

Bonree SDK 白皮书

北京博睿宏远数据科技股份有限公司

2021 年 7 月 2 日

目录

一、背景	4
二、企业面临的问题	5
三、产品介绍	5
四、产品特点与价值	6
1. 同时支持 HTTP、HTTPS（1.0/1.1,2.0 版本协议）、WEBSOCKET、SOCKET、QUIC（HTTP3.0）	6
2. 监测指标更细	7
3. 错误类型更多	7
4. 崩溃信息更全面	7
5. 设置更灵活	8
6. 报告数据更精确	8
7. 数据获取方式更多样	9
8. 数据采集数量自定义	9
9. 操作便捷，数据清晰	9
10. 按照用户个性化需求，灵活定制分析图表	9
五、用户受益	9
1. 帮助企业快速定位问题，节约成本	9
2. 协助持续关注 APP 性能状态，并得到实时反馈	10
3. 协助建立 APP 运营管理机制	10
4. 直观透视业务对性能的影响	11
六、功能介绍	11
1. 采集的性能指标	11
2. 功能列表	24
3. 应用概览	错误!未定义书签。
4. 性能分析	错误!未定义书签。

5.用户会话.....	错误!未定义书签。
6. 关键元素.....	错误!未定义书签。
7. 问题分析.....	错误!未定义书签。
8. 体验大屏.....	错误!未定义书签。
9. 客服检索.....	错误!未定义书签。
10. 仪表盘.....	错误!未定义书签。
11. 应用设置.....	错误!未定义书签。
12. 智能报警.....	错误!未定义书签。
13. 自动报告.....	错误!未定义书签。
14. 系统管理.....	错误!未定义书签。

一、背景

随着 5G 互联网时代的到来,国内手机电子市场的飞速发展以及智能手机的普及化,推动了 APP 应用开发市场的发展。据 2019 年 2 月 28 日中国互联网络信息中心发布的第 43 次《中国互联网发展状况统计报告》显示:

- 1、截至 2018 年 12 月,我国网民规模达 8.29 亿,全年新增网民 5635 万,互联网普及率为 59.6%,较 2017 年底提升 3.8%。
- 2、截至 2018 年 12 月,我国手机网民规模达 8.17 亿,全年新增手机网民 6433 万。网民中使用手机上网的比例由 2017 年底的 97.5%提升至 2018 年底的 98.6%,手机上网已成为最常用的上网渠道之一。
- 3、截至 2018 年 12 月,我国手机网络支付用户规模达 5.83 亿,年增长率为 10.7%,手机网民使用率达 71.4%。

截至 2018 年 12 月,我国网民上网设备中,手机使用率达 98.6%,早已成为第一大上网终端设备,同时网民在手机移动支付类、电子商务类、休闲娱乐类、信息获取类、交流沟通类等应用的使用率都在快速增长,移动互联网带动整体互联网各类应用发展。

● 全球移动互联网稳定发展,移动 APP 行业规模不断增长

2018 年全球移动 APP 下载量达到 1940 亿次,全球应用商店用户支出达到 1010 亿美元,用户平均每天在移动设备上花费的时间达到 3 小时。

● 手机 APP 带来的品牌效应

通过全方位展示产品,让消费者进一步了解品牌或产品,建立起品牌与消费者的情感关联,是企业 APP 营销的核心所在。在这个需要越来越基于消费者的情感、信任基础上的开展营销活动的时代,利用品牌 APP 传递品牌理念,神话品牌形象,梳理品牌口碑,帮助品牌和产品认知的提升,搭建起品牌与消费者间沟通的桥梁,无疑是企业营销者的明智之举。

● 手机 APP 带来的经济效益

APP 现在一般指只能手机的应用程序,APP 是目前流行的移动营销载体,是企业手

机推广的核心信息传播源，为企业塑造品牌形象和精准的促进销售。APP 是企业量身打造的移动互联网营销解决方案，能帮助企业精准的锁定目标客户群体，为企业创造看得见的经济效益。

二、企业面临的问题

面对增长如此迅速的 APP 数量，APP 同质化也日趋严重。为了提高用户的粘性（忠诚度），从众多 APP 中成功突围，保证能给公司带来持续的收益，很多公司都一直把 APP 的用户体验放在很重要的位置，包括：利益性、趣味性、内容性、创新性、社交性、操控性……等等。并且会通过持续快速迭代来进行版本的更新，满足用户的需求变化。但有时实际效果并不理想。

