

插件支持

由于使用第三方插件后，小程序不允许重写App、Page 对象，不允许修改官方API的定义，导致监控失效，探针提供一种针对插件的解决方案，需要手工调用探针接口来支持。插件相关API需要在开启 插件模式 前提下使用，通过探针启动配置中设置生效。插件模式开启后，探针不会主动监控，需要用户手动设置。

- 为了解决低版本启动监控引发的错误，小程序探针 插件模式 默认开启。即探针默认行为是 **低于小程序SDK版本2.6.4不启动监控，2.6.4及以上版本自动启动监控**(微信小程序SDK在2.6.4及以后版本放开了引入插件时的限制，因此可以直接启动监控)。如果小程序项目低于2.6.4并且没有使用插件，可以通过配置 `plugin: false` 强制关闭插件模式来监控低版本。
- 企业微信小程序可能默认SDK版本较低，若无数据且没有使用插件，建议设置 `plugin: false` 关闭插件模式。

标识使用插件

在小程序探针初始化配置中可以设置插件模式开关: `plugin`。

- `true`: 开启
- `false`: 关闭

示例：

```
const monitor = require('./agent/tingyun-mp-agent.js');
monitor.config({
  ...,
  plugin: true,
  ...
})
```

App

使用 `hookApp` 接口封装App原始参数。

app.js

```
const monitor = require('./agent/tingyun-mp-agent.js');

App(monitor.hookApp({
  onLaunch: function() {

  },
  ...
}))
```

Page

使用 `hookPage` 接口封装Page原始参数。

```
const monitor = require('./agent/tingyun-mp-agent.js');

Page(monitor.hookPage({
  onLoad: function() {

  },
  ...
}))
```

Component

对于使用组件构建页面的情况使用 `hookComponent`。

```
const monitor = require('./agent/tingyun-mp-agent.js');

Component(monitor.hookComponent({

...
}))
```

网络请求

使用探针提供的接口 `monitor.request` 发送网络请求，参数与官方的 `wx.request` 接口一致。

```
const monitor = require('./agent/tingyun-mp-agent.js');

monitor.request({
  url: '...'
});
```