

腾讯云可观测平台

终端性能监控

产品文档





【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有,未经腾讯云事先书面许可,任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算(北京)有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标,依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况,部分产品、服务的内容可能有所调整。您 所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定,除非双方另有约定,否则, 腾讯云对本文档内容不做任何明示或默示的承诺或保证。



文档目录

终端性能监控概述 终端性能监控概述 接入指南 安卓应用场景 集成和初始化 功能配置 网络监控 WebView、JsError、Web 网络监控

Crash、ANR 监控

卡顿、帧率监控

启动监控

iOS 应用场景

集成和初始化

控制台操作指南

崩溃

ANR

卡慢

启动

网络

Webview

应用管理



终端性能监控 终端性能监控概述

最近更新时间:2024-05-14 12:36:06

简介

终端性能监控是全方位定位检测 App 应用性能和用户体验的工具,多维度自动分析出各维度存在可疑的性能缺陷。 帮助您精确衡量 App 应用的性能,以低成本、高效率发现 App 应用各类问题。

APP 监控功能说明

功能名称	说明
崩溃 (Crash)	通过对崩溃问题个例提取关键特征进行聚合,便于您定位分析崩溃根因。
启动	支持启动耗时、慢启动占比等进行启动指标分析,您可通过慢启动问题列表定位分析慢启动根 因。
卡慢	通过流畅度等指标分析,便于您分析 App 页面性能情况。
ANR	多维度还原线上用户真实体验,通过收集用户真实使用 App 过程中碰到的 ANR 问题及问题发生时各线程堆栈信息,提取关键特征聚类。
网络	通过请求耗时、慢请求占比、网络错误率等指标进行网络问题分析。
WebView	通过页面加载耗时、慢加载占比以及 JS 错误率等进行 WebView 指标分析,并通过问题列表 对 WebView 以及 JSError 问题进行下钻。

数据存储说明

问题个例数据(慢启动、慢请求等问题列表):存储 60天。 15分钟的指标数据(指标分析界面):存储30天。 1小时的指标数据(指标分析界面):存储30天。



接入指南 安卓应用场景 集成和初始化

最近更新时间:2024-05-24 11:56:45

操作场景

本文指导您使用 Android SDK 的集成与初始化。

操作步骤

步骤一:Gradle 集成

1. 在 settings.gradle 中添加 maven 仓库源。







```
pluginManagement {
    ...
    repositories {
        ...
        // 加入下面内容
        maven { url 'https://qapm-maven.pkg.coding.net/repository/qapm_sdk/android_
        }
    }
    dependencyResolutionManagement {
        ...
    }
}
```



```
repositories {

...

// 加入下面内容

maven { url 'https://qapm-maven.pkg.coding.net/repository/qapm_sdk/android

}

}
```

注意:

如果您的 gradle 版本低于 7.0, 请在 project 的 build.gradle 添加 maven 仓库源, 如下所示:



2. 在 project 的 build.gradle 文件下增加插件的依赖。



QAPMInject ~/Desktop/workspace	Gradle files have changed since last project sync. A project s
> 🗀 .gradle	1 // Top-level build file where you can add con
> 🗀 .idea	
> 🕞 app	3 Vbuildscript {
> 🗀 gradle	4
Ø .gitignore	5 🗸 dependencies {
සි build.gradle	6 classpath <u>'com.tencent.qapmplugin:qap</u>
ତ୍ତ gradle.properties	7 }
🔄 gradlew	8 }
\equiv gradlew.bat	
local.properties	10 🗸 plugins ┨
R ²⁷ settings gradle	11 alias(libs.plugins.androidApplication) ap
	12 alias(libs.plugins.jetbrainsKotlinAndroid
	13 }
Scratches and Consoles	





```
buildscript {
    ...
    dependencies {
        ...
        // 加入下面的内容
        classpath 'com.tencent.qapmplugin:qapm-plugin:3.1'
        ...
        }
    }
}
注意:
```



如果您的 gradle 版本小于 7.4,则将插件版本改为 2.39。

3. 在 app 的 build.gradle 文件下增加以下代码:

Project 🗸	⊕ ≎ × : –	M+ READ	ME.md	<i>©</i> ² build.gradle (QAPMDemo)	≡ proguard-rules.pro
QAPMDemoBelow [QAP	PMDemo] ~/Desktop/work	🚯 Gradi	le files have o	changed since last project sync. A	A project sync may be necessa
> 🗀 .gradle		1 0	lugins {		
🗸 📴 app		2	id('com	android annlication')	
> 🗀 build			id('gan	m-nlugin')	
> 🗈 src		4 }			
⊘.gitignore					
ස? build.gradle		6 p	reBuild.de	pendsOn(UUIDGenerator) //生	E成UUID,用于符号表的唯一标识
≡ proguard-rules.p	ro	7	.		
> 🗈 gradle		8 ⇒ a	ndroid 🚛		
Ø build.oradle					
lo gradle properties		36 d	ependencie	s {	
El gradiew			impleme	ntation fileTree(dir: "lib	s", include: ["*.jar"])
= gradiew bat			impleme	ntation <u>'androidx.appcompa</u>	t:appcompat:1.2.0'
gradiew.bat			impleme	ntation 'androidx.constrai	ntlayout:constraintlayou
local.properties			testImp	lementation 'junit:junit:4	.12'
apm.properties			android	TestImplementation <u>'androi</u>	dx.test.ext:junit:1.1.2'
M+ README.md			android	TestImplementation 'androi	dx.test.espresso:espress
ଯି settings.gradle			impleme	ntation 'com.tencent.qapm:	qapmsdk:5.4.6-pub'
> ① External Libraries					
Scratches and Console	S	45 }			





```
plugins {
    ...
    id('qapm-plugin') //添加插件
    ...
}
...
preBuild.dependsOn(UUIDGenerator) //生成唯一标识,用于后续堆栈翻译
...
dependencies {
```



}

```
...
//添加qapmsdk依赖
implementation 'com.tencent.qapm:qapmsdk:5.4.6-pub'
...
```

4. 请通过以下内容检查是否需要执行此步骤。

请在 Application 所在的类中输入 attachBaseContext, 检查是否有这个的重写方法, 如有重写方法则忽略该步骤, 如 没有请执行下一步。



请将 Application 的包名路径添加进以下配置,如下所示:







```
QAPMPluginConfig {
    // 可选,默认为空,请在Application所在的类中输入attachBaseContext,看有没有这个的重写方法,如
    // tinkerApplication = 'com/tencent/qapm/demoApplication'
}
```

5. 此时您可以尝试进行编译。编译相关的 FAQ 可以查看此步骤。

Q1:如果编译时出现"feature is disabled"的错误,如下:



android} → composeOptions}	
d Sync × Build Output × Build Analyzer ×	
 Build QAPMInject: failed At 2024/5/14, 20:10 with 1 error 221 ms Download info 	Cause: defaultConfig contains custom BuildConfig fields, but to To enable the feature, add the following to your module-level
Cause: defaultConfig contains custom BuildConfig fields, but the feature	android.buildFeatures.buildConfig true

A1: 这是因为插件需要动态在 BuildConfig 插入属性,请在 app 模块的 build.gradle 文件下增加以下代码。







```
android {

....

// 加入下面内容

buildFeatures {

....

buildConfig true

}

....

}
```

Q2: 如果编译时出现 "类冲突" 的错误, 如下:



Sync ×	Build Output × Build	d Analyzer 🛛 🖂	
 Build Q Dow A API O :app A :app Dup 	IAPMinjec' 2 sec, 408 ms vnload info 'variantOutput.getProcess oxcheckDebugDuplicateCla oxdexBuilderDebug 86 ms slicate class found	Duplicate class Duplicate class Duplicate class Duplicate class Duplicate class Duplicate class Duplicate class Duplicate class Duplicate class	android.support.v4.app.INotificationSideChannel found in modules core-1.13.1-runtime (androidx.core: android.support.v4.app.INotificationSideChannel\$Stub found in modules core-1.13.1-runtime (androidx. android.support.v4.app.INotificationSideChannel\$Stub\$Proxy found in modules core-1.13.1-runtime (androidx. android.support.v4.os.IResultReceiver found in modules core-1.13.1-runtime (androidx.core:core:1.13.; android.support.v4.os.IResultReceiver\$Stub found in modules core-1.13.1-runtime (androidx.core:core:: android.support.v4.os.IResultReceiver\$Stub\$Proxy found in modules core-1.13.1-runtime (androidx.core:core:: android.support.v4.os.ResultReceiver\$Stub\$Proxy found in modules core-1.13.1-runtime (androidx.core:core:1.13.1] android.support.v4.os.ResultReceiver\$1 found in modules core-1.13.1-runtime (androidx.core:core:1.13.1] android.support.v4.os.ResultReceiver\$I found in modules core-1.13.1-runtime (androidx.core:core:1.13.1] android.support.v4.os.ResultReceiver\$MyResultReceiver found in modules core-1.13.1-runtime (androidx.core:core:1.13.2]

A2: 这是由于 qapm 与您的工程使用了 android 不同的 support 库,这里请在依赖的时候移除掉qapm 的 android.support 库,如下:







```
dependencies {
    ...
    implementation ('com.tencent.qapm:qapmsdk:5.4.6-pub') {
        // 加入下面内容
        exclude group: 'com.android.support'
    }
    ...
}
```

```
注意:
```



如果未使用 qapm-plugin 插件,则会影响启动、网络的监控。

步骤二:参数配置

1. 在 Android Manifest.xml 中添加以下权限。



<!--上报信息所需--> <uses-permission android:name="android.permission.INTERNET" /> <!--采集信息所需--> <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" /> <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />



2. 为了避免混淆 SDK, 在 App 的 proguard-rules.pro 文件中增加以下配置:



```
-keep class com.tencent.qapmsdk.**{*;}
# 如需要网络监控, 请确保okhttp3不被混淆
-keep class okhttp3.**{*;}
```

步骤三:初始化 SDK

1. 登录 腾讯云可观测平台 控制台,在终端性能监控页面,选择 应用管理 > 应用设置 后,获取 Appkey(上报 ID)。



应	用管理					
Л	业务系统 应用 i	设置 白名单	管理			
Л	业务系统:rum-oPNsq1	Dv.腾讯云监控团队	▼ 应用接入			
	应用名			上报 id	应用 ID	类型
	云监控 andriod dem	0		440**1ee-2574I		安卓
	云监控 iOS demo			e50**570-80311⊡		iOS
	共 2 条					

2. 拷贝下面代码,并修改其中部分字段。下列项均是必需的接口设置,其余接口配置请参考初始化的接口分析(建议在 Application 中初始化 QAPM)。





// 设置手机型号和设备ID。

// 需要传入设备的标识,任意字符串。deviceId(必需!!)

// deviceId可以用来开启白名单,避免数据被抽样上报(崩溃和启动以外的数据抽样率为0.1%)

QAPM.setProperty(QAPM.PropertyKeyDeviceId, "设备的标识");

// 需要传入手机型号(必需!!)