在分析失败原因时，往往会被定义在应用商店推广成本过高，产品定位有误，功能不达标，产品迭代速度过慢，产品逻辑问题，……等等。但却忽略了一个很重要的因素：应用性能。

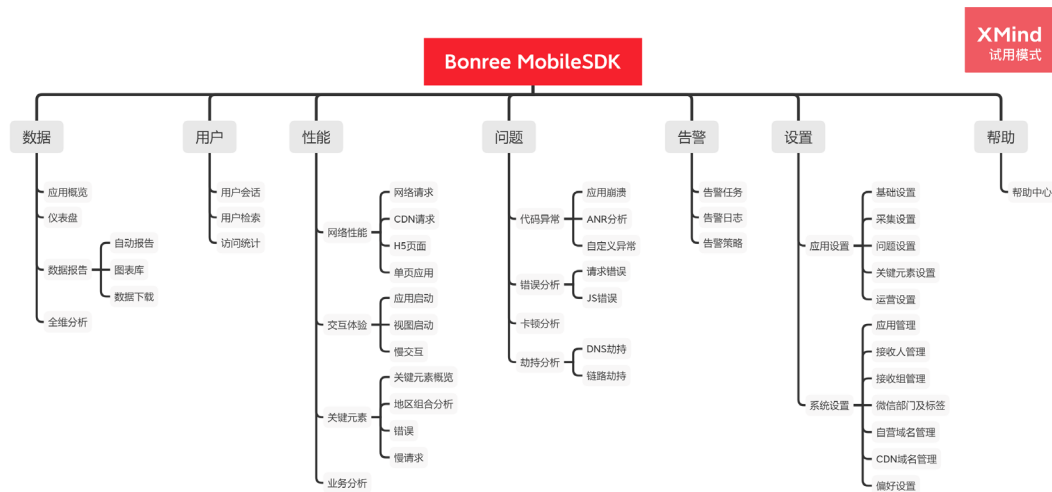
举个例子：手机用户对性能期望非常高，希望手机应用能像在电脑上运行那么快。据统计：71%用户希望在手机上打开网页能同电脑上一样快，5 秒钟被认为是用户能忍受的最长响应时间，如果响应时间超过 5 秒，74%上网用户和 50%移动应用用户会放弃，三分之一失望的用户会转向竞争对手的应用。通常，手机用户会尝试两次，如果第三次依然出现同样问题，半数人再也不会使用该应用。

三、产品介绍

Bonree SDK 监测平台（以下简称“BonreeSDK 监测平台”）基于公司核心网络监测技术，以 APP 性能监测为主，为 APP 开发者提供全面的性能监测解决方案。整个系统完全基于用户需求，从界面友好度到可操作性，再到监测数据统计与分析，Bonree SDK 监测平台都能够提供全面的功能展示，整个监测平台属于被动式监测。

Bonree SDK 监测平台分为应用概述、性能分析、关键元素、问题分析、体验大屏、客服检索、仪表盘、应用设置、智能报警、自动报告和系统管理等多个功能模块。运维人员通过丰富的报告查看监测结果，对 APP 运行中的性能状态进行分析；也可对每一个 APP 进行

多维度的智能报警设置，降低由性能因素带来的损失。除此之外，灵活的过滤器设置，既可按 APP 运行需要进行数据多维度过滤，所需问题范围，也可有针对性优化。



Bonree SDK 监测平台功能结构

整个监测属于被动式监测。在对 APP 进行监测前，用户需要将 SDK 包植入到 APP 中，当用户开启 APP 后，性能数据将会被自动抓取，并根据预设的数据采集规则回自动回传到 Bonree SDK 监测平台进行分析。用户可通过登录 Bonree SDK 监测平台查看被检测 APP 的分析报告。

- 使用人员可通过“应用概览”模块，宏观掌握 App 质量，判断问题归属；
- 运维人员可通过“性能分析”模块，全面了解 APP 性能情况，及优化环节；
- 研发人员可通过“问题分析”模块，复现异常问题，快速排查并处理；
- 客服和研发人员可通过“客服检索”模块，对终端环境或异常问题进行信息检索，提高问题排查速度及服务主动性。
- 运维人员，研发人员，客服人员，业务人员，可通过使用“用户会话”模块，追溯用户使用全过程，查看问题出现的前后路径，了解用户使用方法等。

四、产品特点与价值

1. 同时支持 HTTP、HTTPS（1.0/1.1,2.0 版本协议）、WEBSOCKET、SOCKET、QUIC（HTTP3.0）

特点

- Bonree SDK 监测平台，同时获取 HTTP、HTTPS、WEBSOCKET、TCP、QUIC 通讯协议的性能数据，HTTP 和 HTTPS 支持 1.0/1.1 版本、2.0 版本、3.0 版本（QUIC）协议，

了解整个请求过程中每一个阶段的耗时情况，真正做到 APP “全方位” 监测，从而帮助用户快速定位故障点，减少故障历时和运维成本。

价值

- 帮助用户“完整”了解 APP 性能情况，让 APP 的性能状况更透明。

2. 监测指标更细

特点

- 细化到以请求为单位，不但可以监测每个请求的整体耗时，更可以监测整个请求过程中每一个阶段的耗时，如：DNS 耗时、TCP 建连耗时、请求耗时、响应耗时、传输耗时。

价值

- 对性能状态进行更细致的剖析，能让用户更快速、更精确地定位 APP 的性能瓶颈，缩短了 APP 瓶颈的分析及定位时间，减少了运维成本。

3. 错误类型更多

特点

- 不但可采集 HTTP 错误，更可采集 TCP 错误，全面反映 APP 发生的网络错误；

价值

- 充分了解 APP 异常的原因，避免因网络错误统计不全而产生的误判；

4. 崩溃信息更全面

特点

- 将 APP 崩溃问题按照具体原因进行统计分类，获取每一种崩溃原因的详细崩溃堆栈信息。此外，还可以对每一崩溃原因进行 OS 版本、设备型号和 APP 版本的分布分析，进一步定位可能引起崩溃的因素，并且可直接标记问题状态，便于对问题进行管理。平台同时为每一个崩溃样本提供出现崩溃前的操作路径。

