Ant Technology

Location Based Service
User Guide

Document Version: 20231226

蚂蚁集团
ANT GROUP

Legal disclaimer

Trademark statement

and other trademarks related to Ant Group are owned by Ant Group. The third-party registered trademarks involved in this document are owned by the right holder according to law.

Disclaimer

The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Ant Group reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through channels authorized by Ant Group. You must pay attention to the version changes of this document as they occur and download and obtain the latest version of this document from Ant Group's authorized channels. Ant Group does not assume any responsibility for direct or indirect losses caused by improper use of documents.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Location Based Service

1.1. About Location Based Service

mPaaS provides a positioning component to support Location Based Service. The positioning component provides the following capabilities by encapsulating system API operations:

- Obtain the latitude and longitude of the location of the current device in an easy way.
- Obtain the time when the latitude and longitude is obtained and more accuracy information.
- Cache data and convert coordinates.
- Hook the calling of all system API operations and standardize the positioning process.

> **Note**
>
> Currently, the function of querying reverse geocoding information based on latitude and longitude is temporarily not supported.

1.2. Integrate Android SDK

The location-based services (LBS) SDK is a set of simple LBS interfaces. You can use this set of LBS API to obtain positioning results.

The LBS component can be integrated to the mPaaS in the **native AAR mode** and the **component mode**.

Prerequisites

- If you want to integrate the component to the mPaaS based on the native AAR mode, you need to first complete the prerequisites and the subsequent steps. For more information, see Add mPaaS to your project
- If you want to integrate the component to the mPaaS based on components, you need to first complete the Component-based integration procedure.

Add the SDK

Native AAR mode

In your project, install the **LBS** component on the **Component Management (AAR)** page. For more information, see Manage component dependencies in the native AAR mode.

Component-based mode

In your Portal and Bundle projects, install the **LBS** component on the **Component Management** page.

For more information, see Manage component dependencies.

Apply for an AMAP key

Before you use the LBS, go to the AMAP Open Platform to apply for an account and obtain the LBS key. The following figure shows an example of a key.

Key名称	Key	绑定服务	操作
mPaaSDemoKey	27a6196e434070dfaa75ea9976c06040	Android平台	设置 删除

Configure the AndroidManifest.xml file

Add the AMAP positioning key and AMAP positioning service to the `AndroidManifest.xml` file.

```
<!--The AMAP positioning key-->
 <meta-data
       android:name="com.amap.api.v2.apikey"
       android:value="The AMAP positioning key that you applied for" />
 <!--The AMAP positioning service-->
 <service android:name="com.amap.api.location.APSService"></service>
```

For the baseline version 10.1.68.18 and later, the auto check-in feature is disabled by default. To enable this feature, add the following code to the `AndroidManifest.xml` file:

```
<meta-data android:name="com.mpaas.lbs.autoCheckIn" android:value="true" />
```

API description

- Call the positioning service

```
LBSLocationManagerProxy.getInstance().requestLocationUpdates(MainActivity.this, new L
BSLocationListener() {
          @Override
          public void onLocationUpdate(LBSLocation lbsLocation) {
              Toast.makeText(MainActivity.this, "lbsLocation is " + lbsLocation.getAdd
ress(), Toast.LENGTH_LONG).show();
          }
          @Override
          public void onLocationFailed(int i) {
              Toast.makeText(MainActivity.this,
                      "onLocationFailed" + i, Toast.LENGTH_SHORT).show();
          }
      });
```

- Call other API operations

```
// Register the location listener.
public void requestLocationUpdates(Context context, LBSLocationListener
locationListener)
// Remove the callback of the location monitoring registration.
public void removeUpdates(Context context, LBSLocationListener listener)
// Obtain the location of the latest successful positioning.
public LBSLocation getLastKnownLocation(Context context)
```

1.3. Integrate iOS SDK

The Location-based Service (LBS) SDK is a set of simple LBS interfaces. You can use this set of LBS API to obtain positioning results.

