

# 位置情報収集基盤ジオフェンスライブラ リiOSインターフェース仕様書

株式会社ゼンリン

Version 4.0.0, 2024.10.30

# 目次

1. 改定履歴	1
2. はじめに	2
3. 概要	3
3.1. 機能概要	3
3.2. 動作環境	3
3.3. 提供の仕方	3
3.4. 基本的な使用方法	3
4. ライブラリの導入方法	5
4.1. ライブラリをプロジェクトに追加	5
4.2. プロジェクトの設定を変更	5
4.3. Swiftを用いている場合	5
5. API	6
5.1. defaultManager	6
5.2. checkLocation	7
5.3. resetGeoFenceStatus	13
5.4. addNotificationConditionInformation:	14
5.5. removeNotificationConditionInformation:	16
5.6. notificationConditionInformationList:	17
5.7. addAreaInformation:	18
5.8. addAreaInformationForGeoJson:	23
5.9. removeAreaInformation:	26
5.10. areaInformationList:	26
5.11. clearAllInformation	28
5.12. libraryVersion	29

# 1. 改定履歴

版数	改定内容	改定日
1.0.0	新規作成	2022/07/15
1.1.0	5.2. checkLocation 5.2.2. プロトタイプ宣言 ・パラメータ「認証サーバ環境」の有無でAPIを分けた 5.2.3. パラメータ ・認証サーバ環境をパラメータに追加	2022/09/14
2.0.0	対応するライブラリバージョンを「2.0.0」に変更 対応OSを「iOS15 / iOS16」に変更	2022/09/29
2.1.0	対応するライブラリバージョンを「2.1.0」に変更	2023/03/29
2.2.0	対応するライブラリバージョンを「2.2.0」に変更 認証エラーの記載を変更 「エラーコード一覧」を見直し checkLocationの引数（APIキー,クライアントID,秘密鍵）を必須項目に変更	2023/06/02
2.2.1	checkLocationに「Swiftを用いている場合」の記載を追加	2023/08/30
3.0.0	対応するライブラリバージョンを「3.0.0」に変更 対応OSに「iOS17」を追加 「Swiftを用いている場合」にcheckLocationメソッド実装の移行手順を追加	2023/10/06
3.1.0	対応するライブラリバージョンを「3.1.0」に変更	2023/12/21
4.0.0	対応するライブラリバージョンを「4.0.0」に変更 対応OSから「iOS15」を削除し「iOS18」を追加	2024/10/30