价值

- 便于研发人员定位产生崩溃的原因，研发人员可以判断每一种崩溃是否与设备类型/OS 版本/APP 版本有关，以便及时修复或发布通知；

- 可以快速对问题进行复现，提高修改效率；
- 可直接在平台上对崩溃问题进行管理，快速过滤未解决或再次出现的问题。

5. 设置更灵活

特点

- 可按照地域或主机进行更精确的报警设置，系统报警时，可定位到具体的地域或主机；
- 对于接收到的样本数据可以进行报警周期的自定义的设置；
- 系统提供完备的报警条件设置，包括设置样本数量、持续周期、异常样本比例等条件；

价值

- 对网络性能风险进行提前报警，可以减小用户的损失；
- 把报警具体定位到某个地区或某台主机，可以帮助用户判断是普遍存在的网络性能隐患，还是只是某个局部存在隐患，便于准确处理；
- 可避免因样本数量不足而产生的无效报警，

6. 报告数据更精确

特点

- 灵活的过滤器设置，可以为用户过滤掉无效或不想关注的数据，使报警和报告更精准。
- 平台提供超过 60 种报告，对业务数据和性能数据分别进行了统计分析；
- 对于性能数据，可以从视图、主机、域名、地域、运营商、接入方式、OS 版本、设备型号、APP 版本等维度进行全面分析；

价值

- 用户不用担心在报警的样本中掺入了因信号质量导致的响应时间过长的样本，灵活的过滤器设置可以将这些数据排除，使报警变得更真实、更有价值；
- 当用户只想关注某些特定人群时（例如：重要的城市、某个运营商、某种接入方式），或想讲信号不好的样本数据排除掉时，可以通过过滤器对报告进行过滤，真正做到组成报告的每一个样本数据都是用户真实需要的；
- 多维度的分析报告，为用户提供了定位问题的依据；

7. 数据获取方式更多样

特点

- 所有报告都提供图形和数据的导出；
- 灵活的接口设计，可对原始数据进行导出；
- 完善的自动报告设计，可周期性生成一个或多个性能检测报告，按照发送规则发送到收件人的邮箱中；

价值

- 用户可以通过对原始数据的导出，在线下进行二次分析；
- 完备的报告导出机制，使用可以摆脱设备和地域的限制，方便用户及时获取应用性能信息；

8. 数据采集数量自定义

当客户不愿意泄露真实活跃设备数时，可以通过自定义采集数量进行随机采集来控制监测客户端的数量，以达到数据保密的目的。

9. 操作便捷，数据清晰

通过 Bonree SDK 监测平台对 APP 进行实时监测的操作方法并不复杂，只要按照使用说明，在 APP 中植入几行代码即可开始对 APP 进行样本采集。Bonree SDK 监测平台的用户界面也非常友好，除了必要的文字说明外，其核心的报告也都清晰易懂，对于运维和研发人员来说，学习成本基本为 0。

10. 按照用户个性化需求，灵活定制分析图表

支持用户自行定义分析的数据项，指标，指标聚合方式，组合维度，图形类型，筛选条件等，按照自己的需求生成分析图表，对于数据分析更加灵活。

五、用户受益

1. 帮助企业快速定位问题，节约成本

当 APP 发布后，不但能很难预知是否会出现各种问题，而且很难定位问题原因。APP 的运营人员往往只能看到结果。当他们发现客户在流失，公司收益在降低时，需要运维以及研

发人员投入很多的时间、很大的精力去分析问题、定位问题原因。但是否真的有效果，还要持续关注 APP 的运营结果来判断。这对公司来说，成本高，效率低。通过使用 Bonree SDK 监测平台，运营人员可以直观的根据活跃设备数的变化来判断用户数量的变化情况；运维人员可以清晰的了解当前 APP 的应用性能情况，是否是因为非代码原因导致的 APP 用户体验下降，客户的流失，准确发现并定位网络性能异常的原因；研发人员可以根据崩溃信息及时了解发生崩溃的频率、覆盖的设备类型、操作系统，甚至可以复现每一个崩溃发生的过程，便于快速解决问题。对公司来说，既高效、有节约了成本。

2. 协助持续关注 APP 性能状态，并得到实时反馈

对于 APP 开发公司来说，往往需要实时关注已经上线的 APP 的状态。一方面，可以及时发现崩溃以及应用性能问题并第一时间处理；另一方面，可以根据使用者的使用结果对 APP 进行进一步优化。但实际中，企业并没有专人实时查看监测结果，当 APP 出现问题时，并不能及时发现。

通过使用 Bonree SDK 监测平台，一旦开始对 APP 进行检测，系统会持续不断地接收所采集的样本数据，接收的周期为每分钟一次，以保证用户每次看到的报告数据都是最新的，真正做到对 APP 的实时跟踪。并且当发现潜在的网络性能隐患或崩溃率达到一定标准时，会以报警的形式第一时间通知运维人员，引起运维人员的重视，及时处理，最大限度降低损失。

