

This repository

Search

Pull requestsIssuesGist

predict-io / PredictIO-Android

Watch

10

Star

11

Fork

2

<> Code

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

📶 Pulse

📊 Graphs

FAQ

ahmadarbisoft edited this page on 27 Oct 2016 · 4 revisions

About predict.io

General information

The information below provides you with links and explanations to help you find your way around the predict.io SDK Integration guides and how to test your SDK.

In-App callbacks

Once your App (with predict.io) is started, it will continuously monitor user activities (of course that is battery optimised). Whenever predict.io detects any event (started/ended journey etc) it will inform your App using callbacks. i.e Interface in Android. Here are more details on how to work with [Interface](#).

Testing

To make sure that you have integrated correctly our SDK before going live please follow the instructions below to test your implementation before (and after) you submit the app to the play store. This is very important.

Always do field tests, and choose your locations wisely:

- From open skies to crowded sky scrapers
- With limited to no GPS reception
- Change heights during the tests
- Choose different times of the day, weather conditions and locations
- Test most scenarios, from driving real slow to fast and traffic jams
- Test around the device’s power up time, especially for a device that was powered off for days and/or with no internet connectivity
- Use different device settings and different devices

What can I do with this SDK?

predict.io allows you to retrieve real-time updates on the mobility status of a user. You can run the API in Android apps. No additional hardware is required.

Main features include detecting when a user:

- has arrived at a destination.
- has just started a journey.

You may use this information to trigger events or notifications specific to your use case.

What is predict.io?

Our mobile SDK gives you battery friendly background location. So that you always know when and where a user arrives or departs. No fiddling with geofences. No beacons or NFC needed. It uses the sensors embedded in any modern smartphone. You can embed it in minutes. Rather than spending months with the Activity APIs yourself. Our SDK powers many industry leading Android and iOS apps in mobility, retail, hospitality, lifestyle and banking.

► Pages 11

- [Home](#)
- [Use Cases](#)
- [Gradle Integration Guide](#)
- [Manual integration Guide](#)
- [API Documentation & Usage Guide](#)
- [Upgrade Guide](#)
- [Migration Guide to predict.io 3.0](#)
- [Marshmallow Compatibility \(Android 6.0\)](#)
- [ProGuard Guide](#)
- [Using Webhooks](#)
- [FAQ](#)

Clone this wiki locally

https://github.com/predict-io/PredictIO-Android/wiki/FAQ

Using predict.io SDK

How accurate is this SDK?

The native Activity APIs are one-size-fits-all. But that is not enough for granular mobility detection. Hence we made it better. With native APIs you have a ~50% false positive rate for car trips. We cut it down to 10% for a typical urban driver.

How much battery is required to run predict.io in the background?

Our SDK consumes only ~2% of a total charge by running in the background. The same result would consume 20% of the battery on standard OS APIs. If you face any different behavior please don't hesitate to [contact us](#).

How long does it take to install this SDK?

Installing our SDK takes you 30 minutes. Building a battery optimized, normalised output for iOS and Android will cost you years. Trust us. We've been through it.

What about users privacy?

Just like you, our primary concern is protecting people's privacy. Our own [privacy policies](#) reflect just that. We have built our system so that no personal identifiable information is being collected by us. We don't have any user credentials, any unique device ID nor a clear text IP address. The only device identifier used by the system is a proprietary hash generated by us. Which is not used in any other product anywhere in the world. Hence it can not be traced back across other services. We never know who your user is. And we don't want to. We have gone through several security audits with large corporates. We are confident to be able to adhere to any of your standard, too

Which user data is collected by predict.io?

When you implement the predict.io SDK, we collect non-PII (anonymous) data of the users using the SDK. This includes events data as well as device data to improve the service e.g. regarding performance and battery consumption (see [Terms of Service](#) for more details).

Setup

How do I test integration?

To make sure that you have integrated correctly our SDK before going live please follow the instructions below to test your implementation before (and after) you submit the app to the stores. This is very important.

Always do field tests, and choose your locations wisely:

- From open skies to crowded sky scrapers
- With limited to no GPS reception
- Change heights during the tests
- Choose different times of the day, weather conditions and locations
- Test most scenarios, from driving real slow to fast and traffic jams
- Test around the device's power up time, especially for a device that was powered off for days and/or with no internet connectivity
- Use different device settings, and different devices

Is there any effect on my app?

Easy to answer! No, our SDK does not affect your app, it makes it even better.

Which permissions users need to grant?

- Internet
- Access Network State
- Access Fine location
- Access Coarse Location
- Activity Recognition
- Receive Boot Completed

Note: Android will only prompt user for “Location” permission.

Known Issues

- Use same version of [Google Play Services](#) in the whole project which predict.io SDK is using. E.g. if predict.io SDK requires 9.2.1 than use google analytics library of 9.2.1 version as well.
- Make sure your app is [Compatible With Marshmallow \(6.0\)](#) devices and prompt user for required permissions before starting tracker.
- Make sure you add [SDK Proguard Rules](#) in your project's Proguard file.
- Make sure you have given your SDK key in your manifest file and also have added required permissions under manifest tag and required services and receivers under application tag.
- Location\GPS should be enabled\ON in smartphone's settings.
- Airplane mode should be disabled\OFF in smartphone's settings, as predict.io SDK is suspended in Airplane mode for optimal battery performance.
- Device should have internet access while SDK's start() method is invoked.
- Power/Battery saving mode should be disabled\off in device settings.
- Testing predict.io SDK is not possible on any emulator.
- If you are having any issue regarding SDK integration or callbacks, then first try to test provided [Example](#) (make sure you add your API key in the example project). If issue still persist then zip and email your Example along with your API key and APK file to us.

Pricing

Check out our [pricing section](#) to select your plan.