## 2. はじめに

本ドキュメントでは、各種のiOSアプリケーションにて、ジオフェンス機能を提供するiOSジオフェンスライブラリのインタフェースを記載しています。

本仕様書はライブラリのバージョン4.0.0について記載します。 それ以前のバージョンについては、以前のバージョンの仕様書を参照してください。

## 3. 概要

### 3.1. 機能概要

本インタフェースを利用することで、アプリケーションにて特定エリアに対するエリア内外判定結果を取得することが可能となります。

エリア内外判定を行うために必要な通知条件、エリア情報は、事前にアプリケーションからライブラリに設定しておく必要があります。

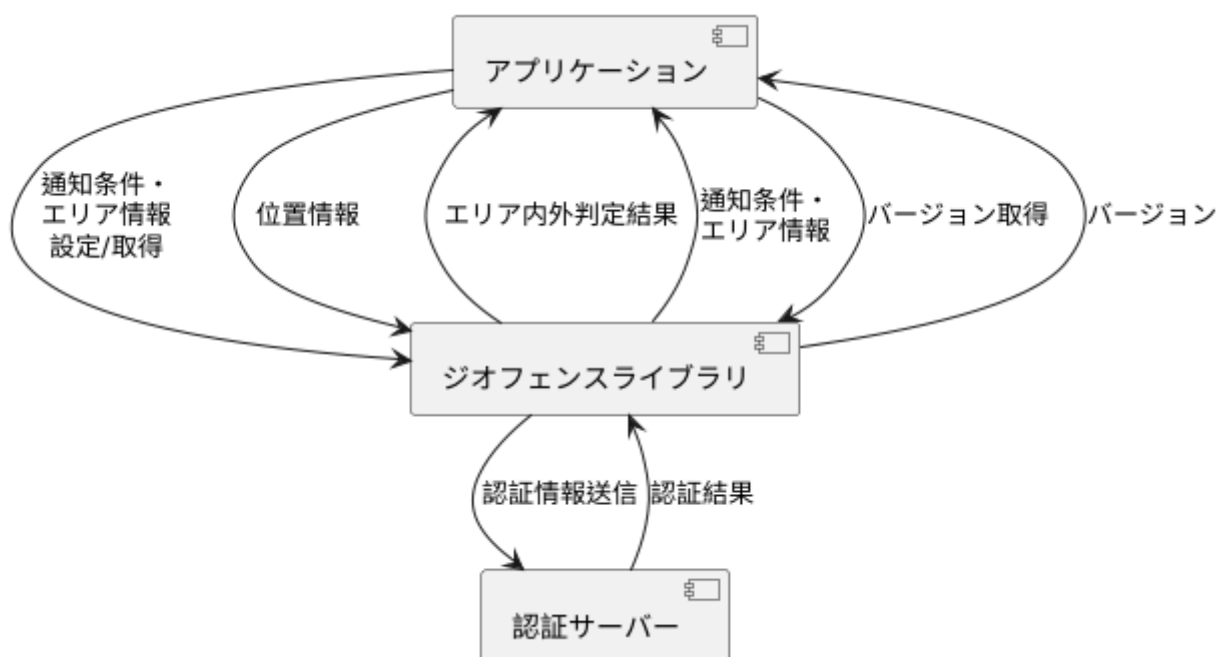


図1. iOSジオフェンスライブラリ概要

### 3.2. 動作環境

#### 3.2.1. 対応OS

iOS16 / iOS17 / iOS18

### 3.3. 提供の仕方

- 本ライブラリは、iOSアプリケーションにxcframework形式として提供されます。
- そのため、ユーザに本ライブラリの機能を使用したアプリケーションを提供する場合は、アプリケーションとリンクした形で提供する必要があります。

### 3.4. 基本的な使用方法

アプリケーションは、本ライブラリのGeoFenceManagerクラスを通して各操作を行います。

1. アプリケーションは、GeoFenceManagerのdefaultManagerを利用してインスタンスを取得します。

2. インスタンスに対して「通知条件」「エリア情報」を設定します。
3. インスタンスに対して「位置情報」を送ると、事前設定した「通知条件」「エリア情報」を基に、エリア内外判定が行われ、判定結果を得ることができます。

## 4. ライブラリの導入方法

既存のアプリにライブラリを導入する場合は、下記の手順で導入をしてください。

### 4.1. ライブラリをプロジェクトに追加

1. 対象プロジェクト内に任意のディレクトリを作成する
2. 作成したディレクトリの中に下記ファイルを入れる
  - GeoFenceManager.xcframework
3. Xcode上で対象プロジェクトに作成したディレクトリをドラッグ&ドロップする

### 4.2. プロジェクトの設定を変更

1. TARGETS>Build Setting>Other Linker Flags 項目を開き、-ObjCを追加する

### 4.3. Swiftを用いている場合

1. swiftの場合は[アプリ名]-Bridging-Header.hファイルを作成しTARGETS>Build Setting>Objective-C Bridging Headerにファイルのパスを追加する
2. [アプリ名]-Bridging-Header.hファイルには下記を記載する

```
#import "GeoFenceManager.h"
```

エラーが発生した場合は、サンプルアプリと見比べながら設定をご確認ください。

## 5. API

表 1 にジオフェンスライブラリのAPI一覧を示す。

表 1. API一覧

API	概要
defaultManager	クラスのインスタンスを取得します。
checkLocation	エリア内外判定を行います。
resetGeoFenceStatus	エリア滞在状態と、通知状態をリセットします。
addNotificationConditionInformation	通知条件を追加します。
removeNotificationConditionInformation	通知条件を削除します。
notificationConditionInformationList	通知条件を取得します。
addAreaInformation	エリア情報を追加します。
addAreaInformationForGeoJson	エリア情報（GeoJSON形式）を追加します。
removeAreaInformation	エリア情報を削除します。
areaInformationList	エリア情報を取得します。
clearAllInformation	通知条件、エリア情報をすべて削除します。
libraryVersion	ライブラリのバージョンを取得します。

### 5.1. defaultManager

#### 5.1.1. 説明

インターフェイスクラスのインスタンスを取得します。

#### 5.1.2. プロトタイプ宣言

```
+ (GeoFenceManager *)defaultManager;
```

#### 5.1.3. パラメータ

なし

#### 5.1.4. 戻り値

ジオフェンスライブラリのオブジェクト

#### 5.1.5. コメント

- ・シングルトンパターンを適用したクラスであるため、本APIはアプリケーションが起動してから終了するまで同一のオブジェクトを返します。



## 5.2. checkLocation

### 5.2.1. 説明

指定された位置情報が、エリアの内または、外かの判定を行い、該当する通知条件及び、エリア情報を返します。

処理実行時に認証を行いますので、別途発行されたAPIキー、クライアントID、秘密鍵をパラメータに指定してください。

認証サーバにて認証を行なった結果、認証不許可（APIキーやクライアントID、秘密鍵が誤っている等）と判断された場合は認証エラーとなり、エリア内外判定の結果は取得できません。

ネットワーク障害や通信タイムアウト（30秒）、サーバまたはライブラリの障害など、認証サーバで認証処理ができない場合は認証エラーにはならず、エリア内外判定の結果は取得できます。

本メソッドは、UIスレッド(メインスレッド)以外のスレッド(バックグラウンドスレッド、ワーカースレッドなど)から実行してください。

UIスレッドで実行した場合、認証処理が完了するまでUIが停止します。

### 5.2.2. プロトタイプ宣言

<認証サーバ環境指定なし>

```
- (NSArray *)checkLocation:(NSString *)apiKey
                        clientId:(NSString *)clientId
                        secret:(NSString *)secret
                        timeStamp:(NSDate *)timeStamp
                        latitude:(double)latitude
                        longitude:(double)longitude
                        horizontalAccuracy:(NSInteger)horizontalAccuracy
                        error:(NSError **)error
```