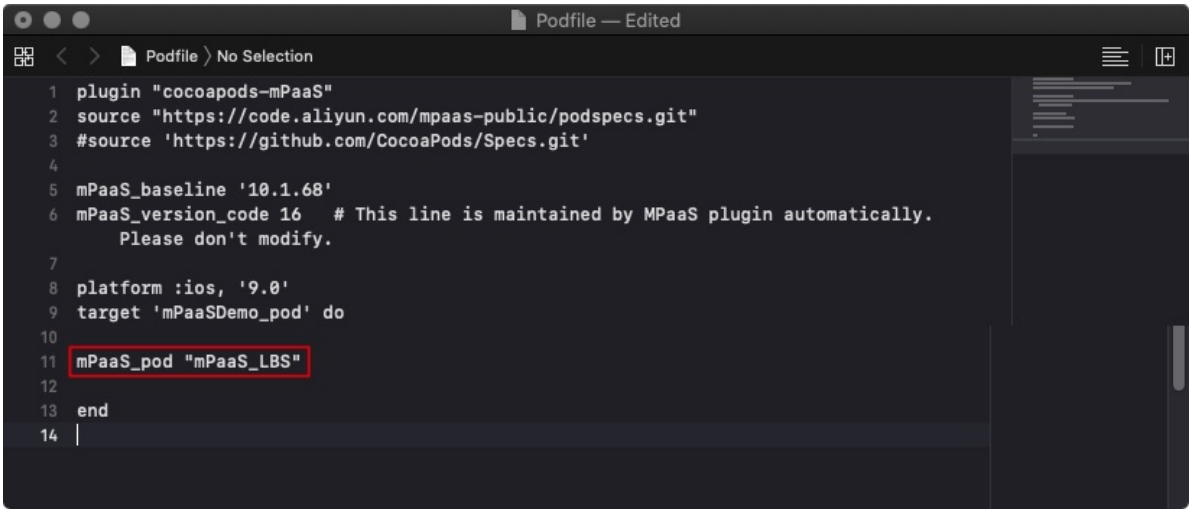
Prerequisite

The project already gets access to mPaaS. For more information, see the following content: Integrate mPaaS based on an existing project and CocoaPods.

Add the SDK

Use CocoaPods plugin to add the Location-based Service SDK. Complete the following steps:

1. In the Podfile file, use `mPaaS_pod "mPaaS_LBS"` to add mobile gateway component dependencies.



2. In the terminal, run `pod install` to complete integration.

Use the SDK

This topic describes how to use the LBS SDK in baseline 10.1.32 later versions on the official LBS demo.

The APMobileLBS module provides a method for obtaining the latitude and longitude of the current location.

Note: The LBS does not support reverse geographic query. To use reverse geocoding, you can call the AMAP API.

API description

To learn LBS API operations, see the following code and the parameter descriptions in the comments.

Use MPLBSConfiguration to set parameters

```
/**
 Configurations of the LBS
 */
@interface MPLBSConfiguration : NSObject

/** The expected precision of a single positioning operation, in meters. We recommend t
hat you specify an acceptable positive number based on the business scenario. For examp
le, set the parameter to 500 to indicate a range within 500 m. */
@property (nonatomic, assign) CLLocationAccuracy desiredAccuracy;

/** The time for caching received data of a single positioning operation, counting from
the current time backward. We recommend that you specify a time longer than 30s. */
@property (nonatomic, assign) APCoreLocationCacheAvaliable cacheTimeInterval;

/** The timeout interval for a single positioning operation or reverse geographical que
ry, in seconds. Default value: 2s. Minimum value: 2s. */
@property (nonatomic, assign) NSTimeInterval timeOut;

/** The level of the information obtained through reverse geographical query. Default v
alue: APCoreLocationReGeoLevelDistrict. */
@property (nonatomic, assign) LBSLocationReGeoLevel reGeoLevel;

/** The location information obtained through reverse geographical query based on the l
atitude and longitude. */
@property (nonatomic, strong) CLLocation *reGeoLocation;

/** Specifies whether coordinates used for reverse geographical query are coordinates i
n AMAP. Default value: YES. This parameter is valid only when the reGeoLocation paramet
er is specified. */
@property (nonatomic, assign) BOOL reGeoCoordinateConverted;

/** Specifies whether to enable check-in. Default value: NO. If check-in is required, s
et this parameter to YES. */
@property (nonatomic, assign) BOOL needCheckIn;

/**
 * Specifies whether high-precision positioning is required. This parameter is not requ
ired for versions earlier than iOS 14. For iOS 14 and later, this parameter is set to
NO (low precision) by default. If high-precision positioning is required, the business
personnel must modify this parameter.
 */
@property (nonatomic,assign) BOOL highAccuracyRequired;

@end
```

