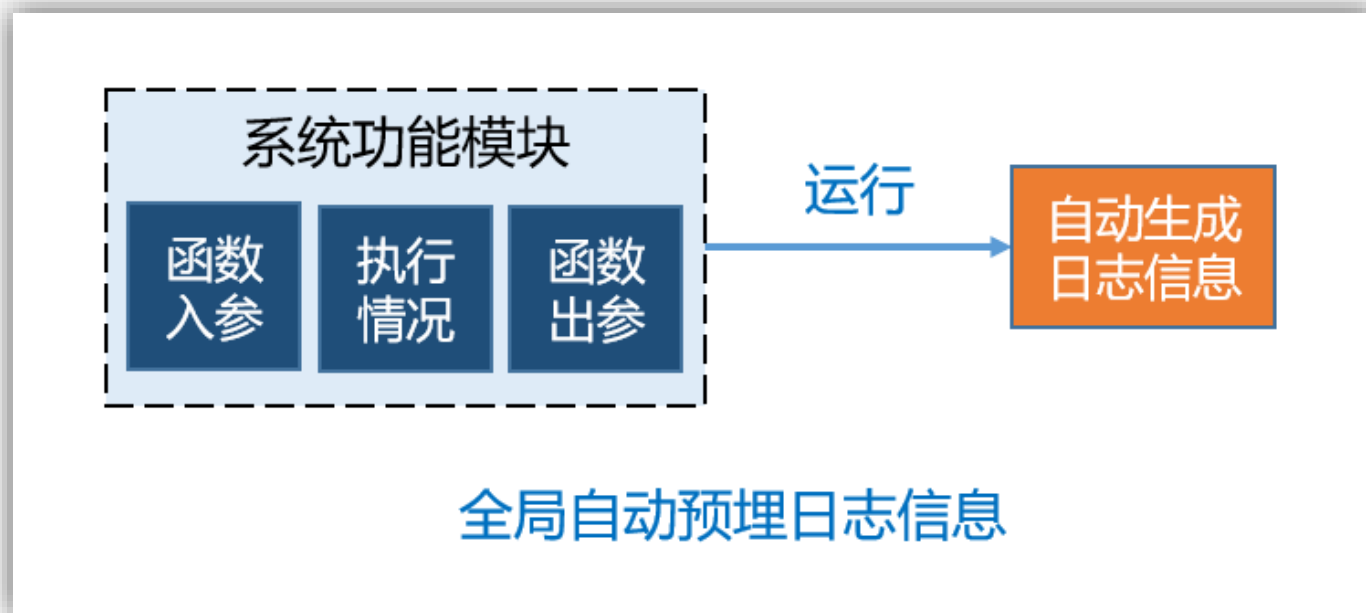
一般来说，通过和后台约定好具体的参数，然后前端在埋点采集的时候，自动转换成接口需要的一些数据格式进行本地存储，等待数据上报。

- Module: 模块  具体页面/功能模块
- Action: 操作  具体的操作/步骤
- Id: 标记 Id  用于识别和统计
- Options: 其他需要上报的数据

有效的用户行为埋点数据，既可用于单用户的链路操作追踪，也可用于整体的用户数据统计。

用户日志自动化埋点

日志通常用户定位用户问题的时候使用，但我们常常需要提前在代码中打印日志。可以通过全局代理、装饰器处理关键模块和方法等方式来进行日志的自动打印。



在每个功能模块运行时，通过使用约定的格式来打印输入参数、执行信息、输出参数，则可以通过解析日志的方式，梳理本次操作的调用关系、功能模块执行信息。

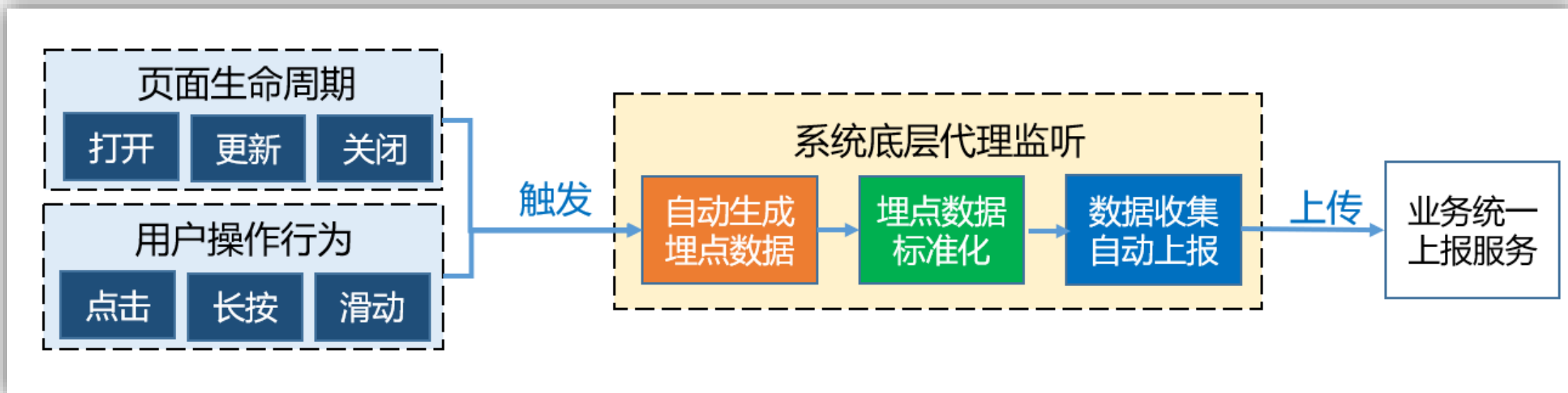
2

前端数据上报



前端采集数据进行上报

数据采集完成后，我们需要将这些数据上报到后台服务，一般还包含时间、名称、会话标记、版本号等信息。



可实时计算出埋点的数量、埋点间的执行时间、埋点间的转换率等，通过可视化系统直观地展示完整的页面使用情况。



数据上报方式

一般来说，我们埋点的数据、运行的日志都需要通过上报发送到后台服务再进行转换、存储和监控。常见的数据上报方式包括：

- 1. 定期/定量上报。 → 达到日志阈值或者定时上报
- 2. 关键生命周期上报。 → 发生异常、页面关闭前上报
- 3. 用户主动提交。 → 引导用户主动上报

用户日志上报

系统出现异常的时候，通常使用日志进行定位。日志的存储通常包括两种方案：

1. 上报到服务器。
 - 日志量过大，可能导致传输、存储、查询速度等问题
 - 过于频繁的上报，会增加服务器压力
 - 网络传输，容易导致日志丢失等问题
2. 本地存储。
 - 需要引导用户手动提交日志
 - 通过后台下发命令拉取日志

数据上报之后还要做什么？



敬请期待~



谢谢



Github: godbasin
@被删