<認証サーバ環境指定あり>

```
- (NSArray *)checkLocation:(NSString *)apiKey
                        clientId:(NSString *)clientId
                        secret:(NSString *)secret
                        authEnv:(NSInteger)authEnv
                        timeStamp:(NSDate *)timeStamp
                        latitude:(double)latitude
                        longitude:(double)longitude
                        horizontalAccuracy:(NSInteger)horizontalAccuracy
                        error:(NSError **)error
```

### 5.2.3. パラメータ

apiKey	[I] : 認証用のAPIキー。 ※必須項目
clientId	[I] : 認証用のクライアントID。 ※必須項目
secret	[I] : 認証用の秘密鍵。 ※必須項目
authEnv	[I] : 認証サーバ環境。(表 2参照) ※引数指定なし、または表に存在しない値が指定された場合は、顧客本番環境に接続されます。
timeStamp	[I] : 測位情報のタイムスタンプ。 ※nil設定不可
latitude	[I] : 緯度。
longitude	[I] : 経度。
horizonataAccuracy	[I] : 測位誤差(m)。
error	[O] : 認証エラーの際のエラー情報。

表 2. 認証サーバ環境の種類

No.	認証サーバ環境の種類	定義名 (※1)	値
1	顧客本番環境	AuthEnv_Pro	0
2	顧客検証環境	AuthEnv_Stg	1
3	内部検証環境	AuthEnv_Test	2

(※1):定義名はGeoFenceManager.hに定義された文字列定数を使用します。

エラーの場合、以下のNSErrorが返却されます。

本エラーを受信した際には、アプリ側にてcheckLocationを再度実施することで、サーバー認証が再度行われます。（本ライブラリ内では、サーバー認証のリトライ処理は行いません）

表 3. 認証エラー内容

NSErrorのパラメーター	説明
code	エラーコード(表 4参照)
localizedDescription	エラーの詳細

表 4. エラーコード一覧

値	説明
1	トークン発行認証失敗
2	認証不許可

## 5.2.4. 戻り値

認証成功時は以下の情報を返却します。

該当した通知条件及び、エリア情報のDictionaryのArray。

該当した情報が無い場合、必須パラメータがない場合はnil。

Array内の各Dictionary の形式を表 5に示します。

表 5. 該当した通知条件およびエリア情報

No.	パラメータ概要	Key (※1)	形式	説明
1	通知条件検索のレスポンス情報	GFIResponseTagKeyCondition	NSDictionary	通知条件検索のレスポンス情報 (※2)
2	エリア情報検索のレスポンス情報	GFIResponseTagKeyArea	NSDictionary	エリア情報検索のレスポンス情報 (※3)

(※1)：KeyはGeoFenceManager.hに定義された文字列定数を使用します。

(※2)：通知条件検索のレスポンス情報の形式を表 6に示します。

(※3)：エリア情報検索のレスポンス情報の形式を表 7に示します。

表 6. 通知条件検索のレスポンス情報

No.	パラメータ概要	Key (※1)	形式	説明
1	通知条件ID	GFIResponseKeyNid	NSNumber (int)	通知条件のID。
2	エリアID	GFIResponseKeyAid	NSString	通知対象のエリアID
3	通知タイミング	GFIResponseKeyNtm g	NSNumber (int)	1:IN 2:OUT
4	通知曜日	GFIResponseKeyNwk	NSString	エリア内に入った際の、通知対象の曜日。 日曜日～土曜日を、0～6で表す。複数の場合はカンマ区切り。 0:日 1:月 2:火 3:水 4:木 5:金 6:土
5	通知開始時間	GFIResponseKeyNst m	NSString	1日のうちで、エリア内に入った際の、通知を開始する時間。(HHmm形式)
6	通知終了時間	GFIResponseKeyNet m	NSString	1日のうちで、エリア内に入った際の、通知を終了する時間。(HHmm形式)

No.	パラメータ概要	Key (※1)	形式	説明
7	通知開始期間	GFIResponseKeyNsvd	NSDate	エリア内に入った際の、通知を開始する日時。 (yyyyMMddHHmm形式)
8	通知終了期間	GFIResponseKeyNevd	NSDate	エリア内に入った際の、通知を終了する日時。 (yyyyMMddHHmm形式)
9	再通知禁止日数	GFIResponseKeyNotnd	NSString	再通知の禁止を行う今日を含めた日数。

