

iOS-SDK接入文档之AliNetworkDiagnosis篇

SDK 方法

```
/**
 * 初始化网络质量分析 SDK
 * @param secretKey 阿里网络质量分析控制台获取的应用密钥
 * @param deviceId 设备唯一标识, 选填, 传入 nil 时 SDK 会自动生成
 * @param siteId 两字母国家标识, 选填, 如中国是 CN, 传入 nil, SDK 会根据客户端出口公网 IP 确定所在国家
 * @param extParamsDict 设置自定义信息, 选填, 会在探测结果数据中带上, 可设置多组 key-value; 可以传入 nil
 */
+ (void)aLiNetworkDiagnosisInitWithSecretKey:(NSString *)secretKey deviceId:(NSString *)deviceId siteId:(NSString *)siteId extParamsDict:(NSDictionary *)extParamsDict;
```

```
/**
 * HTTP 探测接口
 * @param url 探测 URL, 必填
 * @param traceId 非必选, 探测 ID, 本次任务唯一标识, 可传 nil, 如果传入 nil, 会自动生成, 回调的时候返回
 * @param context 非必选, 回调上下文, 可传 nil
 * @param resultBlock 结果回调 block, 可传 nil
 */
+ (void)aLiNetworkDiagnosisStartHttpPingUrl:(NSString *)url traceId:(NSString *)traceId context:(id)context resultBlock:(AliHttpPingCompleteHandler)resultBlock;
```

```
/**
 * PING 探测接口
 * @param url 探测 URL, 必填
 * @param timeout 超时时间, 默认值 1000
 * @param interfaceType 本次探测使用的网络接口
 * @param prefer 偏好, 默认值 1
 * @param traceId 非必选, 探测 ID, 本次任务唯一标识, 可传 nil, 如果传入 nil, 会自动生成, 回调的时候返回
 * @param context 非必选, 回调上下文, 可传 nil
 * @param size 报文大小, ping 包字节数, 默认值 20
```

```

* @param count ping 探测次数，每次发一个 PING 包，默认值 10
* @param interval 间隔时间，默认值 200
* @param resultBlock 结果回调 block，可传 nil
* @param combineResultBlock 组合结果回调 block，可传 nil
*/
+ (void)aLiNetworkDiagnosisStartPingUrl:(NSString
*)url timeout:(NSInteger)timeout
interfaceType:(AliNetDiagNetworkInterfaceType)interfa
ceType prefer:(NSInteger)prefer traceId:(NSString
*)traceId context:(id)context size:(NSInteger)size
count:(NSInteger)count interval:(NSInteger)interval
resultBlock:(AliPingCompleteHandler)resultBlock
combineResultBlock:(AliPingCombineCompleteHandler)com
bineResultBlock;

```

```

/**
* TCP 探测接口
* @param url 探测 URL,必填
* @param timeout 超时时间，默认值 1000
* @param interfaceType 本次探测使用的网络接口
* @param prefer 偏好，默认 0
* @param traceId 非必选，探测 ID，本次任务唯一标识，可传 nil，
如果传入 nil，会自动生成，回调的时候返回
* @param context 非必选，回调上下文，可传 nil
* @param port TCP 端口
* @param count tcpping 探测次数，默认值 10
* @param interval 间隔时间，默认值 200
* @param resultBlock 结果回调 block，可传 nil
* @param combineResultBlock 组合结果回调 block，可传 nil
*/
+ (void)aLiNetworkDiagnosisStartTcpPingUrl:(NSString
*)url timeout:(NSInteger)timeout
interfaceType:(AliNetDiagNetworkInterfaceType)interfa
ceType prefer:(NSInteger)prefer traceId:(NSString
*)traceId context:(id)context port:(NSInteger)port
count:(NSInteger)count interval:(NSInteger)interval
resultBlock:(AliTcpPingCompleteHandler)resultBlock
combineResultBlock:(AliTcpPingCombineCompleteHandler)
combineResultBlock;

```

```

/**
* MTR 探测接口
* @param url 必选，需要探测的目标 IP 或者域名

```

```

* @param maxTtl 探测最多跳数，路径最长长度，默认值 30
* @param maxPaths 探测最多的路径，网络拓扑中两个点之间一般会有多条路径，这里指定本次探测的最大路径值，默认值 1
* @param maxTimeEachIP 每跳中的每个 IP 最多探测次数，默认值 10
* @param timeout 超时时间，默认值 1000
* @param interfaceType 本次探测使用的网络接口
* @param prefer 偏好，默认值 1
* @param traceId 非必选，探测 ID，本次任务唯一标识，可传 nil，如果传入 nil，会自动生成，回调的时候返回
* @param context 非必选，回调上下文，可传 nil
* @param resultBlock 结果回调 block，可传 nil
* @param combineResultBlock 组合结果回调 block，可传 nil
*/
+ (void)aLiNetworkDiagnosisStartMtrUrl:(NSString *)url maxTtl:(NSInteger)maxTtl maxPaths:(NSInteger)maxPaths maxTimeEachIP:(NSInteger)maxTimeEachIP timeout:(NSInteger)timeout interfaceType:(AliNetDiagNetworkInterfaceType)interfaceType prefer:(NSInteger)prefer traceId:(NSString *)traceId context:(id)context resultBlock:(AliMTRCompleteHandler)resultBlock combineResultBlock:(AliMTRCombineCompleteHandler)combineResultBlock;

```

```

/**
* DNS 探测接口
* @param url 探测 URL,必填
* @param nameServer 可以传 nil，默认会用 localdns
* @param type 请求记录类型，默认值@"A"
* @param timeout 超时时间，默认值 2000
* @param interfaceType 本次探测使用的网络接口
* @param traceId 非必选，探测 ID，本次任务唯一标识，可传 nil，如果传入 nil，会自动生成，回调的时候返回
* @param context 非必选，回调上下文，可传 nil
* @param resultBlock 结果回调 block，可传 nil
*/
+ (void)aLiNetworkDiagnosisStartDnsUrl:(NSString *)url nameServer:(NSString *)nameServer type:(NSString *)type timeout:(NSInteger)timeout interfaceType:(AliNetDiagNetworkInterfaceType)interfaceType traceId:(NSString *)traceId

```

```
context:(id)context  
resultBlock:(AliDnsCompleteHandler)resultBlock;
```