

**FOURSQUARE**  
PILGRIM SDK

# Evaluating Location SDKs: A Primer for App Developers

---

Picking the Right SDK for Your Use Case

October 2019



**FOURSQUARE**

# Introduction

With nine out of ten smartphone users relying on location-aware apps, according to the [Pew Research Center](#), developers are well-positioned to elevate how location technology is being used to drive engagement. And

thanks to industry advancements, developers can now use location to provide more contextual app experiences than ever before; this includes everything from delivering a coupon for a latte when a user is across from her local coffee shop, to pushing a notification about a new speakeasy based on a user's past visits to cocktail bars.

But these contextual feature opportunities just scratch the surface. To unlock the full potential of location technology, developers must go beyond the relatively simple location features built into a mobile operating system (OS). Instead, they must choose more sophisticated third-party software development kits

(SDKs) that allow them to build features around users' activities, for example, when they are commuting or traveling on vacation.

But with so many location technologies available, how is an app developer able to choose what location services to use and the associated SDKs to integrate?

We've prepared this guide to help you make that choice. To find the right SDK to bring location awareness to your apps, follow our simple three step process.



## Step 1: Define Scope

We recommend that you start with a clear understanding of the features you'd like to build into your app. As you'd expect, your specific use case will determine what you'll need to test as part of the SDK selection process.

For example, it might be enough for the SDK to provide geofencing, in which notifications get triggered based on whether a user enters or exits a broad geographical boundary. Or you may need to reach users with a higher degree of accuracy, for example when they visit a competitor's storefront, which requires that an SDK have visit detection technology.

Once you've narrowed in on your use case, you should also consider what types of insights and features will elevate the contextual experience. For example, behavioral data—informed by patterns of visits to places and categories of

places—can unlock powerful features that power use cases around contextual push notifications.

Two Foursquare features—User States and SDK Segments—show how a state-of-the-art location SDK enables hyper-targeting.

- User States leverages visit patterns to allow you to infer behavior, such as when a user is at home or work, commuting during the weekday, or traveling on vacation.
- SDK Segments groups users into pre-defined buckets based on how recently and how frequently they've visited specific store chains and categories of locations.

As you can see, it's important that you do some pre-work before you begin the testing process. Understanding your app's purpose ensures that you arrive at the right set of capabilities to evaluate, allowing you to narrow down the SDKs you'll evaluate to a manageable number.

From there, it's time to install your candidate SDKs on the mobile devices you will use for testing.

## Step 2: Build an Evaluation Model

To effectively evaluate whether an SDK is right for you, it's important to collect data for assessing accuracy and performance. Here are our recommendations for factors to examine.

### **False-negative Rate**

**Question:** What percentage of venue visits did the SDK detect in comparison to others that you are evaluating?

**Pro-Tip:** Measure this as an absolute value that takes in all locations visited in testing.

### **False-positive Rate**

**Question:** What percentage of detected venue visits did not match an actual venue visit?

**Pro-Tip:** An SDK might detect a visit when, in reality, the user was only driving by the location. The false-positive rate also should be measured as an absolute value, regardless of location.

### **Precision**

**Question:** Of the detected venue visits, what percentage of them correctly matched to the right venue?

**Pro-Tip:** If you are testing visits to locations that are in close proximity, you should assess whether the SDK correctly identified the one the user actually visited.

### **Performance**

**Question:** How quickly did each SDK pick up the visit compared to other SDKs?

**Pro-Tip:** Be sure to test SDKs on the same device to ensure consistent results.

Taking all of these factors into account will give you the strongest basis for comparison between SDKs.

## Step 3: Test

Now that you know what parameters you want to consider, you're ready to proceed to testing. Ask your testers to go about their daily routine as usual, carrying their test phones or other devices with them at all times.

When testing visit detection accuracy, in particular, allow for enough dwell time for the SDK to register a visit. We recommend the following best practices.

- Do remain in the venue for at least 10 minutes. If the tester walks into a venue, stops for a few seconds and leaves, for example, it does not give the SDK a good chance to log the activity as a visit.
- Don't stop outside a venue for any reason other than to record a visit. Doing so may trigger a false positive.
- Do put some distance between venue visits. We recommend breaking a 100-meter radius to give an SDK's visit-detection functionality a chance to reset after detecting a previous visit.

Along with testing, it's critical to document test data. Use the following guidelines for documentation.

- Ask each tester to keep a separate log of the places he or she visited over a given test window. This allows for objective measurement of the success rate along with SDK-measured results.
- Remember not to stop outside a venue to fill out logs or evaluation forms, as this could trigger false-positive results.
- Don't forget to document false positives as well as true positive results.



In addition to manually filling out test logs, you may be able to have SDKs automatically log events, for example, by sending events to cloud-based logs or online platforms. Look to the developer of each SDK for more information.

Finally, make sure you get a large enough sample size to make your test results meaningful. As a general rule, it takes a sample size of around 330 data points to keep the margin of error within plus or minus five percent. This figure assumes independent measurements from different testers using different devices in different locations.

If a single tester goes to only 20 locations, for example, there is not enough data for that test to be statistically significant. In this case, the test could easily indicate an accuracy as low as 30 percent or as high as 90 percent, which is too wide a range to determine what truly works and what doesn't.

## Conclusion

Now that you have clear and detailed guidelines on how to best test location SDKs, we wish you the best of luck on your journey as you work to integrate location-awareness into your apps.

As you go, consider Foursquare's location awareness technology. Our Pilgrim SDK allows developers to seamlessly bring location services to their apps. It also provides access to additional features such as audience Segments and our comprehensive venue database.

To get started with the Pilgrim Toolkit, [sign up](#) for a developer account with Foursquare and then install [TestFlight](#) for iOS or download the [Android app](#). You can also integrate our SDK in your own app for free by visiting the [Foursquare developer site](#).