

TingYun_App_iOS_SDK_接口说明-协议扩展

本文档介绍了 TingYun_SDK 基于自定义执行单元相关接口的使用及说明

一、协议扩展相关接口

1、创建执行单元

注意：startSpanWithName:operation: 需和 finish或finishWithStatus: 成对调用。

- 相关接口

```
/**
 * @brief 创建一个独立的span「执行单元」
 * @param name span 名称，不能为空，长度限制为1024字符，超过截取前1024 字符
 * @param operation span所属的operation名称，如websocket，不能为空，长度限制为128
 * 字符，超过截取前128字符
 */
+ (id<NBSSpan>)startSpanWithName:(NSString *)name operation:(NSString
*)operation;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpan startSpanWithName:@"span_name"
operation:@"span_operation"];
[span finish];
```

- Swift

```
let span = NBSSpan.startSpan(withName: "span_name", operation:
"span_operation")
span?.finish()
```

2、完成执行单元

注意：startSpanWithName:operation: 需和 finish或finishWithStatus: 成对调用。

- 相关接口

```
/**
 * @brief 完成一个span，并设置完成时间，状态值默认为NBSSpanStatusSuccess
 */
- (void)finish;

/**
 * @brief 完成一个span并设置完成时间与span的状态
 * @param status 状态，值可为：NBSSpanStatusUndefined、NBSSpanStatusSuccess、
NBSSpanStatusFailed
 */
- (void)finishWithStatus:(NBSSpanStatus)status;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpan startSpanWithName:@"span_name"
operation:@"span_operation"];
[span finishWithStatus:NBSpanStatusSuccess];
```

- Swift

```
let span = NBSSpan.startSpan(withName: "span_name", operation:
"span_operation")
span?.finish(with: NBSSpanStatus.success)
```

3、创建子执行单元

注意：startChildWithName:operation: 需和 finish或finishWithStatus: 成对调用。

- 相关接口

```
/**
 * @brief 创建一个带description「描述」的子span
 * @param name 名称；不能为空，长度限制为1024字符，超过截取前1024字符。
 * @param operation 类别；不能为空，长度限制为128字符，超过截取前128字符
 * @param description 描述；长度限制为1024字符，超过截取前1024字符
 */
- (id<NBSSpan>)startChildWithName:(NSString *)name operation:(NSString
*)operation;
- (id<NBSSpan>)startChildWithName:(NSString *)name operation:(NSString
*)operation description:(nullable NSString *)description;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpan startSpanWithName:@"span_name"
operation:@"span_operation"];
id<NBSSpan> sub = [span startChildWithName:@"sub_name"
operation:@"sub_operation"];
[sub finishWithStatus:NBSSpanStatusSuccess];
[span finishWithStatus:NBSSpanStatusSuccess];
```

- Swift

```
let span = NBSSpanAgent.startSpan(withName: "span_name", operation:
"span_operation")
let sub = span!.startChild(withName: "sub_name", operation:
"sub_operation")
sub.finish(with: NBSSpanStatus.success)
span!.finish(with: NBSSpanStatus.success)
```

4、设置 Data 数据

- 相关接口

```
/**
 * @brief 设置data的值
 */
- (void)setDataValue:(nullable id)value forKey:(NSString *)key;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpan startSpanWithName:@"span_name"
operation:@"span_operation"];
[span setDataValue:@"dataValue" forKey:@"dataKey"];
[span finishWithStatus:NBSSpanStatusSuccess];
```

- Swift

```
let span = NBSSpanAgent.startSpan(withName: "span_name", operation:
"span_operation")
span?.setDataValue("dataValue", forKey: "dataKey")
span?.finish(with: NBSSpanStatus.success)
```

5、设置 Tag 数据

- 相关接口

```
/**
 * @brief 设置tags的值
 * @warning value类型必须为字符串
 */
- (void)setTagValue:(NSString *)value forKey:(NSString *)key;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpanAgent startSpanWithName:@"span_name"
operation:@"span_operation"];
[span setTagValue:@"tagValue" forKey:@"tagKey"];
[span finishWithStatus:NBSSpanStatusSuccess];
```

- Swfit

```
let span = NBSSpanAgent.startSpan(withName: "span_name", operation:
"span_operation")
span?.setTagValue("tagValue", forKey: "tagKey")
span?.finish(with: NBSSpanStatus.success)
```