3. 协助建立 APP 运营管理机制

管理层可依据平台给出的各维度、各层面的技术指标，及真实用户的体验数据。将问题第一时间定位到相关职能部门，大大减小故障响应时间，提升工作效率

对已一款 APP 来说，有可能只面对某一部分人群，例如：高端品牌的电商 APP，有可能客户群体只针对一线城市。但是开发者并不能控制只有一线城市居住人群安装此 APP。但过多的非一线城市的采集样本会影响到分析结果的真实性。

Bonree SDK 监测平台为开发者提供自定义过滤器功能，通过过滤器设置，可以将不关注的使用群体进行过滤，只针对某一类特定人群进行分析，使分析结果更有价值。

4. 直观透视业务对性能的影响

决定企业 APP 价值的四因素其中之一就是持续的产品功能和性能改善。随着公司的业务拓展与调整，势必需要进行相应版本的迭代和功能更新。新功能的增加是否仍然保证 APP 具有高性能，是否 APP 的性能随着功能的变化已经在无形中降低而产生了用户的流失？如果没有专业的监测工具，是很难实时了解的。Bonree SDK 监测平台所展现的数据都是实时的，能够帮助企业在由于业务变动而产生功能变动的同时，了解 APP 性能的情况，真正让企业透视业务与性能的关联。

六、功能介绍

1. 采集的性能指标

一级分类	二级分类	三级分类	指标	定义和说明
数据&性能	概览 & 关键元素 概览	概览	活跃会话数	在指定时间范围内，客户端在使用的所有会话的数量。
			活跃设备数	在指定时间范围内，所有活跃的设备 ID 去重的数量
			响应时间	当前应用所有请求响应时间的平均值，包含了自营域名和第三方域名的所有请求。
			下载速度 (CDN)	平均每秒的下载数据量，一般是资源类（图片、文件、音视频等）下载 下载速度=下载大小/接收用时
			请求错误率	发生请求错误次数与 APP 总请求数比率。 请求错误率=错误请求次数/总请求次数*100%
			应用崩溃率	在指定时间范围内，崩溃次数与用户访问量的比率。 应用崩溃率=应用发生崩溃的次数/应用启动次数*100%

			HTTP 错误率	所有请求发生 HTTP 错误的占比
			网络错误率	所有请求发生网络错误的占比
		拓扑	APP 应用	被监测 APP
			视图启动时间	指定时间范围内容，视图启动时间
			视图慢启动率	视图慢启动率=视图慢启动次数/视图访问次数
			应用卡顿率	该应用发生卡顿的活跃会话数与总活跃会话数比率
			自定义异常率	某维度应用发生自定义异常的活跃会话数/该维度总活跃会话数 *100%
			ANR 率	某维度发生 ANR 的活跃会话数次数/该维度总活跃会话数*100%
			慢交互率	应用发生慢交互的活跃会话数与总活跃会话数比率
			JS 错误率	发生 JS 错误的页面访问次数与该页面的总访问次数
			资源类型	资源类型划分为：自营、CDN 和第三方等类型。
			首包时间	从请求发起到成功接收到服务器第一个响应包的时间。
			错误率	请求发生错误的次数/请求总次数比率
			每分钟请求数	平均每分钟的总请求次数
			慢速比	慢请求所占全部请求的比例，慢速比=慢请求次数/正确请求总次数*100%
用户	用户会话	--	活跃设备	所选时间范围内，当前应用所有活跃的设备 ID 去重的数量
			活跃会话	所选时间范围内，当前应用所有活跃的且至少有一个有效事件的会话的去重数量
			活跃用户	所选时间范围内，当前应用所有活跃的用户 ID 去重的数量

			异常会话	所选时间范围内，当前应用所有有效的活跃会话中，异常的（包含了崩溃/ANR，卡顿，自定义异常类事件）的会话的数量（取完整会话）
			健康率	所选时间范围内，当前应用所有健康会话的数量占全部会话数量的比例（取完整会话）
			平均事件数	所选时间范围内，当前应用所有活跃会话中的事件总数/总的有效的活跃会话数（取完整会话）
			平均持续时间	所选时间范围内，当前应用所有有效的活跃会话的总持续时间/总的有效的活跃会话数（取完整会话）
			平均会话持续时长	所选时间范围内，当前设备/用户的所有出现的有效活跃会话的平均持续时长（取完整会话）
			平均会话事件数	所选时间范围内，当前设备/用户的所有出现的有效活跃会话的平均事件数量（取完整会话）
	访问统计	活跃设备统计	日活跃	在指定时间范围内，访问过 APP 的独立设备数量（每日去重）
			月活跃	在指定时间范围内，访问过 APP 的独立设备数量（每月去重）
			平均日活	在指定时间范围内，平均每日活跃设备数量 日平均活跃设备数=日活跃设备数量总和/天数
性能	网络性能 & 关键元素	网络请求 & 关键元素地区组合	平均首包时间	指定时间范围内，当前 App 的平均首包时间。
			累计响应时间	在指定时间范围内，所有请求响应时间总和。
			平均响应时间	指定时间范围内平均响应时间。
			吞吐率	平均每分钟的总请求次数（正确请求次数+错误请求次数）