QAPM.setProperty(QAPM.PropertyKeyModel, "填写手机型号");

// 设置Application (必需)

QAPM.setProperty(QAPM.PropertyKeyAppInstance, getApplication());
// 设置AppKey(必需,用于区分上报的产品,该值由终端性能监控的产品配置页面获取,可参考上一步骤)
QAPM.setProperty(QAPM.PropertyKeyAppId, "YourAppKey");



// 设置产品版本,用于后台检索字段(必需) QAPM.setProperty(QAPM.PropertyKeyAppVersion, "YourApp Version"); // 设置UUID,用于拉取被混淆堆栈的mapping (必需,若使用了QAPM符号表上传插件,可以直接使用该变量) // 如有使用qapmplugin插件,则该变量会在build时生成,报错信息可不用关注。如无使用则需要自行传入, QAPM.setProperty(QAPM.PropertyKeySymbolId, BuildConfig.QAPM_UUID); // 设置用户ID,任意字符串,用于后台检索字段(必需) // userId可以用来开启白名单,避免数据被抽样上报(崩溃和启动以外的数据抽样率为0.1%) QAPM.setProperty(QAPM.PropertyKeyUserId, "123456"); // 设置Log等级,(可选),线上版本请设置成QAPM.LevelOff或者 QAPM.LevelWarn QAPM.setProperty(QAPM.PropertyKeyLogLevel, QAPM.LevelInfo); // 设置QAPM的外网上报域名(必需)。国内站:https://app.rumt-zh.com 国际站:https://app.rum QAPM.setProperty(QAPM.PropertyKeyHost,"https://app.rumt-zh.com"); QAPM.setProperty(QAPM.PropertyKeyHost,"https://app.rumt-sg.com"); // 启动QAPM QAPM.beginScene(QAPM.SCENE_ALL, QAPM.ModeStable);

说明:

AppKey 可参考步骤三-步骤1,在终端性能监控 > 应用管理 > 应用设置页面获取。

崩溃和启动数据是全量上报,其他数据因为数据数目过多,采取抽样上报,抽样率为0.1%(千分之一)。如果需要 全量上报,可以开启白名单, App 将会在下次启动时改变抽样率。 可以将设置好 userld 或者 deviceld 通过 应用管理 页面添加白名单里,开启白名单。

多个进程需要各自初始化 QAPM。

步骤四:接入验证

1. 若打印以下日志, 代表该用户未被抽样命中, 需重新设置下抽样率:

-				
Log	cat			
	Xiaomi MI 5X Android 8.1.0, API : 👻 No debuggable processes		Verbose	▼ QAPM
=	logcat			
Ê	्- HttpActivity			* Oresults 🛧 4 🗔 the the Rev 🖬 🏹
± 1	2021-09-23 17:22:08.165 6461-6461/com.example.auto2 W/QAPM_manager 2021-09-23 17:22:08.226 6461-6461/com.example.auto2 W/QAPM_Monitor 2021-09-23 17:22:12.717 6461-6408/com.example.auto2 E/QAPM_manager	_QAPMConfi : type=148 _QAPMLaund	igureWizard 00 audit(0. cher: no ch	d: AppInstance is not android.app.Application. 1.0:93719): avc: denied { read } for name="address" dev="sysfs" ino=40244 scontext thosen to get info

参见 TAG : QAPM_manager_QAPMLauncher

2. 若打印以下日志,则代表初步接入成功,可以验证数据上报/尝试开启高级功能:



参见 TAG : QAPM_manager_QAPMPluginManager



初始化的接口分析

接口名称	参数	参数说明	注意事项	
	key	必填 。需要设置的 Key。		
public static QAPM setProperty(int key, Stringvalue)	QAPM.PropertyKeyLogLevel	选填。开启日志等级 (建议 Debug 版本开 启 QAPM.LevelDebug, release 版本开启 QAPM.LevelWarn)。	-	
QAPM 的相关参 数	QAPM.PropertyNeedTranslate	选填。堆栈是否需要翻译,这里默认是需要翻译的。如果 apk 要翻译的。如果 apk 是没有混淆的需要传入 false,否则前端可能会全部展示为 unTranslated。		
	sceneName	必填 。场景名。		
	mode	必填 。开启的功能。		
public static boolean	QAPM.ModeStable	选填 。开启全部功能 (建议外发版本开 后。包含区间性能、 crash、anr、webview 页面加载、JsError、 网络)。	使用或运算的方式自定义开 后性能模块,如开后 Crash 和 Apr:	
sceneName, int mode)	QAPM.ModeWebView	选填 。开启 WebView 页面加载监控。	beginScene("Crash&ANR", QAPM.ModeCrash	
作用:升启监控	QAPM.ModeJsError	选填 。开启 WebView 的 JS 异常监控。	QAPM.ModeANR)	
	QAPM.ModeHTTPInWeb	选填 。开启 WebView 的网络监控。		
	QAPM.ModeHTTP	选填 。开启网络监 控。		
public static	sceneName	必填 。需要关掉的场	-	



boolean endScene(String		景名(与 beginScene 的要相对应)。
mode) 作用:结束监控	QAPM.ModeDropFrame	选填 。关闭掉帧监 控。
(只针对掉帧和区 间性能采集有效)	QAPM.ModeResource	选填。关闭区间性能 监控。

其他问题

说明:

通过 qapm 插件编译打包 App 时, App 需要一个 uuid 作为构建 id,如果项目目录下存在 qapm.properties 文件,并 且文件里 qapm_uuid 属性的值存在,该值将被作为构建 id,否则插件会随机生成一个构建 id。

qapm-plugin 2.39及之前版本在编译 App 的过程中会报 IO 错误: java.io.FileNotFoundException,

<code>qapm.properties</code> (No such file or directory) $_{\circ}$

all the		_
Call		
jav	a.io. <u>FileNotFoundException</u> Create breakpoint : 👘 👘 👘 👘 👘 👘 👘 👘 👘 👘 👘 👘 👘	ctory
	at java.base/java.io.FileInputStream.open0(<u>Native Method</u>)	
	at java.base/java.io.FileInputStream.open(<u>FileInputStream.java:219</u>)	
	at java.base/java.io.FileInputStream. <init>(FileInputStream.java:157)</init>	
	at com.tencent.qapm.QAPMTransformerTask.loadBuildId(QAPMTransformerTask.java:201)	
	at com.tencent.qapm.QAPMTransformerTask.beforeTransform(QAPMTransformerTask.java:147)	
	at com.tencent.qapm.QAPMTransformerTask.transform(QAPMTransformerTask.java:91)	
	at com.android.build.gradle.internal.pipeline.TransformTask\$2.call(TransformTask.java:281)	
	at com.android.build.gradle.internal.profile.NoOpAnalyticsService.recordBlock(NoOpAnalyticsService.kt:72)	
Ð	at com.android.build.gradle.internal.pipeline.TransformTask.transform(TransformTask.java:239) <61 internal lines>	
Ð	at java.base/java.util.Optional.orElseGet(Optional.java:369) <13 internal lines>	
Ŧ	at java.base/java.util.Optional.orElseGet(Optional.java:369) <49 internal lines>	

该报错仅在编译期间产生,不会影响 App 运行。



功能配置 网络监控

最近更新时间:2024-05-14 12:36:06

开启功能

网络监控需要使用 qapm-plugin 插件进行插桩才可使用,默认会插在网络层的各个出入口。

前提条件

需要在 终端性能监控 > 应用管理 > 白名单 中添加用户白名单或设备白名单,白名单用于初始化 SDK。 在 app 级别的 build.gradle 中配置了 qapm-plugin 插件,参见 集成和初始化。 目前只支持 okhttp3 监控。okhttp3 还依赖 okio 1.14.0 以上版本的库。

配置步骤

在 app 目录的 proguard-rules.pro 文件里增加混淆规则, 防止 okttp3 的代码被混淆。





-keep class com.squareup.okhttp3.**{*;}



	J.	# progoardines secting in borta.gradie
		#
> build		# For more details, see
> 🖿 libs		<pre># http://developer.android.com/quide/</pre>
> 🖿 src		
🛃 .gitignore		
🗬 build.gradle		# If your project uses WebView with JS,
🗧 proguard-rules.pro		# and specify the fully qualified class
> 🖿 gradle		# class:
👩 .gitignore		#-keepclassmembers class fqcn.of.javasc
🗬 build.gradle		# public *;
📊 gradle.properties		#}
🚓 gradlew		
🖆 gradlew.bat	15	# Uncomment this to preserve the line p
local.properties	19	# discomment this to preserve the time h
apm.properties	16	# debugging stack traces.
R settings.gradle		#-keepattributes SourceFile,LineNumberT
> IIIII External Libraries		
Scratches and Consoles		# If you keep the line number informati
		# hide the original source file name.
		#-renamesourcefileattribute SourceFile
		<pre>-keep class com.tencent.gapmsdk.**{*;}</pre>
		<pre>-keep class com.squareup.okhttp3.**{*;}</pre>
		-keep class java.net.**{*;}
Build Variants 🌼 🗘 —		

校验功能是否正常

检索 TAG:QAPM_manager_QAPMPluginManager

每次网络请求后1分钟,如打印以下日志,则代表网络数据上报成功:



检索 TAG: [plugin::142]



Lo	gcat				
	HUAWEI ANG-ANOO Android 10,	No debuggable processes	▼ Verbos	e 🔻	Q- [plugin::142]
Ξ	' logcat				
î	2022-05-18 15:53:46.883 24174-2425	4/com.example.sdkapp I/QAPM_base_J	IsonUploadRunna	bleWith	NewProtocol: [plugin::1

注意:

需要使用 qapm-plugin 插件进行插桩才可用,否则无效。

SDK 只负责抓取网络请求的相关信息,问题数据由后台分析,如慢请求(请求时间大于xxs),网络错误(请求响应码 > 400)。

数据在终端性能监控>网络>慢请求和错误请求列表中查看。

若没有配置正确的白名单, SDK 不会开启网络监控。



WebView、JsError、Web 网络监控

最近更新时间:2024-05-14 12:36:06

开启功能

初始化需要开启 WebView、JsError、Web 网络监控,如下是在 Stable 的基础上开启当前栏目的三个功能。代码如下:





QAPM.beginScene(QAPM.SCENE_ALL, QAPM.ModeStable | QAPM.ModeWebView | QAPM.ModeJsErr

除此之外,还需要配置以下代码:

WebView 监控需要开启与 JavaScript 交互, 在 WebView 初始化时调用如下代码开启:



```
WebSettings webSetting = webView.getSettings();
webSetting.setJavaScriptEnabled(true);
```

在 WebView 初始化完成之后加入 Java 与 JS 之间的调用接口通道,目的是让 JS 层获取到 Java 层的一些配置信息:





webView.addJavascriptInterface(QAPMJavaScriptBridge.getInstance(),"QAPMAndroidJsBri

在 WebView 的 shouldInterceptRequest 代码里加入以下方法,用于拦截 web-sdk 并改用本地 SDK 资源,请确保在 该回调中的最早地方调用以下代码。 如果是 x5 请使用以下代码:





```
@Overridepublic
public WebResourceResponse shouldInterceptRequest(WebView webView, String s) {
    Object response =QAPMJavaScriptBridge.getInstance().shouldInterceptRequestWithX
    if (response != null) {
        return (WebResourceResponse)response;
    }
    return super.shouldInterceptRequest(webView,s);
    }
```

如果是原生 WebView 请使用以下代码:





```
@Overridepublic
public WebResourceResponse shouldInterceptRequest(WebViewwebView, String s) {
    WebResourceResponse response =QAPMJavaScriptBridge.getInstance().shouldIntercep
    if (response != null) {
        return response;
    }
    return super.shouldInterceptRequest(webView, s);
}
```

在 WebView 的 onPageFinished 代码里加入以下方法,用于注入 JS 脚本:





```
webView.setWebViewClient(new WebViewClient(){
    @Override
    public void onPageFinished (WebView view, String url) {
        super.onPageFinished(view, url);
        QAPMJavaScriptBridge.getInstance().initFileJS(view);
    }
});
```



校验功能是否正常

原生 WebView、JsError 监控: 1. 代码中加入以下代码(用于远程调试)。



WebView.setWebContentsDebuggingEnabled(true);