(※1)：KeyはGeoFenceManager.hに定義された文字列定数を使用します。

表 7. エリア情報検索のレスポンス情報

No.	パラメータ概要	Key (※1)	形式	説明
1	エリアID	GFIResponseKeyAid	NSString	通知対象のエリアID
2	ポリゴン情報	GFIResponseKeyPinf	NSString	ジオフェンスとなる情報。
3	中心緯度	GFIResponseKeyLat	NSNumber (double)	通知条件毎の固有エリアを示す中心緯度。
4	中心経度	GFIResponseKeyLon	NSNumber (double)	通知条件毎の固有エリアを示す中心経度。
5	半径(m)	GFIResponseKeyRad	NSNumber (int)	通知条件毎の固有エリアを示す半径(m)。
6	IN精度閾値	GFIResponseKeyInlv	NSNumber (int)	エリア内に入った際に、判定に利用可能と判断する位置情報の精度。 0:NON(不明) 1:レベル 1 (およその位置情報。誤差範囲：約300m以上) 2:レベル 2 (比較的正確な位置情報。誤差範囲：約300m未満) 3:レベル 3 (ほぼ正確な位置情報。誤差範囲：約50m未満)。
7	IN確定回数閾値	GFIResponseKeyInfn um	NSNumber (int)	エリア内に入ったと判定する位置情報の通知回数。
8	OUT精度閾値	GFIResponseKeyOutlv	NSNumber (int)	エリア外に出た際に、判定に利用可能と判断する位置情報の精度。 0:NON(不明) 1:レベル 1 (およその位置情報。誤差範囲：約300m以上) 2:レベル 2 (比較的正確な位置情報。誤差範囲：約300m未満) 3:レベル 3 (ほぼ正確な位置情報。誤差範囲：約50m未満)
9	OUT確定回数閾値	GFIResponseKeyOutfn um	NSNumber (int)	エリア外に出たと判定する位置情報の通知回数。

No.	パラメータ概要	Key (※1)	形式	説明
10	有効期間(開始)	GFIResponseKeyVst	NSDate	エリアの有効期間(開始)
11	有効期間(終了)	GFIResponseKeyVet	NSDate	エリアの有効期間(終了)

(※1)：KeyはGeoFenceManager.hに定義された文字列定数を使用します。

### 5.2.5. Swiftを用いている場合

ライブラリのバージョン3.0.0にて、SwiftコードからcheckLocationメソッドを呼び出す時の実装方法が変更となりました。

バージョン2.2.0以前のライブラリを利用している場合、以下を参照し、移行を行なってください。

#### <移行手順>

- checkLocationメソッドでエラーが発生した場合、例外をスローせず、引数のerrorにエラー内容をセットするようにしました。引数のerrorがnilかどうかでエラー有無を判断してください。
- メソッドの戻り値がnilでも例外（Foundation.\_GenericObjCError）はスローされません。戻り値がnilかどうかでチェック結果を判断してください。
- checkLocationメソッドが例外をスローすることはなくなりましたので、メソッドを囲っていたdo-try〜catchは不要となります。

#### 【ライブラリバージョン2.2.0以前の実装イメージ】

```
do {
    result = try GeoFenceManager.default()?.checkLocation(...)
    // 正常処理 (resultの値がnil以外の場合)
} catch let error {
    let err = (error as NSError)
    if err.code == 0 && err.domain == "Foundation._GenericObjCError" {
        // 正常処理 (resultの値がnilの場合)
    } else {
        // エラー処理
    }
}
```

#### 【ライブラリバージョン3.0.0以降の実装イメージ】

```
var error: NSError?
let result = GeoFenceManager.default()?.checkLocation(..., error:
&error)
if (error != nil) {
    // エラー処理
} else if (result == nil) {
    // 正常処理 (該当なし.または必須パラメータなし)
} else {
```

```
// 正常処理（該当あり）  
}
```

### 5.2.6. コメント

- 各パラメータに関して、必須パラメータがない場合は内外判定に失敗し、コンソールログを出力します。コンソールログは、「[checkLocation]エラー内容」という形式で出力します。

## 5.3. resetGeoFenceStatus

### 5.3.1. 説明

エリア滞在状態と、通知状態をリセットします。

### 5.3.2. プロトタイプ宣言

```
- (void)resetGeoFenceStatus;
```

### 5.3.3. パラメータ

なし

### 5.3.4. 戻り値

なし

### 5.3.5. コメント

- リセットによりすべてのエリアが未通知と同等の状態となるため、以降の測位で再度通知条件を満たした場合は通知を行います。(再通知禁止期間が設定されているエリアを含む)

## 5.4. addNotificationConditionInformation:

### 5.4.1. 説明

通知条件を追加します。

### 5.4.2. プロトタイプ宣言

```
- (NSArray *)addNotificationConditionInformation:(NSArray  
*)informationList;
```

### 5.4.3. パラメータ

informationList [I] :追加する通知条件(辞書形式)をリストで指定します。

辞書形式のKeyと形式を表 8に示します。

※nil設定不可

表 8. 通知条件リスト

No.	パラメータ概要	Key (※1)	形式	必須	説明
1	通知条件ID	GFLResponseKeyNid	NSNumber (int)	○	通知条件のID。
2	エリアID	GFLResponseKeyAid	NSString	○	通知対象のエリアID
3	通知タイミング	GFLResponseKeyNtmg	NSNumber (int)	○	1:IN 2:OUT
4	通知曜日	GFLResponseKeyNwk	NSString	-	エリア内に入った際の、通知対象の曜日。 日曜日～土曜日を、0～6で表す。複数の場合はカンマ区切り。 0:日 1:月 2:火 3:水 4:木 5:金 6:土 省略した場合、すべての曜日で有効になります。
5	通知開始時間	GFLResponseKeyNstm	NSString	-	1日のうちで、エリア内に入った際の、通知を開始する時間。(HHmm形式) 省略した場合、通知開始時間の判断は行いません。