业务分析	关键元素慢请求	平均传输数据量	选定时间内某请求的平均下载大小，一般为资源类（图片、文件、音视频等）的下载。 平均传输数据量=一个请求（合并）总下载量/该请求正确的请求总次数
		首包时间	从请求发起到成功接收到服务器第一个响应包的时间。 首包时间=DNS 用时+TCP 用时+（SSL 用时）+请求用时+服务响应用时 说明：平台未展示请求用时数据
		DNS 用时	域名解析时间
		TCP 用时	建立 TCP 连接的时间
		SSL 用时	SSL 握手时间
		服务响应用时	请求结束到客户端成功接收到服务器第一个响应包之间的时间间隔
		响应时间	请求从发起到结束的完整时间。 响应时间=首包时间+下载时间
		正确请求次数	选择时间范围内，排除错误请求的所有请求次数。
		错误率	发生请求错误次数与 APP 总请求数比率。 请求错误率=错误请求次数/总请求次数*100%
		总请求次数	在选定时间范围内，请求的总次数，包括正确和错误
		健康请求次数	健康标准内的正确请求总次数
		慢请求次数	超过健康标准的正确请求总次数

			慢速比	慢请求数量占全部请求的比例，通过该指标掌握请求受损范围，确定问题优先处理程度。 慢速比=慢请求次数/总请求次数*100%
			错误请求次数	发生错误的请求出现的总次数 包含但不限于 TCP、SSL、DNS 等网络错误以及 HTTP 错误等。
			下载大小	该请求的下载数据量
			接收用时	下载数据所消耗的时间
			请求地址	以请求维度，显示请求的 URL 地址
			主机	以主机维度，显示主机 IP
			HTTP (s) 请求	带有标准 HTTP(s)协议头的请求
			Socket 请求	不带有标准 HTTP(s)协议头，与用户在 socket 白名单中用户配置 的请求一致的请求
			请求快照	慢请求深度分析：显示请求的慢请求次数、请求次数、慢速比及慢请求下载大小，下方展示慢请求各环节耗时及性能分布情况 (地域、运营商、主机等) 慢请求快照：发送时间、请求地址、城市运营商接入方式、客户端 IP、主机、LocalDNS、慢请求首包时间、响应时间、慢请求 DNS/TCP/SSL/服务响应/接收等用时、慢请求下载大小、服务端处理用时、服务端快照（与 Server 平台关联）。
			曝光率	系统默认隐藏小于 1%曝光率，主要是隐藏出现的概率小、性能有 可能大的请求，便于用户优先处理影响面大的请求问题。计算公 式：

			曝光率= 第一层某个维度的请求次数/全部第一层维度的请求次数 网络请求中的曝光率=单个域名的全部（正确+错误）请求次数/所有域名全部（正确+错误）请求总数*100%
		整体性能	页面的加载用时，即页面从开始访问到全部打开的整体耗时。 loadEventEnd - navigationStart
		首包时间	从页面开始浏览到收到服务器第一包数据的时间。 responseStart-navigationStart
		HTML 下载用时	浏览器下载完整的 HTML 文档内容所需要的时间 responseEnd-responseStart
		DOM 建立时间	浏览器建立文档对象模型（DOM）的用时 domContentLoadedEventStart-domLoading
		资源加载时间	从浏览器开始下资源下载到页面渲染完成的时间间隔 loadEventEnd-domContentLoadedEventStart
		吞吐率	平均每分钟的总访问次数
		正确访问次数	选定时间范围内容，排除 JS 错误访问的所有访问次数。
		总请求次数	该页面总的请求次数。
		JS 错误率	页面发生 js 错误的比率 JS 错误率=发生 JS 错误的页面访问次数/该页面的总访问次数 *100%
		JS 错误次数	js 发生的总次数。如一次页面访问中出现过 10 次 js 错误，则错误次数为 10

		曝光率	单个页面的全部（正确+错误）访问次数/所有页面全部（正确+错误）访问总数*100%，默认隐藏小于 1%曝光率。
		下载速度	下载速度=总下载字节数/总响应时间
		错误率	错误率=CDN 请求错误次数/CDN 总请求数*100%
		慢速比	整体速度慢的请求数量占全部请求的比例 慢速比=慢请求次数/总请求次数*100%
		流量	监测时间内全部请求的数据传输量（下载字节数）
	交互体验	视图性能	平均启动时间 监测时间范围内，视图启动时间的平均值。 平均启动时间=视图总启动时间/视图启动次数
			累计启动时间 监测时间范围内，所有视图启动时间总和
			启动次数 监测时间范围内，视图的总启动次数
			视图慢启动率 视图慢启动率=视图慢启动次数/视图访问次数
			健康启动次数 监测时间范围内，符合健康标准的视图启动次数
			慢启动次数 监测时间范围内，超出健康标准的视图启动次数
			总启动次数 监测时间范围内，视图总启动次数
		应用启动	首次启动 应用安装后的第一次启动，由于有一些应用在安装后的首次启动会进行一些初始化的操作，包括会有首次启动的一些引导设计，因此首次启动相较于之后的冷启动可能存在一些不同点
			冷启动 当启动应用时，后台没有该应用的进程，此时系统会重新创建一个新的进程分配给该应用，这个启动方式就是冷启动。
			热启动 当启动应用时，后台已有该应用的进程（例：按 home 键回到桌面，但是该应用的进程依然会保留在后台，可进入任务列表查