2. 打开谷歌浏览器,地址栏输入 chrome://inspect ,在出现的设备中单击 inspect。



 O O O O Inspect with Chrome Develope: × + ← → C O Chrome chrome://inspect,#devices 	
DevTools	Devices
Devices Pages Extensions Apps Shared workers Service workers Other	 Discover USB devices Port forwarding Discover network targets Configure Open dedicated DevTools for Node Remote Target #LOCALHOST MI 5X #B7359B950604 WebView in com.example.sdkapp (71.0.3578.99) trace
	<pre>@sentry/browser SDK examples file:///android_asset/monitor.html at (0, 1063) size 1080 × 693 pause @sentry/browser SDK examples file:///android_asset/monitor.html at (0, 371) size 1080 × 692 inspect pause</pre>

3. 进入后找到 Console 模块查询日志,如出现 web start success , vxxx ,则代表 WebSDK 注入成功。


	DevTools - file:///android_asset/a.html
← → C file:///andro	i 🕞 📊 Sources Elements Application Lighthouse Console Security Network Performance Redux Memory
	Filter I Hide data ORLS All Cars US CSS inig Media Font Doc WS Manifest Other As blocked cookes Blocked Reques
	10 ms 20 ms 30 ms 40 ms 50 ms 60 ms 70 ms 80 ms 90 ms
	Recording network activity
	Perform a request or hit % R to record the reload.
	Learn more
	Console What's New Search
	Image: Second
	↓ I = 56 messages
	QAPM, track-js, value is empty, sdk will not set value collectRespBody false
	QAPM, track-js, qapmSetField allowHeaders ► Array(0)
	A 1 werping OAPM, track-is, gapmSetField whsConfig
	<pre> Waining Waining Waining Sinfo Sinfo Comple_ration":1,"upload_cumulative":200,"upload_interval":60000,"upload_type":"json","is_collect_data' ncryption":"true","max_report_count":100} </pre>
	No verbose QAPM, track-js, qapmSetField usrConfig {"sample_ration":1,"max_report_count":1000}
	<pre>QAPM, track-js, qapmSetField webLaunchConfig {"max_report_count":100,"sample_ration":1,"upload_cumulative":200,"upload_interval":60000,"upload_type":' ryption":"false"}</pre>
	<pre>QAPM, track-js, qapmSetField jserrorConfig {"upload_interval":60000,"upload_type":"json","is_encryption":"false","max_report_count":100,"sample_rat: _cumulative":200}</pre>
	<pre>QAPM, track-js, qapmSetField networkConfig {"upload_interval":60000,"upload_type":"json","is_encryption":"false","max_report_count":100,"sample_rat: _cumulative":200}</pre>
	QAPM, track-js, function switch is ⊳Object
	QAPM, track-js, document ready!!
	QAPM, track-js, timing ⊨ Object
	QAPM, track-js, event ⊳ Object
	QAPM, track-js, event ▶ Object
	QAPM, track-js, web sdk start success, SDK_VER: v5.1.13, app_key is 34
	QAPM, track-js, uploadWebLaunchData ▶ Object
	uarm, track-js, customPageState stop

4. 检测各个功能是否上报正常,以 JsError 上报为例,如下:

检索 TAG: [plugin::143]

每次触发 JsError 错误,如打印以下日志,则代表 JsError 数据上报成功。



其余检索 TAG 分别如下:

页面加载:plugin::141(每次 Web 页面加载完成后即会上报)。

网络请求:plugin::154(出现错误网络和慢请求时会上报)。

注意:

1. 如需查看 WebView 监控是否正常需要通过 chrome 等浏览器调试查看。



- 2. 页面加载只有页面加载时长大于3.5s才可在问题个例详情里查看。
- 3. 网络请求只有在网络错误和网络慢时才会上报。



Crash、ANR 监控

最近更新时间:2024-05-14 12:36:06

开启功能

初始化需要开启 Crash、ANR 监控,该监控会默认监控 Crash 和 ANR 信息。



// ModeStable模式默认包含了Crash、ANR监控



QAPM.beginScene(QAPM.SCENE_ALL, QAPM.ModeStable);

QAPM 提供了相关接口,如有额外的需要,可以发生了 Crash 或者 ANR 时,上传用户自定义的日志文件,示例如下:



```
QAPM.setProperty(QAPM.PropertyExtraDataListener, new IExtraDataListener() {
    // 发生ANR时会走这个回调
    @Override
    public List<String> onAnrExtraFileHandler() {
        List<String> files = new ArrayList<>();
        File[] fileArray = new File("xxxx").listFiles(); //xxx处请填写文件夹名称
```



```
for (File file : fileArray) {
            files.add(file.getAbsolutePath());
        }
       return files;
    }
    // 发生Crash时会走这个回调
    @Override
   public List<String> onCrashExtraFileHandler() {
       List<String> files = new ArrayList<>();
       File[] fileArray = new File("xxxx").listFiles();//xxx处请填写文件夹名称
       for (File file : fileArray) {
            files.add(file.getAbsolutePath());
        }
       return files;
    }
});
```

校验功能是否正常

检索 TAG: QAPM_manager_QAPMPluginManager



检索 TAG: QAPM_crash

当发生 Crash、Anr 时,打印如下日志,则代表 QAPM 正常收集了此次异常:



Logcat	
[☐ Emulator Nexus_S_API_31 Andro ▼ com.example.myapplication8 (124 ▼ Verbose ▼ Q+ QAPM_crash	
<pre>tendose in textus_pre_sintut i contexample.myappications (121 i centose i ce cumultash</pre>	llector com.t com.tencent.c llector com.t com.tencent.c llector com.t com.tencent.c llector com.t com.tencent.c llector com.t cile com.tencent.c llector com.t ile com.tencent.c ile com.tencent.c des prior to m.example.mys le - /storage worker endec default Exce
1⁄2 Version Control ▶ Run : III TODO	

检索 TAG: plugin::144

当打印如下日志,则代表 QAPM 将此次异常上报成功,此处举例 JavaCrash 的上报情况:



其他 crash 检索 TAG 分别如下:

ANR: [plugin::140]。

NativeCrash : $[plugin::146]_{\circ}$

说明:

为避免出现卡死的情况,接口回调里的逻辑请尽量简单明了。 上传的文件大小限制为20MB,大于限制则不上传,请选择认为有帮助的日志文件。 Crash可以在移动监控的崩溃页面查看,ANR可在总览页面中查看 ANR 率。





卡顿、帧率监控

最近更新时间:2024-05-14 12:36:06

前提条件

已完成集成与初始化。

功能配置

开启监控

初始化需要开启卡顿监控。卡顿无需埋点,而掉帧率需要额外埋点,建议打点在滑动列表上,如(ListView、GridView、RecyclerView等)。

掉帧率埋点

在每次滑动前调用 QAPM.beginScene("xxx滑动", QAPM.ModeDropFrame); 在滑动结束后调用 QAPM.endScene("xxx滑动", QAPM.ModeDropFrame); 一般可以通过重写滑动组件 onScrollStateChanged 方法来实现,示例如下:





```
@Override
public void onScrollStateChanged(AbsListView view, int scrollState) {
    if (scrollState == AbsListView.OnScrollListener.SCROLL_STATE_IDLE) {
        QAPM.endScene("xxx滑动", QAPM.ModeDropFrame); //xxx滑动名称由您自定义,您可以填写
    } else {
        QAPM.beginScene("xxx滑动", QAPM.ModeDropFrame);//xxx滑动名称由您自定义,您可以填结
    }
}
```

校验功能是否正常

版权所有:腾讯云计算(北京)有限责任公司



检索 TAG: QAPM_dropframe_DropFrameMonitor

滑动结束(调用 endScene)后,打印出以下日志则代表掉帧率数据已经存入了本地数据库:



检索 TAG: [plugin::101]

存在 App 本地数据库的掉帧数据将会上报,打印以下数据,代表上报成功:



检索 TAG: QAPM_looper_LooperPrinter 如打印以下日志,则代表卡顿监控正常:

	.ogc	at	·
		Xiaomi MI 5X Android 8.1.0, API 🔻 No debuggable processes 🔹 Verbose 👻	Q- QAPM_looper_LooperPrinter
	ΞI	ogcat	
Į.	î	2020-06-24 16:39:40.947 6202-6202/com.example.sdkapp I/QAPM_looper_LooperPrinter	main, cost=2003, >>>> Dispatching to Handler (android.os.Handler) {9255f9f} com
		2020-06-24 16:39:43.949 6202-6202/com.example.sdkapp I/QAPM_looper_LooperPrinter	<pre>main, cost=2002, >>>>> Dispatching to Handler (android.os.Handler) {9255f9f} com</pre>
	-	2020-06-24 16:39:46.952 6202-6202/com.example.sdkapp I/QAPM_looper_LooperPrinter	main, cost=2003, >>>>> Dispatching to Handler (android.os.Handler) {9255f9f} com
		2020-06-24 16:39:49.954 6202-6202/com.example.sdkapp I/QAPM_looper_LooperPrinter	main, cost=2002, >>>>> Dispatching to Handler (android.os.Handler) {9255f9f} com
		2020-06-24 16:39:52.957 6202-6202/com.example.sdkapp I/QAPM_looper_LooperPrinter	main, cost=2003, >>>>> Dispatching to Handler (android.os.Handler) {9255f9f} com
		2020-06-24 16:39:55.961 6202-6202/com.example.sdkapp I/QAPM_looper_LooperPrinter	main, cost=2004, >>>>> Dispatching to Handler (android.os.Handler) {9255f9f} com

如打印以下日志,则代表卡顿上报正常:



gcat					
∃ HUAWEI ANG-AN00 Android 10, . 🔻	com.example.sdkapp (29156)	•	Verbose	•	्र [plugin::10
- logcat					
2022-05-20 11:07:43.366 29156-29222	2/com.example.sdkapp I/QAPM_base_	FileUplo	adRunnable	With	NewProtocol: [



启动监控

最近更新时间:2024-05-14 12:36:06

启动监控需要使用 qapm-plugin 插件在编译期间进行插桩操作,默认插桩点为 Application 与 Activity 的各个生命周期。在 App SDK 统计默认的启动耗时为 Application 的 attachBaseContext 到第一个 Activity 的 onResume 结束。

前提条件

在 App 级别的 build.gradle 中配置了 qapm-plugin 插件。

配置步骤

1. 需要手动添加一个 Application 的子类,例如 BaseApplication (名称不作要求,子类可以不用实现任何方法,可以不用添加任何属性)。

2. 在 AndroidManifest.xml 文件的 application 的节点添加 android:name 属性, 属性的 value 为"包名+Application 子 类的类名"。

Y 📑 app	7	<uses-permission android.pe<="" android:name="android.pe</th></tr><tr><td>🗸 🖿 manifests</td><td></td><td><uses-permission android:name=" td=""></uses-permission>
릚 AndroidManifest.xml		<uses-permission android:name="android.pe</td></tr><tr><td>🕆 🖿 java</td><td></td><td></td></tr><tr><td>> b com.example.myapplication8</td><td></td><td><pre>conslication</pre></td></tr><tr><td>> com.example.myapplication8 (androidTest)</td><td></td><td>Capplication</td></tr><tr><td>> com.example.mvapplication8 (test)</td><td></td><td>android:allowBackup=" td="" true"<=""></uses-permission>
> kaiava (generated)		android:dataExtractionRules="@xml/dat
> E assets		android:fullBackupContent="@xml/backu
> 📭 jniLibs	15 🔤	android:icon="@mipmap/ic_launcher"
> Tes		android:label="My Application8"
res (generated)		android:name="com.tot,]
> 🔗 Gradle Scripts		android:networkSecurityConfig="@xml/r
	19 🔳	android:roundIcon="@mipmap/ic_launche
		android:supportsRtl="true"
		android:theme="@style/Theme.MyApplica
		android:usesCleartextTraffic="true"