No.	パラメータ概要	Key (※1)	形式	必須	説明
6	通知終了時間	GFIResponseKeyNetm	NSString	-	1日のうちで、エリア内に入った際の、通知を終了する時間。(HHmm形式) 省略した場合、通知終了時間の判断は行いません。
7	通知開始期間	GFIResponseKeyNsvd	NSDate	-	エリア内に入った際の、通知を開始する日時。 (yyyyMMddHHmm形式) 省略した場合、通知開始時間の判断は行いません。
8	通知終了期間	GFIResponseKeyNevd	NSDate	-	エリア内に入った際の、通知を終了する日時。 (yyyyMMddHHmm形式) 省略した場合、通知終了期間の判断は行いません。
9	再通知禁止日数	GFIResponseKeyNotnd	NSString	-	再通知の禁止を行う今日を含めた日数。 指定する場合、1以上の整数を設定してください。 省略した場合、再通知禁止日数の判断は行いません。

(※1)：KeyはGeoFenceManager.hに定義された文字列定数を使用します。

#### 5.4.4. 戻り値

追加できなかった通知条件の辞書（引数で与えられたもの）をリストで返します。  
すべて追加できた場合、空の配列を返します。

#### 5.4.5. コメント

- 各パラメータに関して、必須パラメータがない場合、型が異なる場合、または説明欄に記載の範囲以外の値を入力した場合は、追加失敗しコンソールログを出力します。コンソールログは、「[addNotificationConditionInformation]エラー内容」という形式で出力します。

## 5.5. removeNotificationConditionInformation:

### 5.5.1. 説明

通知条件を削除します。

### 5.5.2. プロトタイプ宣言

```
- (NSArray *)removeNotificationConditionInformation:(NSArray  
*)informationIDList;
```

### 5.5.3. パラメータ

informationIDList [I] :削除する通知条件IDをNSNumberのリストで指定します。  
※nil設定不可

### 5.5.4. 戻り値

削除した通知条件IDをリストで返します。

### 5.5.5. コメント

- 各パラメータに関して、必須パラメータがない場合は削除に失敗し、logcatにコンソールログを出力します。コンソールログは、「[removeNotificationConditionInformation]エラー内容」という形式で出力します。

## 5.6. notificationConditionInformationList:

### 5.6.1. 説明

設定されている通知条件を返します。

### 5.6.2. プロトタイプ宣言

```
- (NSArray *)notificationConditionInformationList:(NSArray  
*)informationIDList;
```

### 5.6.3. パラメータ

informationIDList [I] :取得する通知条件IDをNSNumberのリストで指定します。  
nilがセットされた時は、すべての通知条件IDを返します。

### 5.6.4. 戻り値

通知条件の辞書をリストで返します。  
通知条件がない場合は、空のリストを返します。

### 5.6.5. コメント

なし

## 5.7. addAreaInformation:

### 5.7.1. 説明

エリア情報を追加します。

### 5.7.2. プロトタイプ宣言

```
- (NSArray *)addAreaInformation:(NSArray *)informationList;
```

### 5.7.3. パラメータ

informationList [I] :追加するエリア情報(辞書形式)のリスト。  
辞書形式のKeyと形式を表 9に示します。  
※nil設定不可

表 9. エリア情報リスト

No.	パラメータ概要	Key (※1)	形式	必須	説明
1	エリアID	GFIResponseKeyAid	NSString	○	通知対象のエリアID
2	ポリゴン情報(※2)	GFIResponseKeyPin f	NSString	△	ジオフェンスとなる情報。(※3) 省略する場合は、中心緯度、中心経度、半径を指定してください。
3	中心緯度(※2)	GFIResponseKeyLat	NSNumber (double)	△	通知条件毎の固有エリアを示す中心緯度。 省略する場合は、ポリゴン情報を指定してください。
4	中心経度(※2)	GFIResponseKeyLon	NSNumber (double)	△	通知条件毎の固有エリアを示す中心経度。 省略する場合は、ポリゴン情報を指定してください。
5	半径(m) (※2)	GFIResponseKeyRad	NSNumber (int)	△	通知条件毎の固有エリアを示す半径(m)。 省略する場合は、ポリゴン情報を指定してください。