Use MPLBSLocationManager to initiate a positioning request

```
/**
 Callback blocks that are triggered based on positioning results

 @param success Whether positioning is successful
 @param locationInfo Location information
 @param error Error information of positioning failure
 */
typedef void(^MPLBSLocationCompletionBlock)(BOOL success,
                                            MPLBSLocationInfo *locationInfo,
                                            NSError *error);

/**
 LBS
 */
@interface MPLBSLocationManager : NSObject

/**
 Initialize

 @param configuration Parameter settings
 @return Instance
 */
- (instancetype)initWithConfiguration:(MPLBSConfiguration *)configuration;

/**
 Initiate a single positioning operation

 @param needReGeocode Indicates whether reverse geographic query is required. The LBS d
oes not support reverse geographic query. You must set this parameter to NO.
 @param block The callback block that is triggered after positioning ends.
 */
- (void)requestLocationNeedReGeocode:(BOOL)needReGeocode
                   completionHandler:(MPLBSLocationCompletionBlock)block;
```

MPLBSLocationInfo in callbacks

```
/**
 Reverse geographic information
 */
@interface MPLBSReGeocodeInfo : NSObject

@property (nonatomic, strong) NSString* country;        // The country.
@property (nonatomic, strong) NSString* countryCode;    // The country code.
@property (nonatomic, strong) NSString* province;      // The province.
@property (nonatomic, strong) NSString* city;           // The city.
@property (nonatomic, strong) NSString* district;       // The district.
@property (nonatomic, strong) NSString* street;         // The street.
@property (nonatomic, strong) NSString* streetCode;     // The street code.
@property (nonatomic, strong) NSString* cityCode;       // The city code.
@property (nonatomic, strong) NSString* adCode;         // The administrative district
code.
@property (nonatomic, strong) NSArray* poiList;         // The POI information list.

@end

/**
 Data structure of location information in the positioning result
 */
@interface MPLBSLocationInfo : NSObject

@property (nonatomic, strong) CLLocation* location;         // The location
information.
@property (nonatomic, strong) MPLBSReGeocodeInfo* rgcInfo;  // The reverse geographic i
nformation.

@end
```

Sample code

```
- (void)getLocation {
    MPLBSConfiguration *configuration = [[MPLBSConfiguration alloc] init];
    configuration.desiredAccuracy = kCLLocationAccuracyBest;

    self.locationManager = [[MPLBSLocationManager alloc]
initWithConfiguration:configuration];
    [self.locationManager requestLocationNeedReGeocode:NO completionHandler:^(BOOL succ
ess, MPLBSLocationInfo * _Nonnull locationInfo, NSError * _Nonnull error) {
        NSString *message;
        if (success) {
            message = [NSString stringWithFormat:@"Positioning success. Longitude:
%.5f. Latitude: %.5f. Precision: %.3f. Whether high precision is required: %d", locatio
nInfo.location.coordinate.longitude, locationInfo.location.coordinate.latitude, locatio
nInfo.location.horizontalAccuracy,
!locationInfo.location.ap_lbs_is_high_accuracy_close];
        } else {
            message = [NSString stringWithFormat:@"%@", error];
        }
        dispatch_async(dispatch_get_main_queue(), ^{
            AUNoticeDialog *alert = [[AUNoticeDialog alloc] initWithTitle:@"Positioning
result" message:message delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];
            [alert show];
        });
    }];
}
```

Adaptation in iOS 14

In iOS 14, precise location is a permission option. You can turn on the option when you apply for positioning permissions. On the positioning permissions setting page, you can adjust the setting.