6、设置 Metric 数据

- 相关接口

```
/**
 * @brief 设置metrics的值
 * @param unit value的单位名称，比如毫秒“ms”，保留单位类型见NBSDefines.h中的
NBSSpanMetricUnitNames[ ]，支持自定义单位
 * @warning value的类型必须为NSNumber，unit的类型必须为字符串
 * @discussion 当span的operation为websocket时，如果设置的是sendSize、sendCount、
receiveSize、receiveCount、pingpongCount的值，SDK会自动累加
 */
- (void)setMetricValue:(NSNumber *)value forKey:(NSString *)key;
- (void)setMetricValue:(NSNumber *)value unit:(NSString *)unit forKey:(NSString
*)key;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpan startSpanWithName:@"span_name"
operation:@"span_operation"];
[span setMetricValue:@100 unit:@"ms" forKey:@"time"];
[span finishWithStatus:NBSSpanStatusSuccess];
```

- Swift

```
let span = NBSSpan.startSpan(withName: "span_name", operation:
"span_operation")
span?.setMetricValue(100, unit: "ms", forKey: "time")
span?.finish(with: NBSSpanStatus.success)
```

7、移除 Data 数据

- 相关接口

```
/**
 * @brief 从data中移除
 */
- (void)removeDataForKey:(NSString *)key;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpan startSpanWithName:@"span_name"
operation:@"span_operation"];
[span setDataValue:@"dataValue" forKey:@"dataKey"];
[span removeDataForKey:@"dataKey"];
[span finishWithStatus:NBSSpanStatusSuccess];
```

- Swift

```
let span = NBSSpan.startSpan(withName: "span_name", operation:
"span_operation")
span?.setDataValue("dataValue", forKey: "dataKey")
span?.removeData(forKey: "dataKey")
span?.finish(with: NBSSpanStatus.success)
```

8、移除 Tag 数据

- 相关接口

```
/**
 * @brief 从tags中移除
 */
- (void)removeTagForKey:(NSString *)key;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpan startSpanWithName:@"span_name"
operation:@"span_operation"];
[span setTagValue:@"tagValue" forKey:@"tagKey"];
[span removeTagForKey:@"tagKey"];
[span finishWithStatus:NBSSpanStatusSuccess];
```

- Swift

```
let span = NBSSpan.startSpan(withName: "span_name", operation:
"span_operation")
span?.setTagValue("tagValue", forKey: "tagKey")
span?.removeTag(forKey: "tagKey")
span?.finish(with: NBSSpanStatus.success)
```

9、移除 Metric 数据

- 相关接口

```
/**
 * @brief 从metrics中移除
 */
- (void)removeMetricForKey:(NSString *)key;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpan startSpanWithName:@"span_name"
operation:@"span_operation"];
[span setMetricValue:@100 unit:@"ms" forKey:@"time"];
[span removeMetricForKey:@"time"];
[span finishWithStatus:NBSSpanStatusSuccess];
```

- Swift

```
let span = NBSAppAgent.startSpan(withName: "span_name", operation:
"span_operation")
span?.setMetricValue(100, unit: "ms", forKey: "time")
span?.removeMetric(forKey: "time")
span?.finish(with: NBSSpanStatus.success)
```

10、设置 Status

- 相关接口

```
/**
 * @brief 设置span的status
 */
- (void)setStatus:(NBSSpanStatus)status;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSAppAgent startSpanWithName:@"span_name"
operation:@"span_operation"];
[span setStatus:NBSSpanStatusSuccess];
[span finish];
```

- Swift

```
let span = NBSAppAgent.startSpan(withName: "span_name", operation:
"span_operation")
span?.setStatus(NBSSpanStatus.success)
span?.finish()
```

11、设置 Duration

- 相关接口

```
/**
 * @brief 主动设置span执行耗时，默认耗时为startSpanWithName 到 finish。
 * @param value span执行耗时
 */
- (void)setDuration:(NSNumber *)value;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpan startSpanWithName:@"span_name"
operation:@"span_operation"];
[span setDuration:@100];
[span finish];
```

- Swift

```
let span = NBSSpan.startSpan(withName: "span_name", operation:
"span_operation")
span?.setDuration(100)
span?.finish()
```

12、设置 StatusCode

- 相关接口

```
/**
 *@brief 设置statusCode, 默认为0
 */
- (void)setStatusCode:(NSString *)value;
```

- 代码示例

- Objective-C

```
id<NBSSpan> span = [NBSSpan startSpanWithName:@"span_name"
operation:@"span_operation"];
[span setStatusCode:@"200"];
[span finish];
```

- Swift

```
let span = NBSSpan.startSpan(withName: "span_name", operation:
"span_operation")
span?.setStatusCode("200")
span?.finish()
```