No.	パラメータ概要	Key (※1)	形式	必須	説明
6	IN精度閾値 (※4)	GFIResponseKeyInlv	NSNumber (int)	-	エリア内に入った際に、判定に利用可能と判断する位置情報の精度。 0:NON(どんな精度でも利用可能と判断します) 1:レベル1(おおよその位置情報。) 2:レベル2(比較的正確な位置情報。誤差範囲：約300m未満) 3:レベル3(ほぼ正確な位置情報。誤差範囲：約50m未満) 省略した場合、どんな精度でも利用可能と判断します。
7	IN確定回数閾値	GFIResponseKeyInfnum	NSNumber (int)	○	エリア内に入ったと判定する位置情報の通知回数。 1以上を設定してください。
8	OUT精度閾値 (※4)	GFIResponseKeyOutlv	NSNumber (int)	-	エリア外に出た際に、判定に利用可能と判断する位置情報の精度。 0:NON(どんな精度でも利用可能と判断します) 1:レベル1(おおよその位置情報。) 2:レベル2(比較的正確な位置情報。誤差範囲：約300m未満) 3:レベル3(ほぼ正確な位置情報。誤差範囲：約50m未満) 省略した場合、どんな精度でも利用可能と判断します。
9	OUT確定回数閾値	GFIResponseKeyOutfnum	NSNumber (int)	○	エリア外に出たと判定する位置情報の通知回数。 1以上を設定してください。
10	有効期間(開始)	GFIResponseKeyVst	NSDate	-	エリアの有効期間(開始) 省略した場合、開始有効期間の判断を行いません。
11	有効期間(終了)	GFIResponseKeyVet	NSDate	-	エリアの有効期間(終了) 省略した場合、終了有効期間の判断を行いません。

(※1)：KeyはGeoFenceManager.hに定義された文字列定数を使用します。

(※2)：ポリゴンと中心緯度、中心経度、半径のすべてが指定された場合は、追加されません。どちらかの指定のみにして下さい。

(※3)：設定するポリゴン情報は以下の形式で設定してください。

例：シングルポリゴン（ポリゴンが1つ）

MULTIPOLYGON( ( ( 130.10 33.26,131.28 33.302,132.033 31.2973,129.528 31.44,130.10 33.26 ) ) ) ※最低4組の緯度経度が必要です

例：マルチポリゴン（ポリゴンが2つ）

```
MULTIPOLYGON( ( ( 130.10 33.26,131.28 33.302,132.033 31.2973,129.528 31.44,130.10 33.26 ) ),( ( 132.561 34.3615,134.846 34.2889,134.846 33.2662,132.385 32.7133,132.561 34.3615 ) ) )
```

例：ドーナツ型ポリゴン（マルチポリゴンと同等）

```
MULTIPOLYGON( ( ( 139.0211004 36.0713996, 139.0183039 34.8450133, 140.8528174 34.8268591, 140.8385415 36.1670114, 139.0211004 36.0713996 ),(139.7570529 35.6860405, 139.7798065 35.68661, 139.7798072 35.6689764, 139.7591868 35.6692602, 139.7570529 35.6860405) ) )
```

