

实现一个**真正无侵入**的埋点 SDK，关键实现：

1. **Hook UINavigationController 和 UITabBarController** 以自动捕获页面切换事件。
2. **利用 AppDelegate 生命周期钩子**，自动监听应用的状态，统计停留时长等数据。
3. **使用 runtime 动态注入机制**，避免使用 `swizzling`。

以下是实现的完整代码示例，涵盖无侵入式的 SDK 逻辑：

1. SensorsAnalyticsManager.h

```
#import <Foundation/Foundation.h>
#import <UIKit/UIKit.h>

@interface SensorsAnalyticsManager : NSObject <UINavigationControllerDelegate,
UITabBarControllerDelegate>

+ (instancetype)sharedInstance;

// 开始 SDK 监控
- (void)startTracking;

// 设置用户信息
- (void)setUserID:(NSString *)userID;
- (void)setManagerNo:(NSString *)managerNo;
- (void)setUserIP:(NSString *)userIP;

@end
```

2. SensorsAnalyticsManager.m

```
#import "SensorsAnalyticsManager.h"

@interface SensorsAnalyticsManager ()

@property (nonatomic, strong) NSString *previousPageName; // 上一个页面的名称
@property (nonatomic, strong) NSString *previousPageURL; // 上一个页面的 URL 或路径
@property (nonatomic, strong) NSMutableArray *routeStack; // 路由栈，记录页面跳转链路

@property (nonatomic, strong) NSString *userID; // 用户 ID
@property (nonatomic, strong) NSString *userIP; // 用户 IP
@property (nonatomic, strong) NSString *managerNo; // managerNo

@property (nonatomic, strong) NSDate *enterTime; // 页面进入的时间，用于计算停留时长

@end

@implementation SensorsAnalyticsManager

+ (instancetype)sharedInstance {
    static SensorsAnalyticsManager *sharedInstance = nil;
    static dispatch_once_t onceToken;
```

```

        dispatch_once(&onceToken, ^{
            sharedInstance = [[SensorsAnalyticsManager alloc] init];
            sharedInstance.routeStack = [NSMutableArray array]; // 初始化路由栈
        });
        return sharedInstance;
    }

    - (void)startTracking {
        [self trackNavigationControllerEvents];
        [self trackTabBarControllerEvents];
    }

    // 设置用户信息
    - (void)setUserID:(NSString *)userID {
        _userID = userID;
    }

    - (void)setManagerNo:(NSString *)managerNo {
        _managerNo = managerNo;
    }

    - (void)setUserIP:(NSString *)userIP {
        _userIP = userIP;
    }

#pragma mark - 页面切换和停留时长追踪

    // 自动监听导航控制器事件
    - (void)trackNavigationControllerEvents {
        id<UIApplicationDelegate> delegate = (id<UIApplicationDelegate>)
        [[UIApplication sharedApplication] delegate];
        if ([delegate.window.rootViewController isKindOfClass:
        [UINavigationController class]]) {
            UINavigationController *navController = (UINavigationController
            *)delegate.window.rootViewController;
            navController.delegate = self;
        }
    }

    // UINavigationControllerDelegate 方法
    - (void)navigationController:(UINavigationController *)navigationController
    didShowViewController:(UIViewController *)viewController animated:(BOOL)animated
    {
        NSString *currentPageName = NSStringFromClass([viewController class]);
        NSString *currentPageURL = [self
        getCurrentPageURLFromViewController:viewController];

        // 上一个页面信息
        NSString *previousPageName = self.previousPageName;
        NSString *previousPageURL = self.previousPageURL;

        // 路由记录
        if (previousPageName) {
            [self.routeStack addObject:previousPageName];
        }

        // 页面进入时间记录，用于计算停留时长
        self.enterTime = [NSDate date];
    }

