

阿里云 应用实时监控服务

APP 监控

文档版本：20190910

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
<code>##</code>	表示参数、变量。	<code>bae log list --instanceid Instance_ID</code>
<code>[]或者[a b]</code>	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{ }或者{a b}</code>	表示必选项，至多选择一个。	<code>swich {stand slave}</code>

目录

法律声明.....	I
通用约定.....	I
1 开始监控 Android 应用.....	1
2 开始监控 iOS 应用.....	4
3 参考信息.....	7
3.1 崩溃指标说明.....	7

1 开始监控 Android 应用

本文适用于 Android 应用的接入。APP 监控提供了完备的客户端崩溃分析监控，具体包括 JAVA Crash 监控、Native Crash 监控和 ANR 监控等。

操作步骤

1. 登录 [ARMS 控制台](#)，在左侧导航栏中选择 APP 监控。
2. 在 APP 监控页面，单击右上角的添加 APP 监控。
3. 在弹出的添加 APP 监控对话框的填写应用信息页签输入应用名称和 Android 应用的 PackageName，并选择 Android 平台，单击创建应用。

The screenshot shows a dialog box titled "添加APP监控" (Add APP Monitoring) with a close button (X) in the top right corner. It features three tabs: "填写应用信息" (Fill in application information), "查看配置" (View configuration), and "添加SDK" (Add SDK). The "填写应用信息" tab is active. It contains three input fields: "应用信息" (Application information) with the placeholder "请输入应用名称" (Please enter the application name), "平台" (Platform) with radio buttons for "Android" (selected) and "iOS", and "PackageName" with the placeholder "com.company.android". At the bottom right, there are two buttons: "取消" (Cancel) and "创建应用" (Create application).

4. 在查看配置页签复制 AppKey 和 AppSecret 的值，单击下一步。

The screenshot shows the same dialog box, but the "查看配置" (View configuration) tab is now active. It displays the generated credentials: "AppKey: 278888888", "AppSecret: 02ed1c8d85dc707798d308d0bdc9b3e1", and "PackageName: com.company.android". At the bottom right, the buttons are "取消" (Cancel) and "下一步" (Next step).

5. 添加 Android SDK。此内容适用于使用 Gradle 管理依赖的 Android Studio 项目。

a) 在程序中引入阿里云 Maven 仓库依赖。

A. 在项目 *build.gradle* 中添加阿里云 Maven 仓库地址。

```
repositories{
    maven { url "http://maven.aliyun.com/nexus/content/
repositories/releases" }
}
```

B. 在 APP 模块的 *build.gradle* 的 *dependencies* 节点中添加以下依赖。

```
compile('com.aliyun.ams:alicloud-android-ha-adapter:1.1.2.2-
open@aar') {
    transitive=true
}
```

b) 初始化实例。

A. 在自定义 *Application* 类的 *onCreate* 里面启动服务。

```
public class MyApplication extends Application {
    @Override
    public void onCreate() {
        initHa();
    }

    private void initHa() {
        Log.e("ha", "init");
        //这里必须启动，否则服务端收不到数据
        AliHaAdapter.getInstance().openPublishEmasHa();

        AliHaConfig config = new AliHaConfig();
        config.appKey = "<yourAppKeyId>"; //appkey
        config.appVersion = "x.xx"; //应用的版本号信箱
        config.appSecret = "<yourAppKeySecret>"; //appsecret
        config.channel = "<yourChannelId>"; //应用的渠道号标
记, 自定义
        config.userNick = null;
        config.application = this;
        config.context = getApplicationContext();
        config.isAliyunos = false; //是否为yunos
        AliHaAdapter.getInstance().startCrashReport(config);
        //启动CrashReport
    }
}
```

B. *AndroidManifest.xml* 里面指定自定义 *Application*。

```
<application
    android:name=".MyApplication"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme" >
    ...
    ...
```

```
</application>
```

c) 更新渠道标记。

```
AliHaAdapter.getInstance().updateChannel("600000");
```

d) 更新自定义标记。

```
AliHaAdapter.getInstance().updateUserNick("aliyun");
```

预期结果

在完成上述步骤之后，您可以对您的 APP 进行测试，并登录 [ARMS 控制台](#) 查看数据报表。

2 开始监控 iOS 应用

本文档适用于使用 CocoaPods 管理依赖的 Xcode 项目。iOS APP 监控提供了完备的客户端崩溃分析监控，具体包括 Crash 监控和 ANR 监控。