			看)，所以在已有进程的情况下，这种启动会从已有的进程中来启动应用，这个方式叫热启动。通常来说热启动的耗时会比冷启动短。
		平均启动时间	按照应用的启动方式分别统计冷启动耗时和热启动耗时，用各自启动耗时的总和除以各自的启动次数，即为冷启动和热启动的平均启动时间
		累计启动时间	按照应用的启动方式分别统计冷启动耗时和热启动耗时的总和，即为冷启动和热启动的累计启动时间。
		启动次数	按照应用的启动方式分别统计冷启动次数和热启动次数，需要注意的是这里只统计正常启动有启动耗时的次数，对于使用旧版 SDK 不支持采集启动耗时，或闪退没有采集到启动耗时的次数不计算在内，因此会比概览总的启动次数要少一些。
	交互分析	交互展示维度	按操作系统版本、设备型号和 APP 版本等三种维度进行展示。 面积图中的方块为某一维度的具体信息的慢交互（如：维度为设备型号，面积图中的方块为不同型号的手机慢交互信息）
		慢交互次数	监测时间范围内出现过慢交互的视图数量。 判断慢交互的标准：视图主线程任一方法的耗时超过指定值（默认 500ms）
		慢交互占比	指定时间范围内， $\text{慢交互占比} = \text{视图出现慢交互的次数} / \text{全部视图访问次数} * 100\%$
		慢交互率	$\text{慢交互率} = \text{应用发生慢交互的启动次数} / \text{应用总启动次} * 100\%$
		慢交互视图数	在指定时间范围内，发生慢交互的视图数量次数

问题	代码异常		修复问题数	在指定时间范围内，状态为已修复的慢交互问题数量
			交互视图	发生慢交互的视图名称
			视图别名	发生慢交互视图别名
			平均耗时	方法的平均耗时，即视图方法的总耗时/该视图方法的总调用次数
			发生次数	某种慢交互发生次数
		应用崩溃 & 自定义异常	崩溃维度	崩溃维度可按设备型号、操作系统版本和 APP 版本等维度进行展示（默认按设备型号）
			崩溃次数	某种崩溃问题发生的次数
			崩溃占比	崩溃占比=某种崩溃问题发生次数/全部崩溃问题发生次数*100%
			崩溃率	崩溃率=应用崩溃次数/应用启动次数*100%
			发生次数	在指定时间范围内，启动该 APP 时发生的崩溃次数。
			影响用户数	在指定时间范围内，按照崩溃设备 ID 去重后的设备数量进行统计
			问题数	在指定时间范围内，崩溃问题的数量
			修复问题数	在指定时间范围内，状态为已修复的崩溃问题数量
			优先级	优先级分为：高、普通、低等三个级别
			状态	问题状态分：全部激活、首次激活、多次激活、已修复等四种
			最后修复时间	崩溃问题最后修复的时间。
			是否修复	该崩溃问题修复修复
			崩溃问题 ID	数据汇总时系统自动生成，对每一次崩溃进行 ID 标识，便于后期问题检索。
			崩溃问题	展示崩溃的问题信息、原因及管件方法
			发生次数	发生该类崩溃的次数。

		异常问题 ID	数据汇总时依据自定义问题系统自动生成，便于后期问题检索。
		自定义异常问题数	在指定时间范围内，自定义异常问题的数量
		自定义异常占比	某种类型的自定义异常发生次数/全部类型的自定义异常发生次数 *100%
		应用快照	显示崩溃轨迹及堆栈、系统日志、环境信息、自定义附加信息和崩溃信息等内容： 1) 崩溃轨迹：按系统设置的步数，并以倒叙的方式显示崩溃发生前的步数信息，包括：发生崩溃前的时间、视图及操作方法； 2) 崩溃堆栈：显示崩溃线程名和堆栈记录，以及除崩溃线程以外的其他线程和堆栈信息； 3) 系统日志：显示当前快照的崩溃日志； 4) 环境信息：显示当前快照的设备信息，如：CPU 的厂商/型号/指令集/用量、内存消耗、内存剩余、剩余电量、是否越狱/root、APP 名称、UUID、基址等信息； 5) 自定义附加信息：通过 setuserinfo API，基于 memberID 调取的用户信息。
		设备 ID	1、Android 设备 ID 优先选择 IMEI 或 MEID。 如果 IMEI 或 MEID 获取不到，则获取 SN 码 (SerialNo.)； 如果 SN 码也获取不到，则获取 AndroidID； 如果 Android 仍然获取不到，则置为 NaN 2、IOS 设备选择 IDFA 作为设备 ID。
		会员 ID	平台开放接口与企业的会员信息对接，便于对终端用户的识别。