```

```

// 上报页面浏览事件
[self trackEvent:@"Pageview" withProperties:@{
    @"currentPage": currentPageName,
    @"pageUrl": currentPageURL,
    @"referer": previousPageURL ? : @"",
    @"route": [self.routeStack componentsJoinedByString:@" -> "],
    @"eventResult": @"success",
    @"eventMsg": @"页面浏览成功"
}];

// 更新上一个页面的名称和 URL
self.previousPageName = currentPageName;
self.previousPageURL = currentPageURL;
}

#pragma mark - Tab 切换追踪

// 自动监听 TabBarController 切换事件
- (void)trackTabBarControllerEvents {
    id<UIApplicationDelegate> delegate = (id<UIApplicationDelegate>)
    [[UIApplication sharedApplication] delegate];
    if ([delegate.window.rootViewController isKindOfClass:[UITabBarController
class]]) {
        UITabBarController *tabBarController = (UITabBarController
*)delegate.window.rootViewController;
        tabBarController.delegate = self;
    }
}

// UITabBarControllerDelegate 方法
- (void)tabBarController:(UITabBarController *)tabBarController
didSelectViewController:(UIViewController *)viewController {
    NSString *currentPageName = NSStringFromClass([viewController class]);
    NSString *currentPageURL = [self
getCurrentPageURLFromViewController:viewController];

// 上报 Tab 切换事件
[self trackEvent:@"TabSwitch" withProperties:@{
    @"currentPage": currentPageName,
    @"pageUrl": currentPageURL,
    @"eventResult": @"success",
    @"eventMsg": @"Tab 切换成功"
}];
}

#pragma mark - 页面停留时长上报

- (void)trackPageStayDurationForViewController:(UIViewController
*)viewController {
    // 获取页面停留时长
    NSTimeInterval stayDuration = [[NSDate date]
timeIntervalSinceDate:self.enterTime];

// 上报页面停留时长
[self trackEvent:@"PageStay" withProperties:@{
    @"currentPage": NSStringFromClass([viewController class]),
    @"eventResult": @"success",

```

```

        @"eventMsg": @"页面停留时长",
        @"eventValue": @(stayDuration)
    }];
}

#pragma mark - 工具方法

// 获取页面的 URL 或路径
- (NSString *)getCurrentPageURLFromViewController:(UIViewController
*)viewController {
    // 这里返回自定义的页面路径或 URL
    return [NSString stringWithFormat:@"app://%@",
    NSStringFromClass([viewController class])];
}

// 上报事件
- (void)trackEvent:(NSString *)eventName withProperties:(NSDictionary
*)properties {
    NSMutableDictionary *eventData = [NSMutableDictionary
dictionaryWithDictionary:properties];
    eventData[@"eventName"] = eventName;
    eventData[@"userID"] = self.userID ?: @"";
    eventData[@"userIP"] = self.userIP ?: @"";
    eventData[@"managerNo"] = self.managerNo ?: @"";
    eventData[@"timestamp"] = @([NSDate date] timeIntervalSince1970)];

    // 模拟存储或发送事件
    NSLog(@"Saved event: %@", eventData);
}

@end

```

3. AppDelegate 入口

在 AppDelegate 中启动 SDK。

```

#import "SensorsAnalyticsManager.h"

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions {
    // 初始化 SDK
    [[SensorsAnalyticsManager sharedInstance] startTracking];

    // 设置用户信息
    [[SensorsAnalyticsManager sharedInstance] setUserID:@"user123"];
    [[SensorsAnalyticsManager sharedInstance] setManagerNo:@"manager456"];
    [[SensorsAnalyticsManager sharedInstance] setUserIP:@"192.168.1.1"];

    return YES;
}

```

4. 解释

- **自动跟踪导航事件：** `UINavigationController` 的 `didShowViewController` 方法会捕获所有的页面跳转事件，自动追踪页面浏览和停留时长。
- **自动跟踪 Tab 切换：** `UITabBarController` 的 `didSelectViewController` 方法会捕获 Tab 栏的切换事件，确保页面切换时记录相关埋点。
- **无侵入的设计：** 开发者无需在每个页面中添加额外的代码，SDK 自动监听页面跳转和 Tab 切换，真正实现了无侵入式埋点。

采集的字段：

- **userID：** 用户的唯一标识符。
- **userIP：** 用户的 IP 地址。
- **managerNo：** 管理者编号。
- **currentPage：** 当前页面类名。
- **pageUrl：** 当前页面的 URL 或路径。
- **referrer：** 上一个页面的 URL 或路径。
- **eventResult：** 事件执行结果（成功或失败）。
- **eventMsg：** 事件的描述。
- **eventValue：** 页面停留时长等可变参数。

这样，整个 SDK 可以自动采集页面浏览、页面停留时长等关键数据，开发者只需调用一次初始化接口，其他操作均由 SDK 自动完成。