前提条件

应用的 iOS 版本为 8.0 或以上。

操作步骤

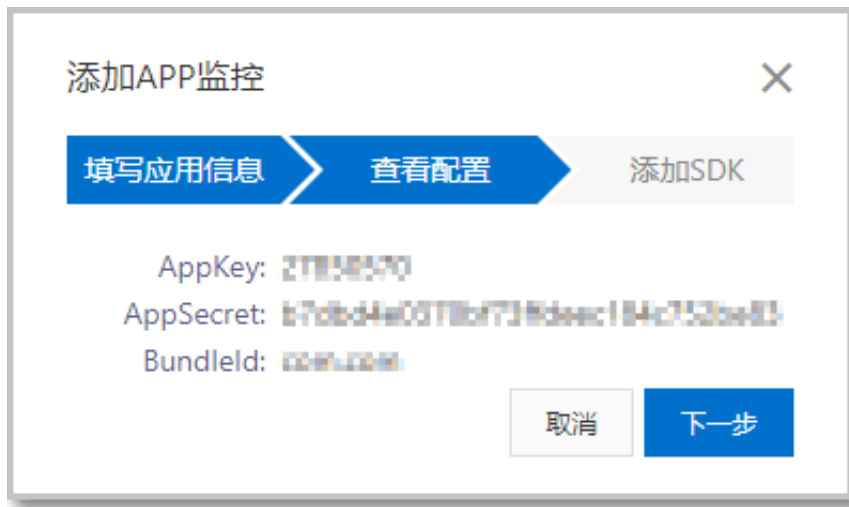
1. 登录 [ARMS 控制台](#)，在左侧导航栏中选择 APP 监控。
2. 在 APP 监控页面，单击右上角的添加 APP 监控。
3. 在弹出的添加 APP 监控对话框的填写应用信息页签输入应用名称和 iOS 应用的 BundleId，并选择 iOS 平台，单击创建应用。

The screenshot shows a dialog box titled "添加APP监控" (Add APP Monitoring) with a close button (X) in the top right corner. The dialog has three tabs: "填写应用信息" (Fill Application Information), "查看配置" (View Configuration), and "添加SDK" (Add SDK). The "填写应用信息" tab is currently selected and highlighted in blue. Below the tabs, there are three labeled input fields:

- * 应用信息 (Application Information): A text input field containing the placeholder text "请输入应用名称" (Please enter application name).
- * 平台 (Platform): A radio button selection area with two options: "Android" (unselected) and "iOS" (selected, indicated by a blue dot).
- * BundleId: A text input field containing the value "com.company.ios".

At the bottom right of the dialog, there are two buttons: "取消" (Cancel) and "创建应用" (Create Application).

4. 在查看配置页签复制 AppKey 和 AppSecret 的值，单击下一步。



5. 添加 iOS SDK。

- a) 添加依赖。

- A. 指定官方仓库和阿里云仓库。

```
source "https://github.com/CocoaPods/Specs.git"
source "https://github.com/aliyun/aliyun-specs.git"
```

- B. 添加依赖。

```
pod 'AlicloudHA', '~> 1.1.0'
```



说明:

~>为模糊指定版本号方式，~> 1.1.0 表明引用位于 1.1.0 版本与 1.2.0 版本之间最新版本的 SDK，用户可根据项目需要指定 SDK 版本，详情请参见 [Podfile Syntax Reference](#)。

- C. 执行 pod update。

如果在 Xcode 9 上出现以下错误，请将 Project Format 改成 Xcode 8.0-compatible。

```
RuntimeError - [Xcodeproj] Unknown object version.
```

- b) 在 *AppDelegate.m* 文件的 `application:didFinishLaunchingWithOptions:` 方法中初始化 SDK。

引入头文件。

```
#import<AliHAAdapter4Cloud/AliHAAdapter.h>
```

```
#import<TBCrashReporter/TBCrashReporter.h>
```

示例代码：

```
NSString *appKey = @"<yourAppKeyId>"; //appId
NSString *secret = @"<yourAppKeySecret>"; //appSecret
NSString *channel = @"<yourChannelId>"; //渠道标记，自定义，比如不同的应用商店等
NSString *appVersion = @"x.x"; //app版本
NSString *nick = @"<yourNickId>"; //选填。自定义标记，会上报给崩溃分析服务，可用于搜索崩溃数据
id<AliHAPPluginProtocol> crashPlugin = [TBCrashReporter sharedReporter];
NSArray<id<AliHAPPluginProtocol>> *plugins = @[crashPlugin];
[AliHAAdapter initWithAppKey:appKey secret:secret appVersion:appVersion channel:channel plugins:plugins nick:nick];
```

c) 进行编译。

- 如果编译报错，请在项目的 build setting 里设置 Allow Non-modular Includes In Framework Modules 为 YES。
- 如果出现包含 duplicate symbol 的错误，请确认其他本地依赖和 CocoaPods 管理的依赖是否有重复。如有重复，请删除本地依赖。建议所有依赖都通过CocoaPods管理。