	ANR 分析	展示维度	按设备型号、操作系统版本和 APP 版本等三种维度进行展示
		ANR 次数	某种 ANR 问题发生的次数
		ANR 占比	ANR 占比=某种 ANR 问题发生次数/全部 ANR 问题发生次数*100%
		ANR 率	某维度发生 ANR 的应用启动次数/该维度应用启动次数*100%
		发生次数	在指定时间范围内，启动该 APP 的 ANR 次数
		影响用户数	在指定时间范围内，按照 ANR 设备 ID 去重后的设备数量进行统计
		问题数	在指定时间范围内，ANR 问题的数量
		修复问题数	在指定时间范围内，状态为已修复的 ANR 问题数量
	卡顿分析	卡顿次数	某类视图发生卡顿的次数
		卡顿占比	卡顿占比=内某一视图的卡顿次数/全部视图的卡顿次数
		卡顿率	卡顿率=应用发生卡顿的启动次数/应用总启动次*100%，视图启动一次出现多次卡顿，只算一次卡顿
		发生次数	在指定时间范围内，发生卡顿的次数。 1) 视图必须出现慢交互，再判断是否出现卡顿。 2) 判断卡顿的逻辑为： Android 卡顿判断：使用流畅度来判断，当平均流畅度在 40 以下时用户能够感知到卡顿，我们定义平均流畅度<40 的为出现卡顿。 说明：流畅度是指系统每 60ms 发送一次 vSync 信号，如果前端渲染进程能够在 16ms 内完成，则每秒钟的帧率 (fps) 可以达到

			60 帧。我们规定 1s 内 VSync 这个 Loop 运行的次数表示流畅度。 IOS 卡顿判断：通过监测主线程 Runloop 的循环时间是否超过一定值（5s），超过则判断发生了卡顿。
		影响用户数	监测时间范围内，出现过卡顿的用户数，通过用户使用设备的唯一 ID 进行统计
		卡顿视图数	监测时间范围内，出现过卡顿的视图数量
		修复问题数	在指定时间范围内，状态为已修复的卡顿视图的数量
		视图名称	发生卡顿的视图名称
		视图别名	发生卡顿视图的别名，便于非研发人员理解
		平均流畅度	监测时间范围内，视图流畅度的平均值（仅 Android 支持该指标）
		最小流畅度	监测时间范围内，视图流畅度的最小值（仅 Android 支持该指标）
错误分析	请求错误	错误请求维度	错误分析涉及请求错误和 H5 页面错误等内容。按资源类型划分，错误请求分为：自营、CDN 和第三方等类型。 错误请求维度包括：域名、地区、运营商、接入方式等展示维度；
		错误次数	请求发生的错误次数，按分类、按错误类型、按维度统计
		错误占比	错误占比=某错误类型的错误次数/该错误类型所在分类全部错误次数*100%
		错误率	错误率=请求发生错误的次数/请求总次数*100%，按维度

		总请求次数	指定时间范围内容，发生的总请求数。
		发生时间	发生错误的开始时间和结束时间
		域名	发生错误的域名
		错误 URL	错误请求的 URL 地址
		错误分类	按 HTTP 错误、DNS 错误、TCP 错误、SSL 错误及其它等进行分类
		错误快照	发生时间、请求地址、APP 版本、设备型号、错误类型、错误代码、会员 ID、地区、运营商、接入方式、客户端 IP、主机、DNS 地址、操作（错误详情显示该错误请求的请求参数、请求头、响应头等信息）
		错误类型	针对错误分类进行详细划分，便于运维人员快速识别问题点。 1) HTTP 错误：3XX、4XX、5XX 等错误 2) DNS 错误：DNS 解析失败、Memory allocation failure、No address associated with hostname、Resolved protocol is unknown 等相关错误； 3) TCP 错误：建连失败、No route to host、Connection refused、Connection timed out、Connection reset by peer 以及 Software caused connection abort 等； 4) SSL 错误：SSL 握手未完成； 以及其它错误类型
	JS 错误	错误次数	发生 JS 的总次数。如一次页面访问中出现过 10 次 js 错误，则错误次数为 10

			错误率	JS 错误率=发生 JS 错误的页面访问次数/该页面的总访问次数 *100%
			错误占比	当前错误类型的错误次数占全部类型错误次数的比值*100%
			JS 错误快照	显示每条快照信息的 UserAgent 数据和堆栈数据
			错误类型	JS 错误的类型共分七类：涉及 SyntaxError, ReferenceError, RangeError, TypeError, URIError, EvalError 以及其他自定义错误等。
	劫持分析	DNS 劫持 & 链路劫持	劫持维度	按 DNS 劫持和链路劫持两种方式进行展示。
			DNS 劫持次数	域名：某域名发生 DNS 劫持的次数； 地区：某地区发生 DNS 劫持的次数
			DNS 劫持率	劫持率=DNS 劫持次数/该域名总请求次数*100%
			链路劫持率	出现链路劫持启动次数/总启动次数（大屏算分过程用用到）
			链路劫持次数	域名：某域名发生链路劫持的次数； 地区：某地区发生链路劫持的次数