複数のポリゴンがある場合はマルチポリゴンかドーナツ型ポリゴンに関わらず、以下の図のようにポリゴン同士が重なっている箇所はエリア外となる。

(※4)：レベル1～3では精度が1以上であれば利用可能と判断します。

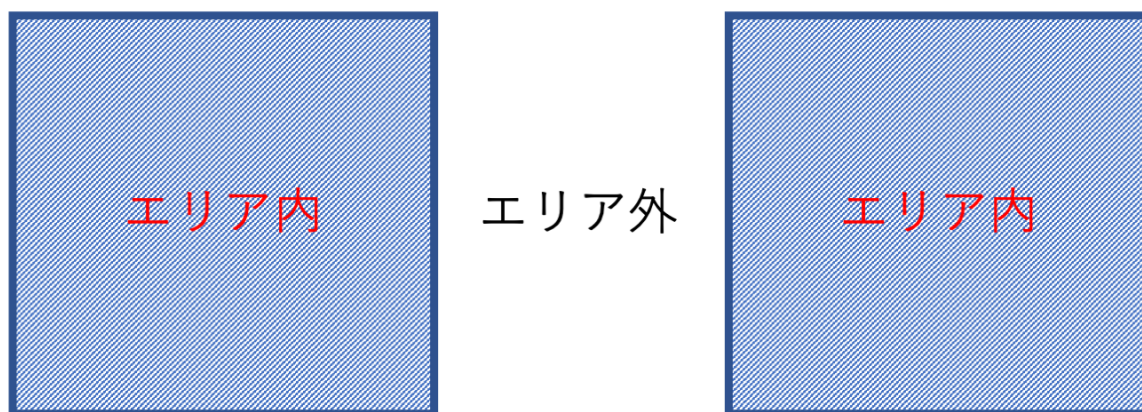


図2. ポリゴンが重なっていない場合

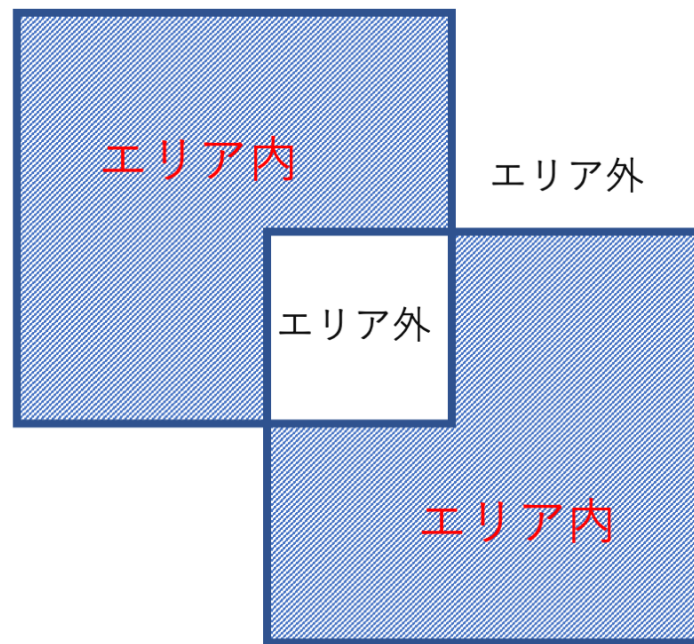


図3. ポリゴンの一部が重なっている場合

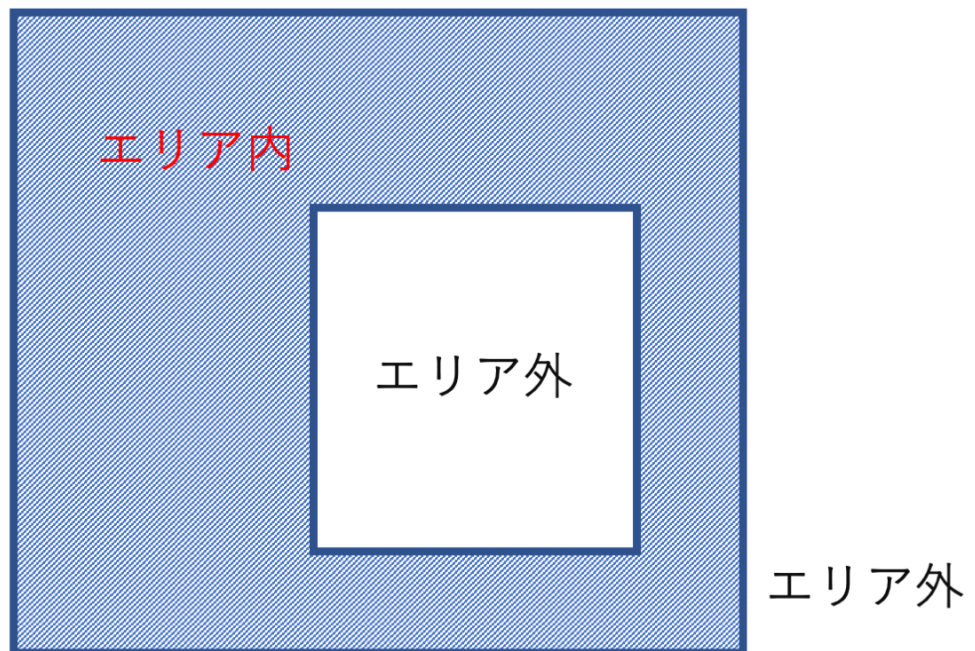


図4. ポリゴン内にポリゴンがある場合

#### 5.7.4. 戻り値

追加できなかったエリア情報の辞書(引数で与えられたもの)をリストで返します。  
すべて追加できた場合は、空の配列を返します。

### 5.7.5. コメント

- 各パラメータに関して、必須パラメータがない場合、または型が異なる場合、または説明欄に記載の範囲以外の値を入力した場合は、追加失敗しコンソールログを出力します。コンソールログは、「[addAreaInformation]エラー内容」という形式で出力します。



## 5.8. addAreaInformationForGeoJson:

### 5.8.1. 説明

GeoJSON形式のエリア情報を使用して、エリア情報を追加します。

### 5.8.2. プロトタイプ宣言

```
- (NSArray *)addAreaInformationForGeoJson:(NSArray *)informationList;
```

### 5.8.3. パラメータ

informationList [I] :追加するエリア情報(辞書形式)のリスト。  
辞書形式のKeyと形式を表 10に示します。  
※nil設定不可

表 10. エリア情報リスト

No.	パラメータ概要	Key (※1)	形式	必須	説明
1	エリアID	GFIResponseKeyAid	NSString	○	通知対象のエリアID
2	GeoJSON情報(※2)	GFIResponseKeyGin f	String	○	ジオフェンスとなる情報。(※3)
3	IN精度閾値(※4)	GFIResponseKeyInlv	NSNumber (int)	-	エリア内に入った際に、判定に利用可能と判断する位置情報の精度。 0:NON(どんな精度でも利用可能と判断します) 1:レベル 1 (おおよその位置情報。) 2:レベル 2 (比較的正確な位置情報。誤差範囲：約300m未満) 3:レベル 3 (ほぼ正確な位置情報。誤差範囲：約50m未満) 省略した場合、どんな精度でも利用可能と判断します。
4	IN確定回数閾値	GFIResponseKeyInf num	NSNumber (int)	○	エリア内に入ったと判定する位置情報の通知回数。 1以上を設定してください。