预期结果

在完成上述步骤之后，您可以对您的 APP 进行测试，并登录 [ARMS 控制台](#)查看数据报表。

3 参考信息

3.1 崩溃指标说明

崩溃分析，是将 Android 和 iOS 平台常见的 APP 崩溃问题进行归类分析，帮助企业根据崩溃指标快速发现、定位问题。

Android 崩溃指标

崩溃指标	数据定义	公式定义
总体 Crash 率	Crash 基础指标，表示 Native + Java 类型的崩溃率。	今天实时 Crash 率 = 今天 0 点到当前时间 Crash 发生累计次数 / 今天 0 点到当前时间应用（版本）总启动次数。
Native Crash 率	Crash 基础指标，表示 Native 的崩溃率。	今天实时 Native Crash 率 = 今天 0 点到当前时间 Native 类型 Crash 发生累计次数 / 今天 0 点到当前时间应用（版本）总启动次数。
Native 用户 Crash 率	Crash 基础指标，表示 Native 的用户崩溃率。	今天实时 Native 用户 Crash 率 = 今天 0 点到当前时间 Native 类型 Crash 影响的设备去重数量 / 今天累计 UV。
Native Crash 次数	Crash 基础指标，表示 Native 的崩溃次数。	今天实时 Native Crash 次数 = 今天 0 点到当前时间 Native 类型 Crash 发生累计次数。
Java Crash 率	Crash 基础指标，表示 Java 的崩溃率。	今天实时 Java Crash 率 = 今天 0 点到当前时间 Java 类型 Crash 发生累计次数 / 今天 0 点到当前时间应用（版本）总启动次数。
Java 用户 Crash 率	Crash 基础指标，表示 Java 的用户崩溃率。	今天实时 Java 用户 Crash 率 = 今天 0 点到当前时间 Java 类型 Crash 影响的设备去重数量 / 今天累计 UV。
Java Crash 次数	Crash 基础指标，表示 Java 的崩溃次数。	今天实时 Java Crash 次数 = 今天 0 点到当前时间 Java 类型 Crash 发生累计次数。

iOS 崩溃指标

崩溃指标	数据定义	公式定义
总体 Crash 率	Crash 基础指标，表示全堆栈 + Abort 类型的崩溃率。	今天实时 Crash 率 = 今天 0 点到当前时间 Crash 发生累计次数 / 今天 0 点到当前时间应用（版本）总启动次数。
全堆栈 Crash 率	Crash 基础指标，表示全堆栈的崩溃率。	今天实时全堆栈 Crash 率 = 今天 0 点到当前时间全堆栈类型 Crash 发生累计次数 / 今天 0 点到当前时间应用（版本）总启动次数。
全堆栈用户 Crash 率	Crash 基础指标，表示全堆栈的用户崩溃率。	今天实时全堆栈用户 Crash 率 = 今天 0 点到当前时间全堆栈类型 Crash 影响的设备去重数量 / 今天累计 UV。
全堆栈 Crash 次数	Crash 基础指标，表示全堆栈的崩溃次数。	今天实时全堆栈 Crash 次数 = 今天 0 点到当前时间全堆栈类型 Crash 发生累计次数。
Abort 异常未捕获 Crash 率	Crash 基础指标，表示 Abort 的崩溃率。	今天实时 Abort Crash 率 = 今天 0 点到当前时间 Abort 类型 Crash 发生累计次数 / 今天 0 点到当前时间应用（版本）总启动次数。
Abort 异常未捕获用户 Crash 率	Crash 基础指标，表示 Abort 的用户崩溃率。	今天实时 Abort 用户 Crash 率 = 今天 0 点到当前时间 Abort 类型 Crash 影响的设备去重数量 / 今天累计 UV。
Abort 异常未捕获 Crash 次数	Crash 基础指标，表示 Abort 的崩溃次数。	今天实时 Abort Crash 次数 = 今天 0 点到当前时间 Abort 类型 Crash 发生累计次数。