额外打点

如果想统计启动区间内的某些方法的耗时,则需要额外的打点,示例如下:



<pre>public class BaseApplication extends Application {</pre>
<pre>@Override public void onCreate() { super.onCreate();</pre>
b(); try { Thread.sleep(millis: 2000); Latthe (Intermented Exception a)
e.printStackTrace();
c();
<pre>private void b() {</pre>
QAPM.beginScene(StageConstant.QAPM_APPLAUNCH, extrainfo: "b方法名", QAPM.ModeResource); /** * 业务逻辑
A */
QAPM.endScene(StageConstant.QAPM_APPLAUNCH, extrainfo: "b方法名", QAPM.ModeResource);
<pre>private void c() {</pre>
QAPM.beginScene(StageConstant.QAPM_APPLAUNCH, extraInfo: "c方法名", QAPM.ModeResource);
* 业务逻辑
· */
uarm.enuscene(Stageconstant.uarm_appLauwch, extramo: "Слжа", uarm.modekesource);

参见代码:





QAPM.beginScene(StageConstant.QAPM_APPLAUNCH, "xxx方法名", QAPM.ModeResource); /**业务逻辑*/

QAPM.endScene(StageConstant.QAPM_APPLAUNCH, "xxx方法名", QAPM.ModeResource);

如果想自定义启动的结束点,则需要在第一个 Activity 调用 onResume 的20s内额外打点,示例如下:





参见代码:



/**

* 需要自定义结束点的用户需要在onResume之后的20s内,否则以APM的结束点为准



*/
QAPM.endScene(StageConstant.QAPM_APPLAUNCH, QAPM.ModeResource);

校验功能是否正常

每次启动后20s或者切换到后台,如打印以下日志,则代表启动指标数据上报成功。

检索 TAG: [plugin::114]



注意:

需要使用 qapm-plugin 插件进行插桩才可用,并且需要手动增加一个 Application 的子类。否则无效。 启动总耗时大于2.5s才会上报个例数据。

启动的问题数据可以在终端性能监控 > 启动 > 问题列表查看。

计算方式

冷启动:

应用自设备启动后或系统终止应用后的首次启动。 Android 计算方式:mainActivityOnResume_end-attachBaseContext_start。 iOS 计算方式:第一个页面首帧 UI 上屏时间-应用进程创建时间。

首次启动:

应用安装后第一次启动,是特殊的冷启动。 Android 计算方式:mainActivityOnResume_end - attachBaseContext_start iOS 计算方式:第一个页面首帧 UI 上屏时间 - 应用进程创建时间。

热启动:

在进程及Activity实例还存在(iOS 是在应用在后台且存活)的前提下,应用切后台三分钟后,再次切回前台,定义为热 启动。

Android 计算方式: activityOnResume_end - activityOnRestart_start。



iOS 计算方式: ApplicationDidBecomeActive - ApplicationWillEnterForeground。

启动耗时:

总启动耗时/总启动次数





iOS 应用场景 集成和初始化

最近更新时间:2024-05-20 10:47:31

SDK 集成

手动集成

1. 下载 SDK。

2. 拖拽 QAPM.framework 文件到 Xcode 工程内(请勾选 Copy items if needed 选项)。

3. 在 TARGETS > Build Phases-Link Binary Libraries 添加依赖库:

libc++.dylib (libc++.tbd)

libz.dylib (libz.tbd)

libresolv.tbd

4. 在工程的 Other LinkerFlags 中添加 -Objc 参数。

5. 导入配置文件:

5.3.5之前的版本将 framework 里面的 js_sdk.js 文件导入到工程根目录;

5.3.5及以后的版本将 framework 里面的 QAPMResourceFile.bundle 文件导入到工程根目录。

cocoaPods 集成

在 podfile 文件中增加如下操作,然后执行 pod install 指令:

pod 'QAPM',:source => 'https://github.com/TencentCloud/QAPM-iOS-CocoaPods.git'

注意:

iOS SDK 最低兼容系统版本 iOS 8.0。

Web 端环境配置

登录 腾讯云可观测控制台,在终端性能监控页面,选择应用管理 > 应用设置,进入应用设置后,获取 Appkey(上报 ID)。



应用管理		
业务系统 应用设置 白名单管理		
业务系统: rum-oPNsq1Dv.腾讯云监控团队 👻 应用接入		
应用名	上报 id	应用 ID
云监控 andriod demo	440**1ee-2574I⊡	
云监控 iOS demo	e50**570-80311⊡	
共 2 条		

SDK 初始化

1. 在工程的 AppDelegate.m 文件导入头文件: #import <QAPM/QAPM.h> , 如果是 Swift 工程, 请在对 应 bridging-header.h 中导入。

2. 初始化 QAPM 在工程 AppDelegate.m 的 application:didFinishLaunchingWithOptions 方法中初始 化:









```
void loggerFunc(QAPMLoggerLevel level, const char* log) {
#ifdef RELEASE
    if (level <= QAPMLogLevel_Event) { ///外发版本log
        NSLog(@"%@", [NSString stringWithUTF8String:log]);
    }
#endif
#ifdef GRAY
    if (level <= QAPMLogLevel_Info) { ///灰度和外发版本log
        NSLog(@"%@", [NSString stringWithUTF8String:log]);</pre>
```



```
#endif
#ifdef DEBUG
   if (level <= QAPMLogLevel_Debug) { ///内部版本、灰度和外发版本log
       NSLog(@"%@", [NSString stringWithUTF8String:log]);
   }
#endif
}
- (BOOL) application: (UIApplication *) application didFinishLaunchingWithOptions: (NSD
//为响应工信部工业和信息化部关于进一步提升移动互联网应用服务能力的通知要求,提供该设置用于告知spr是
//可选个人信息包括但不限于以下信息:设备制造商、系统、运营商等等
//该设置默认设置是YES,代表允许可采集,示例代码如下:
[QAPMConfig getInstance].collectOptionalFields = YES;
// 特别说明: 若上述该设置置为NO, 部分信息将不再获取, 可能会影响到前端的搜索、展示等, 请知悉。
/// 设置QAPM 日志输出
NSLog(@"qapm sdk version : %@", [QAPM sdkVersion]);
[QAPM registerLogCallback:loggerFunc];
//上报地址
//国内站:https://app.rumt-zh.com
//国际站:https://app.rumt-sg.com
 [QAPMConfig getInstance].host = @"https://app.rumt-zh.com";
 [QAPMConfig getInstance].host = @"https://app.rumt-sg.com";
[QAPMConfig getInstance].customerAppVersion = @"设置app自定义版本号";
//根据实际情况设置userid 和deviceID
//设备唯一标识deviceID,例如IDFV配合Keychain使用
[QAPMConfig getInstance].userId = @"设置userId";
   [QAPMConfig getInstance].deviceID = @"自定义deviceId";
   /// 启动QAPM
   [QAPM startWithAppKey:@"产品唯一的appKey"];
   return YES;
}
```

注意:

研发流程内,可以将设置的用户 ID /设备 ID 添加成白名单,确保指定设备的功能上报不会被采样影响,可在终端性能监控控制台 > 应用管理 > 白名单管理中,点击白名单配置 > 添加完成操作。

监控功能启用

推荐使用:



在 QAPMModelStableConfig.h 文件中的接口: - (void)setupModelAll; //此接口可开启全部监控功能。

在 QAPMModelStableConfig.h 文件中接口: - (void)setupModelStable;//此接口可开启除了网络监控和用户行为监控之外的所有监控功能,网络监控与用户行为监控按照工信部26号文要求划分为扩展业务功能,您可根据自身需要选择是否开启。

自定义使用:

如果要自定义开启 QAPM 功能(包括卡顿、crash、原生网络、原生用户行为、启动、webview),可

在 QAPMConfig.h 文件中的 enableMonitorTypeOptions 接口,设置组合的枚举值,添加多个参数枚举值 完成一系列功能的开启,代码如下:





```
// 设置QAPM 开启的监控:
[QAPMConfig getInstance].enableMonitorTypeOptions =
    QAPMMonitorTypeBlue |
    QAPMMonitorTypeCrash |
    QAPMMonitorTypeHTTPMonitor |
    QAPMMonitorTypeIUPMonitor |
    QAPMMonitorTypeLaunch |
    QAPMMonitorTypeJSError |
    QAPMMonitorTypeWebViewNetWork |
    QAPMMonitorTypeWebViewIUPMonitor |
    QAPMMonitorTypeWebViewIUPMonitor |
    QAPMMonitorTypeWebMonitor;
```

接口说明:

接口: QAPMConfig.h 类中 @property (nonatomic, assign) QAPMMonitorType enableMonitorTypeOptions;

参数说明: QAPMMonitorType 类型为功能类型,可选值为下述代码所示。

可选项说明

:





///检测卡顿功能 QAPMMonitorTypeBlue /// Crash监控功能 QAPMMonitorTypeCrash ///原生网络监控 QAPMMonitorTypeHTTPMonitor ///原生用户行为监控 QAPMMonitorTypeIUPMonitor /// 启动个例监控功能 QAPMMonitorTypeLaunch /// webview的JS异常监控



QAPMMonitorTypeJSError ///webview的网络 QAPMMonitorTypeWebViewNetWork ///webview 用户行为监控 QAPMMonitorTypeWebViewIUPMonitor /// webview页面性能监控 QAPMMonitorTypeWebMonitor

注意:

1. 使用或运算方式自定义开启所需监控功能,如卡顿: QAPMMonitorTypeBlue 。

 2. 为响应工信部 工业和信息化部关于进一步提升移动互联网应用服务能力的通知 要求,我们依据工信部对性能监控 类 SDK 基础功能的定义,将网络监控与用户行为监控划分为

扩展业务功能

,这意味着为了避免采集网络日志信息、用户操作记录等个人信息,您可以选择性开启这两个功能。操作层面您可 以在自定义性能模块开启配置中避免填

入 QAPMMonitorTypeHTTPMonitor 、 QAPMMonitorTypeWebViewNetWork 两个参数,以关闭网络监控功能;以此类推,您可以在自定义性能模块开启配置中避免填

入 QAPMMonitorTypeIUPMonitor 、 QAPMMonitorTypeWebViewIUPMonitor 两个参数,以关闭用户行为 监控功能,或直接使用 ModelStable 功能开启模式以在确保关闭所有扩展业务功能的情况下自动开启其他推荐功能。

SDK 功能介绍

卡顿及流畅度监控功能

卡顿检测功能

QAPMMoniterType:QAPMMonitorTypeBlue 卡顿检测将在卡顿时,卡顿时间超过200ms阈值则采集堆栈进行立即 上报。

流畅度监控功能

在滑动场景下相关页面进行如下代码的打点即可开始统计流畅度,日志会在下次启动 App 后上报数据。





```
#pragma mark - TableView Delegate
- (void)scrollViewWillBeginDragging:(UIScrollView *)scrollView {
    [QAPMBlueProfile beginTrackingWithStage:NSStringFromClass([self class])];
}
- (void)scrollViewDidEndDragging:(UIScrollView *)scrollView willDecelerate:(BOOL)de
    if(!decelerate){
      [QAPMBlueProfile stopTrackingWithStage:NSStringFromClass([self class])];
    }
}
```



- (void)scrollViewDidEndDecelerating:(UIScrollView *)scrollView {
 [QAPMBlueProfile stopTrackingWithStage:NSStringFromClass([self class])];
}

}

Crash 监控功能

QAPMMonitorTypeCrash Crash 日志会在下次启动 SDK 后上报数据。

注意:

1. 业务方如果使用了第三方 SDK 收集普通崩溃的功能请卸载掉第三方监控软件,避免出现上报堆栈不准的问题。

2. 在 FOOM 与 deadlock 卡死退出后,将在下次启动上报上一次记录的相关堆栈信息。FOOM 在 debug 或者非 App Store 全量上报, App Store 环境下数据会有2%的采样抽样。

启动耗时监控功能

功能说明

使用启动耗时监控功能,可以统计出 App 进程创建时间到 App 第一帧 UI 上屏的时间。

当启动时间超过阈值(默4000ms),则会上报个例详情。个例详情包括启动耗时、自动打点区间、自定义打点区间 和启动过程堆栈。

相关接口





```
@interface QAPMLaunchProfile : NSObject
/**
    设置自定义打点区间开始,该区间需要在启动时间区间内。begin与end的scene需要一致。
    @param scene 场景名
    */
    Correct C
```



0end

代码示例

在工程对应的类里面导入头文件 #import <QAPM/QAPMLaunchProfile.h> 。 在 main 函数进行启动监控组件:



```
int main(int argc, char * argv[]) {
```

```
@autoreleasepool {
```

[QAPMLaunchProfile didEnterMain];



```
return UIApplicationMain(argc, argv, nil, NSStringFromClass([AppDelegate cl
}
}
```

Web 监控功能

说明:

1. 目前 WKWebView 支持 iOS 11及以上版本。

2. 该功能能够监控 Web 网络资源加载耗时、jserror 监控。

功能配置

Web 端配置

注意:

如果使用手动集成的方式,需要将 framework 里面的 js_sdk 以 Add Files to 方式引入到工程中;如果是用的 cocoaPods 集成方式则不需要。如果用到 TMFWebOffline 离线包功能,工程里面的 wkwebview 相关页面的头文件 需要引入#import <TMFQWebView/QBWKWebView.h>,且遵循TMFWebOfflineWebViewControllerProtocol 代理, wkwebview 继承 TMFWkWebView 如下设置:





```
#import <TMFQWebView/QBWKWebView.h>
@interface WKWebviewController ()<WKUIDelegate, WKNavigationDelegate, WKScriptM
{
    TMFWkWebView *wkWebView;
}</pre>
```

```
iOS SDK 配置
```

类文件中添加 #import <QAPM/QAPMLaunchProfile.h> 导入 SDK 头文件。



在 WKWebView 的代理方法 webView:didFinishNavigation:中添加如下代码,以提供 Web 获取 native 相关信息接口。



- (void)webView:(WKWebView *)webView didFinishNavigation:(WKNavigation *)navigation
{
[webView evaluateJavaScript:[QAPMWebViewProfile qapmBaseInfo:@" "] completionHandle
[webView evaluateJavaScript:[QAPMWebViewProfile qapmJsStart] completionHandler:nil
}



用户隐私协议

隐私合规政策:因隐私合规要求,在用户同意隐私合规之前请确保不调用 QAPM 的任何接口,此外 QAPM 仍然需要设备级的唯一标识用于确定设备的唯一性,用于用户指标级的计算。



// 当用户授权后,方可正常初始化QAPM
if (isAgree) { }
//启动耗时函数的第一个打点
// 需要传入设备的唯一标识,如IDFV配合Keychain使用
[QAPMConfig getInstance].deviceID = @"自定义deviceId";



//用户user ID、第三方登录账号,此接口可以多次在代码位置使用 [QAPMConfig getInstance].userId = @"设置userId";

说明:

标识 deviceID 采集方式变更背景:当前监管要求 SDK 不允许直接或间接采集 UDID 等信息,我们只能通过让用户 自行传入标识符的方法去区分不同的设备,有效的降低 crash 率指标数据的失真。

符号表配置

利用已上报的数据个例上传符号表。根据个例页面的构建 ID 找到对应的符号表,可以使用官方 atos 命令翻译个例页 面某行堆栈,确认符号表和翻译正常。

1. 在个例页面中有上传的入口,例如选择卡慢 > 卡慢分析 > 卡慢问题列表,进入详情页面。

「である」である。			
卡提分析			
Rum-3 👻 前線 2024-05-10 11 27 ~ 2024-05-11 11 27 🛅 藍修 时间地源 5分钟和波	▼ 应用版本 请选择 ▼	系统版本 调选择 🔻 场景	请选择 👻 设备
⑦ 卡欄样本上接對以采样率为0.1%,因此问题列表中问题样本裁量与指标统计不动合为正常现象,样本采样率的增加可能涉及资源扩音,如需调整调联系移动监控团队			
+個问题列表 全部问题状态 ▼ 段集ID ▼ 調給入需要担当的设备的 特定函数或文件名 III 能入品数或文件名			
问题阅述	卡愷用户(占比) \$	卡愷次數(占比) *	卡豐耗时(ms) \$
ID: b1f9525634afb6f45f851e177a1db4e3 java lang. Thread elego (Thread, java) com lanceart, thr damo, performance, activity.DropFrameActivity\$1.run (SourceFile) android.os.j	1(33.33%)	19(34.55%)	2000
ID: 621b8ae85a2656ecaae8bdb3a168a631 web.bea (Source/in) web.b () web.b	1(33.33%)	18(32.73%)	629
ID: 849a1c2e93db3eb7fcfd6d38bc69827d uit SourceFile) com.tencent.tmf.demo.qu	1(33.33%)	18(32.73%)	231
共3条			

2. 在详情页面中,单击上传 dSYM 文件。



卡慢问题详情												
98035433816e16c9ed451d053c163d8f												
UnTranslated												
newCubestation/500277.cubestation竞赛版 🔻 前移	2023-12-01 19:23 ~ 2024-01-09 19:2	3 📋 后移	时间粒度 天粒度		▼ 应	2用版本 请选	择		系统制	这本 请选择		*
及备ID 用户ID												
卡慢问题分析												
样本分析 统计分析												
样本列表 ⑦ 请选择 ▼	上下文信息											
後年10: E4EDPCC9 AEEC 4DP7 9742 7E9D9PEDC77D	用户ID	设备ID		卡慢耗时	ব	10	如用版本		3	系统版本		页面
度面に 3年25223年282-4252-373-773586752775 用户ID: 10000	10000	54FDBCC8	3-AE6C-4DB7-9713-	201 ms		1.	.0		`	/ersion16.6.1(B	uild20G81)	unava
卡慢耗时:201 ms												and have of a
上报时间: 2023-12-29 15:51:52	设备名称	CPU架构		上报时间	0.00.45:54:50	3	2生时间	54.50	;	⊪否Root		間は次
应用版本· LU	iPad12,1	arm64e		2023-12	2-29 15:51:52	2	023-12-29 15	51:52	î	â		已翻译
设备ID: 54FDBCC8-AE6C-4DB7-9713-7F9D8BFDC77D	构建ID	SDK版本										
用户ID: 10000	C8CCACF4-D504-35A8-B829	J 5.3.1										
下便耗时。286 ms 上报时间:2023-12-29 15:51:52												
应用版本: 1.0	错误信息											
	火焰图 堆栈信息											
していていた。 していたのでは、 していたのでいたので、 していたので、 していたので、 していたので、 していたので、 していたので、 していたので、 していたの	10 ms 20 r	ms 30 ms	40 ms	50 ms	60 ms	70 ms	80 ms	90 ms	100 ms	110 ms	120 ms	130 ms
卡慢耗时:286 ms												
上报时间: 2023-12-29 15:51:52							_					
应用版本: 1.0												
设备ID: 54FDBCC8-AE6C-4DB7-9713-7F9D8BFDC77D	10 ms 20 m	ns 30 ms	40 ms	50 ms	60 ms	70 ms	80 ms	90 ms	100 ms	110 ms	120 ms	130 ms
用户ID: 10000	root CubeCompetition 0x104edc000	0x104f2e0d0										
卡慢耗时: 375 ms	UlKitCore 0x1aa262000 0x1aa5f	ebb0										
	1 11/10 0.40000000.0.454	150										

3. 进入符号表上传页面,单击选择文件,然后选择该构建对应的 dSYM 文件即可。

上传dSYM	文件		
注意: ● 请将师 * 从系统 * 需要确 * 需要确	5缀为 .dSYM 的符号表文件直接压缩 上获取的dSYM文件的默认名称为X 保zip文件里面直接就是一个或多个 保XXX.app.dSYM文件夹里面的层线	a成 zip 文件进行上传,如有多 XX.app.dSYM,需要确保不要 XXX.app.dSYM,不能额外增 &是默认层级: Contents/Res	?个需要打包在一起统一上传 修改默认名称 加文件夹层级,可以有多个 .dSYM文 ources/DWARF/XXX
构建ID		文件名称	状态
C8CCACF	4-D504-35A8-B829-3E2BB1DE5	723 选择文件	


查看 QAPM 工作日志

设置查看工作日志

在调用 [QAPM startWithAppKey:] 启动 QAPM SDK 前,设置日志输出函数,可以根据不同发布版本情况进行输出日志控制:



```
void loggerFunc(QAPMLoggerLevel level, const char* log) {
#ifdef RELEASE
    if (level <= QAPMLogLevel_Event) { ///外发版本log</pre>
```



```
NSLog(@"%@", [NSString stringWithUTF8String:log]);
    }
#endif
#ifdef GRAY
   if (level <= QAPMLogLevel_Info) { ///灰度和外发版本log
       NSLog(@"%@", [NSString stringWithUTF8String:log]);
    }
#endif
#ifdef DEBUG
   if (level <= QAPMLogLevel_Debug) { ///内部版本、灰度和外发版本log
       NSLog(@"%@", [NSString stringWithUTF8String:log]);
    }
#endif
}
- (BOOL) application: (UIApplication *) application didFinishLaunchingWithOptions: (NSD
   /// 设置OAPM 日志输出
    [QAPM registerLogCallback:loggerFunc];
   /// ...
   /// 设置启动QAPM SDK
}
```

上报日志分析

在接入完成 SDK 后,通常情况下会通过分析日志来确定监控功能是否已经开启。 监控功能未开启时,日志如下:



[QAPM_LogEvent][QAPM.m:327][Config]	[Custom lag monitoring function] The background is not allowed to be enabled
[QAPM_LogEvent][QAPM.m:331][Config]	[Frame drop rate monitoring function] is not allowed to be enabled in the background
[QAPM_LogEvent][QAPM.m:335][Config]	[Blue (lag monitoring) function] is not allowed to be enabled in the background
[QAPM_LogEvent][QAPM.m:339][Config]	[Yellow(VC leak detection function)] is not allowed to be enabled in the background
[QAPM_LogEvent][QAPM.m:343][Config]	[Foom monitoring function] is not allowed to be enabled in the background
[QAPM_LogEvent][QAPM.m:347][Config]	[Deadlock monitoring function] is not allowed to be enabled in the background
[QAPM_LogEvent][QAPM.m:351][Config]	[Crash indicator] is not allowed to be enabled in the background
[QAPM_LogEvent][QAPM.m:355][Config]	[QQLeak memory leak detection function] is not allowed to be enabled in the backgroup
[QAPM_LogEvent][QAPM.m:359][Config]	[Resource usage monitoring function] is not allowed to be enabled in the background
[QAPM_LogEvent][QAPM.m:363][Config]	[Memory maximum usage value monitoring (peak rate function)] is not allowed in the
[QAPM_LogEvent][QAPM.m:368][Config]	[Chunk malloc monitoring function] is not allowed to be enabled in the background
[QAPM_LogEvent][QAPM.m:373][Config]	[NormalCrash monitoring function] is not allowed to be enabled in the background
[QAPM_LogEvent][QAPM.m:377][Config]	[User behavior monitoring AthenaSDK function] is not allowed to be enabled in the
[QAPM_LogEvent][QAPM.m:381][Config]	[Power consumption monitoring] is not allowed to be enabled in the background
[QAPM_LogEvent][QAPM.m:385][Config]	[Network monitoring] background is not allowed to be enabled
[QAPM_LogEvent][QAPM.m:389][Config]	[JSerror] background is not allowed to be enabled
[QAPM_LogEvent][QAPM.m:393][Config]	[WebView Slow Request] background is not allowed to be enabled
[QAPM_LogEvent][QAPM.m:396][Config]	[Launch monitoring function] is not allowed to be enabled in the background