Adaptation of input parameters

In MPLBSConfiguration, add the highAccuracyRequired parameter. If `highAccuracyRequired = YES` is passed in, but high-precision positioning is disabled, a callback error occurs.

```
/**
 Configurations of the LBS
 */
@interface MPLBSConfiguration : NSObject

/**
 *  Specifies whether high-precision positioning is required. This parameter is not req
uired for versions earlier than iOS 14. For iOS 14 and later, this parameter is set to
NO (low precision) by default. If high-precision positioning is required, the business
personnel must modify this parameter.
 */
@property (nonatomic,assign) BOOL highAccuracyRequired;

@end
```

```
//If highAccuracyRequired = YES is passed in, but high-precision positioning is disabled, a callback error occurs.
Errorcode: APCoreLocationErrorCodeHighAccuracyAuthorization
```

Callback adaptation

If `highAccuracyRequired = NO` is passed in or high-precision positioning is not specified, the callback object CLLocation contains the `ap_lbs_is_high_accuracy_close` field. This field indicates whether high-precision positioning is disabled.

```
// Reconstructs the output parameter.
@interface CLLocation (APMobileLBS)
/*
 *  Specifies whether to disable high-precision positioning. Default value: NO
 */
@property(nonatomic,assign)BOOL ap_lbs_is_high_accuracy_close;
@end
```

Sample code

```
- (void)getLocationWithHighAccuracy {
    MPLBSConfiguration *configuration = [[MPLBSConfiguration alloc] init];
    configuration.desiredAccuracy = kCLLocationAccuracyBest;
    configuration.highAccuracyRequired = YES;

    self.locationManager = [[MPLBSLocationManager alloc]
initWithConfiguration:configuration];
    [self.locationManager requestLocationNeedReGeocode:NO completionHandler:^(BOOL success, MPLBSLocationInfo * _Nonnull locationInfo, NSError * _Nonnull error) {
        NSString *message;
        if (success) {
            message = [NSString stringWithFormat:@"Positioning success. Longitude: %.5f. Latitude: %.5f. Precision: %.3f. Whether high precision is required: %d", locationInfo.location.coordinate.longitude, locationInfo.location.coordinate.latitude, locationInfo.location.horizontalAccuracy,
!locationInfo.location.ap_lbs_is_high_accuracy_close];
        } else {
            message = [NSString stringWithFormat:@"%@", error];
        }
        dispatch_async(dispatch_get_main_queue(), ^{
            AUNoticeDialog *alert = [[AUNoticeDialog alloc] initWithTitle:@"Positioning result" message:message delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];
            [alert show];
        });
    }];
}
```

1.4. Code sample

Android code sample

See Get code sample to obtain code sample and its usage instructions and attentions.

iOS code sample

See Get code sample to obtain code sample and its usage instructions and attentions.

Get started

1. Run the program, and input the parameter values of LBS interface:
   - Business scenario ID: The type of the current business, required. We recommend that you pass in class name for native project while URL for HTML5.
   - Location accuracy: In meters. It is suggested to pass in an acceptable positive number, such as 500, which means the accuracy is within 500 meters.
   - Cache time: Time period in which the cache is valid. It is recommended to set the cache time to 30 seconds or more.
   - Timeout length: The time length before locating timeout, in seconds. Both the default value and the minimum value are 2 seconds.
2. Tap **Locate** to get the positioning result.
3. Tap **reset** on the top-right corner to reset parameters.