No.	パラメータ概要	Key (※1)	形式	必須	説明
5	OUT精度閾値 (※4)	GFIResponseKeyOutlv	NSNumber (int)	-	エリア外に出た際に、判定に利用可能と判断する位置情報の精度。 0:NON(どんな精度でも利用可能と判断します) 1:レベル1(おおよその位置情報。) 2:レベル2(比較的正確な位置情報。誤差範囲：約300m未満) 3:レベル3(ほぼ正確な位置情報。誤差範囲：約50m未満) 省略した場合、どんな精度でも利用可能と判断します。
6	OUT確定回数 閾値	GFIResponseKeyOutfnum	NSNumber (int)	○	エリア外に出たと判定する位置情報の通知回数。 1以上を設定してください。
7	有効期間(開始)	GFIResponseKeyVst	NSDate	-	エリアの有効期間(開始) 省略した場合、開始有効期間の判断を行いません。
8	有効期間(終了)	GFIResponseKeyVet	NSDate	-	エリアの有効期間(終了) 省略した場合、終了有効期間の判断を行いません。

(※1)：KeyはGeoFenceManager.hに定義された文字列定数を使用します。

(※2)：設定するGeoJSON情報は以下の形式で設定してください。

- 例1：円

GeoJSONの標準仕様には円を表す形式がないためGeoJSONのpropertiesを利用して円を表す。下記のようにpropertiesの中にradius(半径)をメートル単位で記述すること（小数点付き数値を指定した場合、小数点以下は切り捨てられ、整数として扱われます）。geometryのtypeはPointにすること。coordinatesには経度、緯度の順番で座標を入れることに注意すること。

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [-122.126986, 47.639754]
  },
  "properties": {
    "radius": 100
  }
}
```

- 例2：ポリゴン

ポリゴンはGeoJSONの標準仕様の形式で設定可能。typeはPolygon。MultiPolygonは対象外。

ド

ーナツ型などの内側に穴がある形状も対象外。 1点目の座標と最終の座標は同じ座標にすること。  
coordinatesには経度、緯度の順番で座標を入れることに注意すること。

```
{
  "type": "Feature",
  "geometry": {
    "type": "Polygon",
    "coordinates": [[[5,25],[14,25],[14,29],[5,29],[5,25]]]
  }
}
```

(※4)：レベル1～3では精度が1以上であれば利用可能と判断します。

#### 5.8.4. 戻り値

追加できなかったエリア情報の辞書(引数で与えられたもの)をリストで返します。  
すべて追加できた場合は、空の配列を返します。

#### 5.8.5. コメント

- 各パラメータに関して、必須パラメータがない場合、または型が異なる場合、または説明欄に記載の範囲以外の値を入力した場合は、追加失敗しコンソールログを出力します。コンソールログは、「[addArealInformationForGeoJson]エラー内容」という形式で出力します。

## 5.9. removeAreaInformation:

### 5.9.1. 説明

エリア情報を削除します。

### 5.9.2. プロトタイプ宣言

```
- (NSArray *)removeAreaInformation:(NSArray *)areaIDList;
```

### 5.9.3. パラメータ

areaIDList [I] :削除するエリアID(NSString)のリスト

※nil設定不可

### 5.9.4. 戻り値

削除したエリアIDをリストで返します。

### 5.9.5. コメント

- 各パラメータに関して、必須パラメータがない場合は削除に失敗し、logcatにコンソールログを出力します。コンソールログは、「[removeAreaInformation]エラー内容」という形式で出力します。

## 5.10. areaInformationList:

### 5.10.1. 説明

設定されているエリア情報を返します。

### 5.10.2. プロトタイプ宣言

```
- (NSArray *)areaInformationList:(NSArray *)areaIDList;
```

### 5.10.3. パラメータ

areaIDList [I] :取得するエリアIDのリスト。

nilが指定された場合、すべてのエリア情報を返します。

### 5.10.4. 戻り値

エリア情報の辞書をリストで返します。

エリア情報がない場合は空の配列を返します。

### 5.10.5. コメント

なし

## 5.11. clearAllInformation

### 5.11.1. 説明

通知条件、エリア情報をすべて削除します。

### 5.11.2. プロトタイプ宣言

```
- (void)clearAllInformation;
```

### 5.11.3. パラメータ

なし

### 5.11.4. 戻り値

なし

### 5.11.5. コメント

保持しているすべての通知条件、エリア情報を削除します。

特定の情報のみ削除したい場合は

[removeAreaInformation](#) または  
[removeNotificationConditionInformation](#) を使用してください。

## 5.12. libraryVersion

### 5.12.1. 説明

ライブラリのバージョンを文字列で取得します。

### 5.12.2. プロトタイプ宣言

```
- (NSString *)libraryVersion;
```

### 5.12.3. パラメータ

なし

### 5.12.4. 戻り値

文字列でバージョンが戻されます。