监控功能开启时, 日志如下:

[QAPM_LogEvent][QAPM.m:179][Config]	[Normal crash monitoring function] is enabled
[QAPM_LogEvent][QAPM.m:226][Config]	[User behavior monitoring AthenaSDK function] is enabled
[QAPM_LogDebug][QAPMEventConn.m:110]][Athena] appid:1498,userid:qapmcharles,buildid:unknown,version:5.3.1-qapm-sdk-debug
[QAPM_LogDebug][QAPMEventConn.m:133]][Athena] The current performance event reporting logic is real-time JSON reporting, and the repo
[QAPM_LogEvent][QAPM.m:229][Config]	[ubs monitoring function] is enabled
[QAPM_LogEvent][QAPM.m:277][Config]	[Blue (lag monitoring) monitoring function] is enabled
[QAPM_LogEvent][QAPM.m:284][Config]	[HTTP monitoring function] is enabled
[QAPM_LogEvent][QAPM.m:292][Config]	[WebMonitor] background configuration is allowed to be enabled, please click on the corresponding

通过初始化日志,可以看到初始化成功,各个监控功能开启,然后就是各功能上报成功的验证。 启动耗时的上报



卡顿个例的上报



[QAPM_LogDebug][QAPMReportCenter.m:264][UploadReport] [<mark>Plugin:2</mark>] QAPM upload file success r	esponse data is
[{"entrance_id":"f856cbd0-0caa-4276-888b-aa1053d12ddd","id":"BA423BB9-A8B0-49F3-A320-5999F0	E20E1B","status":1000,"data":"may be ok."}],h
uploadURL:https://ten.sngapm.qq.com/entrance/v3/uploadData/file/?plugin=2&app_id=1498&data_	num=1
[QAPM_LogInfo][QAPMReportCenter.m:265][UploadReport] [Plugin:2] QAPM upload file success	
[QAPM_LogDebug][QAPMReportCenter.m:280][UploadReport] uploadJson:{	
"meta" : {	
"app_id" :	
"category" : "PERF_LAG",	

FOOM个例上报



Deadlock个例上报



HTTP 监控上报



普通崩溃(normal crash)的上报

在触发 normal crash 的上报时,请不要将数据线连接 Xcode,触发完 normal crash 后,下次重启 App 的时候即可看 到上报信息,该上报日志可通过 Mac 自带的控制台查看上报日志,日志如下:

[QAPM_LogDebug][QAPMReportCenter.m:264][UploadReport] [Plugin:46] QAPM upload file success,response data is [{"entrance_id":"55643c4a-be25-43f3-b01e-bdd873f770b9","id":"4F0CB2EC-5647-4AB7-A6E3-050076337E82","status":1000,"data":"may be ok."}],httpRo uploadURL:https://ten.sngapm.qq.com/entrance/v3/uploadData/file/?plugin=46&app_id=1498&data_num=1 [QAPM_LogInfo][QAPMReportCenter.m:265][UploadReport] [Plugin<mark>:46</mark>] QAPM upload file success

Webview 和 JSerror 的上报

Webview 和 jserror 的上报,可在 xcode 查看日志,以 plugin:43 和 plugin:41 为准。



[QAPM_LogDebug][QAPMReportCenter.m:264][UploadReport] [Plugin:41] QAPM upload file success,response data is [{"entrance_id":"5426ebae-0de3-49c6-99ad-83d30288a9aa","id":"fdb5fd45-a1ec-42d5-b71b-6d42dbfd7031","status":1000,"data":"may be ok."}],httpi uploadURL:https://ten.sngapm.qq.com/entrance/v3/uploadData/json/?plugin=41&app_id=1498&data_num=1 [QAPM_LogInfo][QAPMReportCenter.m:265][UploadReport] [Plugin:41] QAPM upload file success

说明:

更多高级功能配置请参见 GitHub - TencentCloud/qapm-sdk-ios: 主要用来更新 qapm 性能监控组件,以及 shell 脚本 及相关文档文件夹中的文档。



控制台操作指南 崩溃

最近更新时间:2024-05-13 18:03:25

终端性能监控通过对崩溃问题个例提取关键特征进行聚合,便于您针对 App 崩溃的根因分析。

功能入口

1. 登录 腾讯云可观测平台。

2. 在左侧导航栏中选择终端性能监控 > 崩溃,选择需要查看的业务系统、应用、时间范围分析崩溃问题。



多维分析

多维分析基于应用版本、系统版本、崩溃类型、设备类型、应用状态等多个维度分析关键指标,便于您聚焦的现象进行针对性的崩溃根因分析。



多维分析		
崩溃穴数		
崩溃率		
追注页次数	柱状图 列表 系统版本	
SDK启动次数 4.2.4	1.44 % 345次	
4.8.8	1.41% 338次 12	
4.2.8	1.40 % 33627 7	
4.7.7	1.38 % 33227 6	
4.6.5	1.36 % 327% 11	
-		
崩溃类型	社 状面 列表 设备类型	
Java	94.06 % 22.6KX vivo X9L	
Native	5.94 % 1.43K0/t Newig SX	

崩溃问题列表

崩溃问题列表展示了所有设备的崩溃问题。您可以根据问题异常类型、问题设备 ID、特定函数或文件名快速筛选相关崩溃问题。您还可以单击问题概述查看崩溃问题详情,定位分析崩溃根因。

崩溃问题列表	全部件算法型 ▼ 役級D ▼ 資金人業要要要的设备0 特徴機構成文件名① 第金人品的成文件名		
	向踢艇送	崩溃用户(占比) \$	崩溃次数(占比) \$
	ID: : jeva lang (system load).biter (System load). com tencer, till com tencer, till como percentance accung conservations per concurs pour certary and col view Vergentinon (Les(Vergentino)).	1 (20%)	3555 (20.04%)
	ID: javalang.OutOffMemoryError con tenent third famo asforty (CrashActivity (SourceFile) con tenent third familie 432 and is (SourceFile) and roid View View PerformClick (View pava)	1 (20%)	3549 (20.01%)
	ID: native strash ang_stocess2() lifestate lifestate aci) con tencent lonf demmpb/fointer/Test/Native Method)	1 (20%)	3548 (20.00%)
	ID: java Jang MullPointerException cont secent this in the second this of surgeFile) cont secent this in the second se	1 (20%)	3543 (19.88%)
	D: Journal Content of	1 (20%)	3541 (19.97%)

指标说明

相关指标说明如下表所示:

指标名称	指标说明
崩溃率	指定时间范围内崩溃发生次数/App 启动次数



崩溃用户率	指定时间范围内受到崩溃影响的用户数/启动 App 的用户数
崩溃次数	指定时间范围内崩溃发生次数
崩溃用户数	指定时间范围内受到崩溃影响的用户数
崩溃类型	按照崩溃问题发生位置将崩溃类型分类为 Java 崩溃与 Native 崩溃
SDK 启动次数	应用启动次数



ANR

最近更新时间:2024-05-13 18:03:25

终端性能监控通过对 ANR 问题个例提取关键特征进行聚合,便于您针对 App 的 ANR 问题进行根因分析。

功能入口

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中单击终端性能监控 > ANR。
- 3. 选择需要查看的业务系统、应用、时间范围等分析 ANR 问题。



多维分析

多维分析基于 ANR 次数、启动次数、 ANR 率多个维度分析关键指标,便于您聚焦现象进行针对性的 ANR 情况根因分析。



多维分析	
AMR次截 w	
ANREXE	
SDK曲构次数 柱状面 列表	系统版本
	6
462 155 % 12187	10
461 150% I 212 次	13
458 145% [041次	8
4.9.4 1.43 % 202 %	9
设备类型	应用状态
M 5X 33.63 % 4.74K %	前台

ANR 问题列表

ANR 问题列表页提供针对聚类问题的展示、搜索、排序、管理功能,并可通过单击查看详情按钮下钻到问题详情页。

ANR ANR分析 ANR例题列表		
Rum-test500053 num-android-demo + 1896 2024.04.27 15 39 - 2024.04.28 15 39 👸 副修 1999近年 50時紀章 + 成用版本 国用版本	▼ 第四部 ▼ 頁面 目前第	▼ 设备类型 薄选择
ANR问题列表 全部问题状态 • 资金0 • 清全人高生活生的公会0 特益活動成文件名(清全人高快成文件名		
向跟随这	ANR明户(占比) \$	ANR次数(占比) \$
D: jeva.lang. Thread.elegi: Thread.jeva) constance: thrif. damo.performance.AnrActivity.al(SourceFile) constance: thrif. damo.performance.AnrActivity.B.co.Clack(SourceFile) andred: threat.VerapredTocClaCt/Verbus(SourceFile)	1 (100%)	4702 (100%)
共 1 条		

ANR 问题详情

问题详情页提供针对某一类问题的多维统计分析及针对各问题个例的分析功能,您可以单击对应的问题描述进入问题详情。



← ANR问题详情							
27b59a7f875af29ab8808d4d97d6053e java lang. Triread sleep(Thread java) com tencent and demo performance. AnrActivity a(SourceFile) com tencent and demo performance. AnrActivity 1 cnClick(SourceFile) android view Vew performClick(Vew java)							
Rum 5ett 550353 rum-android demo v 前路 2024-04 用户D	27 15 39 ~ 2024-04-28 15 39	后移 时间拉度 5分钟粒度	· 成1	用版本 道池挿 👻	系统版本 请选择	▼ 设备类型 请选择	▼ 应用状态 调选择
Anr何题题分析 样本分析 统计分析							
住在外域 ① 用品目	上下文信息 用中ID 123456 佰 系統版本 10 佰	设命ID 2a0d558b9c3ed36b4c201308te8 页面 com.tencent.tmf.module.qapm.per.	ANR美型 后 ANR 后 设备名称 后 WWO X9L 后	异常美型 java lang RuntimeException 而 应用状态 軟給而	异相同因 ANR Input dispatching timed out (CPU문約 arm64-v8a間	Java虚拟机最大可使用内存总量 384 M m 上限时间 2024-04-28 15:39:38 m	Java直照机未使用的内存总量 Ja 2.40 M 行 8. 发生时间 是 2024-04-28 15.39.26 行 音
訳価印:2305 川戸印:123456 上服时间:2024-04-2815.39:14 欧相版本:4.2.7	APM核识 e9273749-4ae6-47a9-beb5-43e9 错误信息	构建口 10 939598bc-3405-494b-8398-0a8a	SDK版本 1 5.3.2-pub-private唱				
92840: 2000 ЛІ/ЧЮ: 123456 1-1893581: 2024-04-28 15:39:02	堆栈信息 间译 全部展开						

指标说明

ANR 相关指标说明如下表所示:

指标名称	指标说明
ANR 率	指定时间范围内应用发生 ANR 设备数 / 总设备数
ANR 次数	指定时间范围内应用发生 ANR 的次数
SDK 启动次数	应用启动次数
ANR 用户率	指定时间范围内受到 ANR 影响的用户数 / 启动应用总用户数
ANR 用户数	指定时间范围内受到 ANR 影响的用户数
用户数	启动应用总用户数