2. 功能列表

一级分类	二级分类	三级分类	简介
数据	应用概览	--	展示 App 的启动次数、响应时间、下载速度（CDN）、请求错误率、应用崩溃率等关键指标，并可进一步钻取。
	仪表盘	--	用户可将经常关注的图表收藏到仪表盘中，每次登陆直接进入仪表盘，查看关注内容即可。
	数据报告	自动报告	配置自动报告，可定期将指定的图标发送到邮箱
		图表库	收藏图表到图表库中，以备配置自动报告配置时引用

		数据下载	下载的数据条数较多时会转为系统后台下载，之后可在本模块中随时下载至本地
	全维分析	--	允许用户根据自己的使用需求，自由选择查看的数据项，指标，指标聚合方式，维度组合，排序方式，图形类型，过滤条件等，满足个性化的数据分析场景
用户	用户会话	--	展示会话的分布、趋势、用户、详情等情况，展示用户在查询时间内访问目标网站应用的全部会话记录，并可详细查看每一次会话的完整过程。
	用户检索	--	允许用户通过设置查询条件对不同的功能模块中的数据项进行查询。目前在设备 ID、用户 ID、用户信息、请求 URL 四个维度下，支持对慢请求、错误请求、应用崩溃、自定义异常、卡顿、慢交互、ANR、JS 错误、关键元素性能、关键元素性能进行检索。
	访问统计	--	该模块可从总览、设备型号、APP 版本、操作系统版本和地区等五个维度进行活跃设备统计信息展示。
性能	网络性能	网络请求	网络请求可按三部分进行分析：数据切换区、列表区、图形区。请求快照可与用户会话模块实现关联。
		CDN 请求	平台依托链路层性能指标抓取的优势，针对 CDN 性能进行监测。
		H5 页面	对 H5 页面性能进行监控。H5 页面快照可与用户会话模块实现关联。
		单页应用	对使用单页应用框架的路由切换耗时进行监控分析
	交互体验	应用启动	监控应用的首次启动、冷启动、热启动性能，并可分析 App 版本、操作系统版本分布，以及启动过程的阶段和方法耗时分析
		视图启动	监控应用的原生视图启动耗时，并可分析 App 版本、操作系统版本分部，以及启动过程的方法耗时分析
		慢交互	通过对慢交互线程调用逻辑、耗时分析，展现主线程方法的执行效率。
	关键元素	关键元素概览	为使用者提供关键元素的概要信息。

		地区组合分析	按地区、运营商、接入方式的维度对关键元素的性能进行分析。
		错误	按地区、运营商、接入方式的维度对关键元素的错误进行分析。
		慢请求	显示慢请求快照列表。
	业务分析		对请求地址或规则从业务层面进行分组，从业务角度分析性能。慢页面快照可与用户会话模块实现关联。
问题	代码异常	应用崩溃	监控应用的崩溃，并可通过崩溃堆栈，环境信息，系统日志，自定义附加信息对崩溃问题进行定位和分析，同时可关联查看崩溃时的用户操作轨迹
		自定义异常	提供接口，供用户调用上报自定义异常信息
		ANR 分析	针对 Android 版 APP 独有的分析模块，切换到安卓版本下才会显示。
	错误分析	请求错误	对请求的各类型错误进行监测，实现对网络请求、H5 页面错误的全面分析。
		JS 错误	从操作系统、页面和错误类型等维度进行错误规律分析，并进一步钻取错误详情。
	卡顿分析	--	主要获取卡顿轨迹、线程、方法和对应终端的内存 CPU 等信息。
	劫持分析	DNS 劫持	监控并分析应用中请求的域名在解析阶段被劫持到其他主机 IP 的问题
		链路劫持	监控并分析应用中请求在链路中被劫持返回其他劫持请求的问题
告警	智能告警	告警任务	展示系统中各模块的报警列表。
		告警日志	系统将记录报警日志，在报警模块的报警日志页面中可对日志信息进行查看。
		告警策略	模块可对报警策略进行创建、编辑、复制和查看。用户在创建策略出的过程中，可以选择使用动态基线或者静态两种方式。
设置	应用设置	基础设置	可设置健康度阈值，管理域名别名、请求别名、H5 页面别名、视图别名、主机别名、单页应用别名，对基线进行管理，管理自定义筛选器

		采集设置	可设置样本控制，模块开关，请求地址过滤，请求优先展示规则，请求地址合并规则，请求业务分组，H5 页面地址合并规则，采集阈值，用户会话设置
		问题设置	可设置劫持配置，符号化文件管理，崩溃堆栈合并
		关键元素设置	可配置管理关键元素规则
	系统设置	应用管理	可创建和管理应用
		接收人管理	可创建和管理接收人
		接收组管理	可创建和管理接收组
		自营域名管理	可管理账号下所有应用公共的自营域名
		CDN 域名管理	可管理账号下所有应用公共的 CDN 域名
		偏好设置	针对在平台中查看数据的偏好进行设置
	帮助中心	--	将产品使用上的问题通过 Q&A 形式展示给用户。