卡慢

最近更新时间:2024-05-20 10:47:31

终端性能监控通过对卡慢问题个例提取关键特征进行聚合,便于您针对 App 的卡慢问题进行根因分析。

功能入口

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中单击终端性能监控 > 卡慢。
- 3. 选择需要查看的业务系统、应用、时间范围等分析卡慢问题。

卡慢													
卡慢分析 卡慢问题	酠表												
Rum	dam^	前移 2024-05-10 11:27	~ 2024-05-11 11:27	后移时间粒度	5分钟粒度	٣	应用版本	请选择	Ŧ	系统版本	请选择	Ψ.	场景 清选择
			流畅度 86.21 %										
指标分析													
<u>趋势分析</u> 5	A位分析 分布分析												
94.14 %		•											
78.45 %													
62.76 %									-				
47.07 %													
31.38 %									-				
15.69 %													
0 %													
		05/11	11:05						05/11 1	1:10			

指标说明

卡慢相关指标说明如下表所示:

指标名称	指标说明
流畅度	流畅页面两帧之间的间隔时间 16.67ms, 1000ms(1秒)/16.67ms=60FPS, 60FPS 就是理想的流畅的页面。 仅显示占比大于0.1%的数据
场景触发次数	卡慢场景触发次数





启动

最近更新时间:2024-05-20 10:47:31

持启动耗时、慢启动占比等进行启动指标分析,您可通过慢启动问题列表定位分析 App 慢启动根因。

功能入口

1. 登录终端性能监控控制台。

2. 在左侧导航栏中选择启动,选择需要查看的业务系统、应用、时间范围分析启动问题。

多维分析

多维分析基于应用版本、系统版本、设备类型、地区、运营商等多个维度分析关键指标,便于您聚焦的现象进行针 对性的慢启动根因分析。

多维分析		
启动耗时	柱状图 列表	系统版本
4.6.5	3 次(20 %) 2461 ms	
4.5.8	3 次(20 %) 3423 ms	10
		9
4.3.4	3 /×(∠u %) ∠946 ms	6
4.4.5	3 次(20 %) 2952 ms	
4.3.1	3 次(20 %) 2506 ms	13
设备类型	柱状图 列表	地区
vivo X9L	9 次(60 %) 2960.33 ms	Unknown
MI 5X	3 次(20 %) 2461 ms	

慢启动问题列表

慢启动问题列表展示了所有设备的慢启动问题。您可以根据问题异常类型、问题设备 ID、特定函数或文件名快速筛选相关慢启动设备。您还可以单击问题概述查看慢启动问题详情,定位分析 App 慢启动根因。

对于每一个应用冷启动/首次启动样本, Android 系统若启动耗时大于2500ms 为慢启动, iOS 系统若启动耗时大于 4000ms 为慢启动对于每一个应用热启动样本, Android 系统若启动耗时大于 1000ms 为慢启动, iOS 系统若启动耗时大于2000ms 为慢启动仅慢启动样本会显示在问题列表中。



④ 機会动林本上超對认采样率为0.1%,因此问题对表中问题样本教集与指标统计不确合为正常现象,样本采样率的增加可能涉及过源扩音,如需调整编联系移动组验团队						
僧信动问题列表 ⑦ 全部问题状态 ▼ 设备D ▼ 调泡入需要控系的设备D 特定通数成文件名⑦ 调泡入高数近文件名						
问题规述	(提启动用户数(占比) *	個启动次数(占比) ◆	启动耗时(ms) \$			
ID: 9b4b08fd872a15aac11996022d95037f com tensent firmf.etemo. Tim/DelegaleApplication.int/StatServicefSourceFile) comi com.tencent.timter.antry. Interreputicationummers into annuta voice and the second second second second second	1(50%)	20(66.67%)	3848.80			
ID: 542aa415869e7dbd3f8b870088028e07	1(50%)	10(33.33%)	2893			
共 2 条						

指标说明

相关指标说明如下表所示:

指标名称	指标说明
启动次数	应用启动次数。
启动耗时	启动应用耗时。
平均耗时	线上样本启动耗时之和/应用启动次数。
慢启动次数(占比)	发生慢启动的次数/总启动次数。 在 Android 端, 启动耗时超过2.5s 被默认定义为慢启动。 在 iOS 端启动耗时超过4s 被默认定义为慢启动, 阈值可自定义。
首次启动	App 安装后的第一次启动,属于特殊的冷启动。
冷启动	App 结束进程,或退出到后台,进程被系统回收后,再次启动的过程。
热启动	App 切换后台3min 后,从后台被唤起,或从其他 App 界面切换回来的过程。



网络

最近更新时间:2024-05-13 18:03:25

通过吞吐量、请求次数、网络响应时间、慢请求占比、HTTP 错误率、网络错误率、TCP 建连时间等指标进行网络问题分析。

功能入口

1. 登录 腾讯云可观测平台。

2. 在左侧导航栏中选择终端性能监控 > 网络, 支持从业务系统、应用、时间范围等多个维度分析网络问题。

多维分析

多维分析基于应用版本、系统版本、域名、URL、设备类型、网络类型、地区和运营商等多个维度分析关键指标, 便于您聚焦的现象进行针对性的慢请求/错误请求根因分析。

慢请求

多维分析		
请求次数		
前宋次数 ▲		
请求耗时		
发送字节数	125/102 2/34C 3700/00C+	
接收字节数	10.94K X(1.33 %) 1500.02 ms 📗 10	
慢请求次数		
慢请求用户数 ▼	10.98K ½(1.33 %) 1503.28 ms 6	
4.9.8	10.81K /2(1.31 %) 1508.89 ms 11	
447	10.7K W(1.20.85) 1 F0/4 77 mr 12	
45,45,7		
_		
4.7.2	10.75K ½(1.30 %) 1500.75 ms	
のな参加	the Table Ta	
设留央坐	社代語 ジャネ 「開始会社	
MI 5X	277.21K XX(33.62 %) 1503.49 ms 26	
Nexus 5X	274.79K /Xt/33.33 %)11502.98 ms Nn@ermiccion	

错误请求



多维分析		
错误请求次数 👻		
翻阅读先次教 请求次教	柱状图 列表	系统版本
4.2.6	169 次 1.58 % ┃	11
4.3.3	166次 1.55%	13
4.1.3	159 次 1.49 %	6
47.6	156次 1.46%	8
4.9.7	155 次 1.45 %	10
设备类型 MISY	<u> は</u> 北面 列表 3.664 / bl 1.34.36 %	网络类型
	8, U.H. A 100.C	

慢请求问题列表

慢请求问题列表展示了所有设备的慢请求问题。您可以根据问题异常类型、问题设备 ID、特定函数或文件名快速筛选相关慢加载设备。您还可以单击问题概述下的相关链接查看慢请求问题详情,定位分析 App 慢请求根因。

网络		
慢请求 個小市場的人類 错误请求 错误请求问题列表 错误请求问题列表		
Rum-lest500353.rum-android-demo 🔹 前修 2024-03-01 17:25 ~ 2024-04-30 17:25 💼 后修 时间检查 天粒度 💌 应用版本 请选择	▼ 系统版本 调选择 ▼	域名 请选择
 ● 利心市 無面前 200 € ● 利心市 200 € ● 利心市 200 € 		
(資格式內醫列表 ⑦ 全部问题状态 ▼		
向题概述	侵请求用户数(占比) \$	────────────────────────────────────
ID: c m.zhipin.com/wapi/zpgeek/mobile/search/jobilst.json	1 (25%)	127016 (57.20%)
ID: a www.m-toy.com.tw/products/eg001	1 (25%)	31677 (14.27%)
ID: 5	1 (25%)	31677 (14.27%)
ID: a b b b b b b b b b b b b b b b b b b	1 (25%)	31677 (14.27%)
± 4 &		

对于每一个 HTTP 请求样本, 传输数据大于 50KB 时, 传输速度小于 10KB/s 为慢请求;若传输数据小于等于 50KB, 响应时间大于 2s 为慢请求, 这些慢请求样本会显示在问题列表中。



慢请求问题分析							
样本分析 统计分析							
样本列表 ⑦ 请选择 ▼	上下文信息						
设备ID: 2 用户ID: 123456	▲ 用户ID 123456 匝	设备ID 2a0d558b9c3ed36b4c201308fe8	URL . The https://m.zhipin.com/wapi/zpge	参救信息 ek/ ┗ city=101280600&querySour	状态码 rce=1 后 200 后	本地DNS服务器地址 - 但	请求 GE
请求耗时:2151ms 上版时间:2024-04-28 17:45:37 成田時本:4.1.8	DNS查问时间 Oms ī ī	TCP握手时间 0ms 后	SSL时间 Oms 临	TTFB 266ms 🛅	响应时间 1876ms 阳	发送 字节 数 1088 匝	據彬 24.5
	系統版本 10)看 发生时间 2024-04-28 17:45:25 后	设备名称 Nexus 5X 币 是否Root 香 币	网络迷型 WIFI 币 APM标识 e72df99f-8dc9-4fb2-b36a-0ac7	客户鏡頭p 11.142.213.27 百 构建D 939598bc-3405-494b-8398-	国家 美国 店 SDK版本 -0a8a 1 5.1.0_jemter15	运营商 未如 心	地区 - 石
22.000000 1010 現合ID: 122456 満球表別: 2151ms 上期時間。2024.04.28 17.45.29 应用版本: 4.4.7	Hagenec: 前広信息 ⑦ ● "headers": { f ficer "cache-control": "no-cac "connection": "kep-alir "content-encoding": "gai	796" 1° 2°					
(2歳日): 2ac4558bac3ad38b4c201308ba8db510 用户ID: 123456 頃永長規則: 2151ms 上部時间: 2204-04-28 17.45.13 成用版本: 4.5.8	"content-type": "applica "date": "Wed, 15 Jun 2002 "transfer-encoding": "dr]	ion/json" 09:10:23 GMT" unkod"					

错误请求问题列表

发生 HTTP 请求错误、DNS 解析错误、无法建连、连接超时等网络方面的错误。将会展示在错误请求列表中,您可以单击**问题概述**下的相关链接,查看错误请求问题详情,定位分析错误请求根因。

网络		
慢滴求 慢滴求问题列表 错误请求 错误请求问题列表		
Rum-test/500353 rum android demo ● 2024-03-01 17 25 一目 后部 时间拍波 天程章 ● 自用版本 通用版本 通用版本	→ 系统版本 请选择 ▼ 域名 请选择	
网络类型 请选择 ▼ 地区 请选择 ▼ 运营等 请选择 ▼ 超势端P 读选择 ▼		
异常请求问题列表 全部问题状态 ▼ 役留D ▼ 请输入需要控系的论备问		
问题继述	错误用户数(占比) \$	1
ID:	1 (33.33%)	
404(清本资源不存在) w-63209-28716-39529.479064108.sites.hk36.qifelys.com/		
	1 (33 33%)	
404(請求资源不存在)	. ()	
m.zhipin.com/wapi/zpgeek/mobile/searchi/oblist.json		
ID: a	1 (33 33%)	
902(网络过接异常)		
ganklo/		
共 3 条		

您还可以单击问题概述查看错误请求问题详情,定位分析错误原因。



错误请求问题分析 样本分析 统计分析						
样本列表 ⑦ 请选择 🔻	上下文信息					
设备D: 2a0d1 101 101 101 101 101 101 101 101 101	用户ID 123456 匝	设备ID 2a0d:08fe8	URL	参数信息 5 - 石	状态码 404 日	本地DNS服务器地址
上报时间: 2024-04-28 17:58:18 应用版本: 4.2.3	应用版本 4.2.3 百	系统版本 11 日	设备名称 MI 5X 后	客户端源p	国家 美国后	网络类型 NoNetwork 后
设备D: 2800000000 00000 用户ID: 123456	上报时间 2024-04-28 17:58:18 后	发生时间 2024-04-28 17:58:06 位	是否Root 否 后	APM标识 081418cd-7eba-4774-b195-e293	构建ID 「自 939598bc-3405-494b-8398-0a8a	SDK版本 后 5.3.2-pub-private后
上报时间: 2024-04-28 17:58:09 应用版本: 4:5.7	错误信息 ⑦					

指标说明

相关指标说明如下表所示:

指标名称	指标说明
请求耗时	应用请求耗时
慢请求比例	慢请求占比为选定时间段内, 慢请求数量与访问量的比值 传输数据大于50KB时, 传输速度小于10KB/s为慢请求 若传输数据小于等于50KB, 响应时间大于2s为慢请求
慢请求次数	慢请求次数为选定时间段内的慢请求数量 传输数据大于50KB时,传输速度小于10KB/s为慢请求 若传输数据小于等于50KB,响应时间大于2s为慢请求
请求次数	应用请求总次数
慢请求用户比例	指定时间范围内受到慢请求影响的用户数与总用户的比值
慢请求用户数	指定时间范围内受到慢请求影响的用户数
请求错误率	请求错误次数/总请求次数
错误请求次数	在选定时间段内,出现网络错误的数量 错误请求指发生 HTTP 请求错误、DNS 解析错误、无法建连、连接超时等网络方面的 错误
错误用户比例	指定时间范围内受到错误请求影响的用户数与总用户的比值
错误用户数	指定时间范围内受到错误请求影响的用户数



Webview

最近更新时间:2024-05-13 18:03:25

通过页面加载耗时、慢加载占比以及 JS 错误率等进行 WebView 指标分析,并通过问题列表对 WebView 以及 JSError 问题进行下钻。

功能入口

1. 登录 终端性能监控控制台。

2. 在左侧导航栏中选择 webview,选择需要查看的业务系统、应用、时间范围分析 Webview 问题。

慢加载、JS 错误多维分析

多维分析基于应用版本、系统版本、设备类型、页面、网络类型、运营商和地区等多个维度分析关键指标,便于您 聚焦的现象进行针对性的慢加载/ js 错误根因分析。

多维分析			
页面加载次数 👻			
页面加载次数 完全加载耗时		柱状圈 列表	系统版本
4.7.7		874 次(1.38 %) 4839 ms	12
4.1.2		874 次(1.38 %) 4839 ms	9
4.6.5		868 次(1.37 %) 4839 ms	11
4.6.2	4.6.5 市面印建定地 - 868 次	870 次(1.37 %) 4839 ms	13
4.8.9	占比: 1.37 % 完全加戰耗时: 4839 ms	866 次(1.36 %) 4839 ms	10
设备类型		柱状间 利表	页面
Nexus 5X		21.62 K 次(34.03 %) 4839 ms	https://www.nasa.gov/
MI 5X		20.99 K ½(133.04 %) 4839 ms	file:///android_asset/monitor.html

慢加载问题列表

慢加载问题列表展示了所有设备的慢加载问题。您可以根据问题异常类型、问题设备 ID 快速筛选相关慢加载设备。 您还可以单击**问题概述**查看慢加载问题详情,定位分析 App 慢加载根因。 说明:



慢加载样本上报默认采样率为0.1%,因此问题列表中问题样本数量与指标统计不吻合为正常现象。

webview		
備加载 <mark>雙加載何聽列表</mark> JS错误 JS错误问题列表		
Rum-test500053.rum-android-demo	1655 · SANGE 1655	▼ 设备类型 漂白泽
① 他加载样本上相载认采样本为01%,因此问题列表中问题样本数量与指标统计不动合为正常现象。样本采样本的增加可能涉及资源扩音,如素调整清载系修动监控团队		
懐加戦问题列表 ③ 全部问题状态 * 図4単〇 (前私)二言程序的设备 ()		
问题阅述	優加裁用户(占比) ◆	僵加载次数(占比) ≄
ID: 5 would be a set of the set o	1(100%)	31572(100%)
共1条		

对于每一个页面加载样本,页面完全加载时间大于3500ms为慢加载,这些慢加载样本会显示在问题列表中。

慢加载问题分析							
样本分析 统计分析							
样本列表 ⑦ 講选择 🔻	上下文信息						
设备D: 2 A A A A A A A A A A A A A A A A A A	用户ID 123456 日	设备ID 2a0d558b9c3ed36b4c201308fe8	完全加载耗时 6 4845ms 6	应用版本 4.1.9 	系统版本 10} 后	页面 https://www.nasa.gov/ 阳	浏贤 Chr
上版时间: 2024-04-28 17.52.11 完全加载时间: 4845ms 应用版本: 4.1.9	网络美型 WIFI 后 构建ID	运营商 未知 后 SDK版本	地区 - 石	CPU위성 arm64-v8a ff	ua Mozilla/5.0 (Linux; Android	上报时间 d 10; AN 佰 2024-04-28 17:52:11佰	发生 202
设款(D: 2a0	939598bc-3405-494b-8398-0a8a 错误信息 时序分析	F5 5.1.0F5					
▲27日度本: 4.7.3 设备D::	页面加载源布函 苗 ³ DNS 查词 3 8ms	字节: 225ms				DOMReady:	3054ms (?)
12巻60:2 月1年10:123455 上間時間:2024-04-28 17 50:58 完全加減期時:4445ms 放用販売:44.9	TCP连接 SSL连接 网络请求首字节	93ms 81ms 70ms					
20月10:2-4-75 用(中10):123455 上出時111:224453 先会加盟期1:2024-04-28 17 50 34 天会加盟期1:4045ms 設別版本: 4.8.7	pg和texel DOM解析 资源加载	zums					2813ms

JS 错误问题列表

您可以在 JS 错误问题列表查看所有 JS 报错。

说明:

JS 错误问题上报默认采样率为0.1%,因此问题列表中问题样本数量与指标统计不吻合为正常现象。



		_
webview		
慢加载 慢加载问题列表 JS错误 JS错误问题列表		
	and a state of the second seco	
Rum-test500353 rum-android-demo v 前移 2024-03-01 1/25 ~ 2024-04-30 1/25 1 店都 时间和屋 大阪族 v 应用版本 前高诗	 ▼ 熟祝版本 靖遠洋 ▼ 刻克器 靖遠洋 	
网络税型 消息時 ★ 地区 消息時 ▼ 近医 消息時 ▼ 近置商 消息時 ▼ 页面 消息時 ▼		
① JS错误样本上提款认误样率为0.1%。因此问题对表中问题样本教量与指标统计不确合为正常跟途、样本采祥率的描加可能许及资源户省、如需调整课程系称功应按回队		
-		
JS错误问题列表 全部错误失型 ▼ 全部问题状态 ▼ 役争10 ▼ 荷柏入嘉雯提宽的设备10		
问题既还	JS错误用户数(占比) 🕈	
ID: 6	1 (14.29%)	8
Unesught EvalError		
file_iliandroid_asset/monitor.html		
ID- 7540	1 (14 20%)	
Uncaught RangeError	1 (19.20.0)	
Invalid array length file://indoid.asset/monitor.html		
ID: febc	1 (14.29%)	î
Cannot read property T of null Bite internet is approximately be		
ID: 1cebdc:	1 (14.29%)	1
Uncaught ReferenceError		

您还可以单击问题概述查看 JS 错误问题详情,定位分析 JS 错误原因。

JS错误问题分析 样本分析 统计分析							
样本列表 ⑦ 请选择 🔻	上下文信息						
设备ID: 280455 10 用户ID: 123456 上短时间: 2024-04-28 17:53 31	▲ 用户ID 123456 ℃ 演覧器版本	设备ID 2a0d558b9c3ed36b4c201308fe8 设备名称	借误类型 「Uncaught EvalError」 网络类型	错误信息 Hello 后 运营商	错误JS文件名 file:///android_asset/monitor.html 匠 地区	应用版本 4.6.8 后 CPU架构	3
应用版本: 4.6.8	78.0.6.8 lb	vivo X9L l	WIFIE	未知后	-10	arm64-v8al	-
設備10: 2x0 用)中10: 123456 上期時期): 2024-04-28 17:51:55 度用版本: 4.2.1	翻译状态 已翻译后 错误信息	APM68만 104ae98b-87d8-4325-8031-8b57	构建ID 后 5a379973-e977-455c-a13e-7c	SDK版本 357 百 5.1.0-jmeter 百			
设备10:2ad#2556-5-2455.45406-64646 用户口: 上服時詞:2024-04-2817.5019 原用版本:4.5.6	<u>*年18日本</u> 全部折叠						
设备ID: 2a0d55€ 用户ID: 123456 上版时间: 2024-04-28 17.48.45	1 at HTMLButtonElemen	nt. <anonymous> (file:///android_asset/monitor.html</anonymous>	:17:15)				
应用版本: 4.5.6	2 at http://localhost.5000	/static/js/main.dd03d93d.chunk.js:1:398					
设备ID: 2a0d556 用户ID: 123456	3 at i (http://localhost:50	00/static/js/main.dd03d93d.chunk.js:1:438)					

指标说明

相关指标说明如下表所示:

指标名称	指标说明
页面加载次数	打开或刷新页面的次数
完全加载耗时	指整个网页完全加载的时间
JS 错误次数	指定时间内的 JS 错误总数



JS 错误率	发生 JS 错误的用户数/访问 webview 页面的总用户数,由于计算资源限制,该指标分子 及分母为未去重数据
JS 错误用户比例	指定时间范围内受到 JS 错误影响的用户数与总用户的比值
JS 错误用户数	指定时间范围内受到 JS 错误影响的用户数



应用管理

最近更新时间:2024-05-13 18:03:25

操作场景

在应用管理页您可以查看您已接入的终端应用信息,以及新的接入应用、应用 ID、设置白名单信息。

操作步骤

接入应用

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧导航栏中选择终端性能监控 > 总览。

3. 单击**应用接入**,填写应用名称、选择 Android 或 IOS 应用类型和所属业务系统,然后单击**下一步**。

应用接入			
1 创建应用 >	2 应用接入		
应用名称(4-255个字符)	example		${\boldsymbol{ \oslash}}$
应用类型	安卓		• 📀
所属业务系统	rum-5ms0lNiVQ14zVQ.dd		▼ 还没有业务系统? 点哉
		下一步取消	

获取应用 ID

您可以在应用管理 >应用设置页面,您可以切换成菜单为应用设置,在应用设置列表中获取应用 ID。



应用管理					
12.595 dent 12.144 Q.11 C 12.144 E 14: 业务系统:::					
应用名	上报 id	应用 ID	类型	申请时间	展示
rum-android-demo	73129ae7-17456 🖺	50	安卓	2024-04-19 11:06:03	接入摺引 上报量统计
rum-ios-demo	3e3073c0-3712 1	5°	iOS	2024-04-19 11:07:50	接入指引 上报量统计
共 2 条					

白名单配置

您可以在应用管理 > 白名单管理页面,点击白名单配置下方的添加,通过配置用户 ID/设备 ID 白名单可以避免数据 上报受采样影响。即加入白名单的用户/设备不受采样率影响,将会全部采样。您可以选择用户或者设备类型,并填 写相关 ID 即可。

新增白名单账号	
账号类型	用户 ▼
用户ID	请输入用户ID
备注	请输入用户白名单的备注
	确定取消