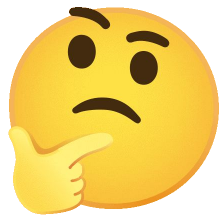
Lessons learned from building an open source Android SDK at Twilio

About me

Android dev. Sometimes gamer.
Podcast lover.

Questions? ferazoguerrero@twilio.com

What is a Mobile SDK



Some of the business aspects: Why companies build SDKs

SDK Types

Mobile-exclusive	API-based	Hardware components
		

Lesson #1:
The developer experience matters

About…

SMS

WhatsApp API

Voice

Video

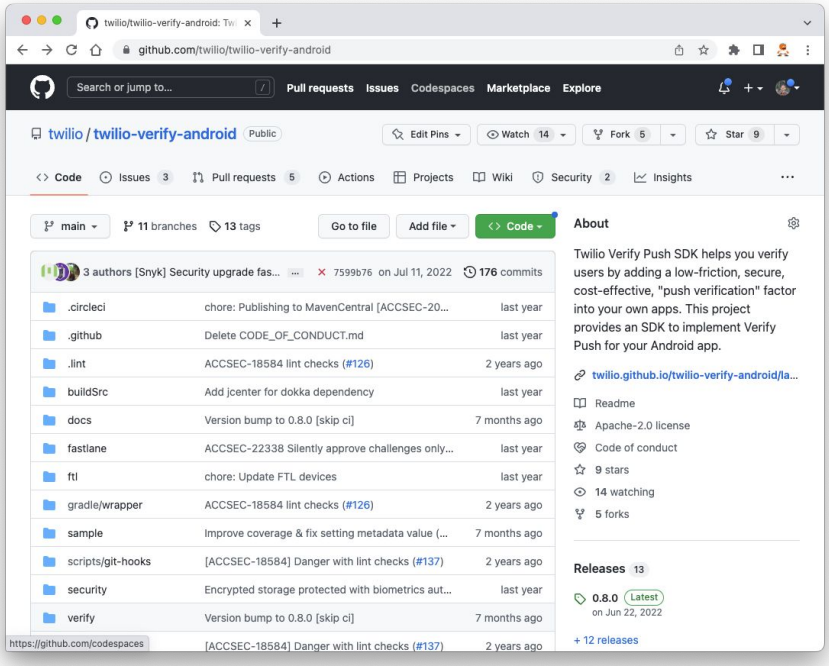
Email

Twilio Flex

…

Verify

Twilio Verify SDK



Allow your backend verifies users challenging them to prove they possess their security key (device).

https://github.com/twilio/twilio-verify-android

Mobile-exclusive	API-based	Hardware components
	 Twilio Verify SDK	

Lesson #2:
The SDK is a foreigner in the app territory

Size

Pursuing the lightweight

Developers choice depends on it

The ARR size vs. the actual size

Evaluate whether include or not external dependencies

Threading

Protect the app from ANR

Create an executor

Throw early exceptions

Transparency

The code should not be a black box

Be conscious about user permissions

Provide a log message mechanism

Fail with grace

Use annotations: deprecate, experimental

Conventions

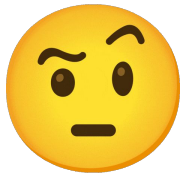
Adhere to platform standards

Official language style-guide

Design patterns

Common practices

What is the best language for building a SDK?



Lesson #3:
The language decision is not technical only

Lesson #4:
Keep platform parity

Platform parity

Repository name

Names and properties of business models

Exceptions

Contracts

Packages

Error codes

Keep idiomatic differences

Lesson #5:
Make the installation as instant as possible

Installation

Quickstart

Assist the developer integrating

Sample app

Include an instance of the installation done.

BoM

Bill of Materials

Gradle BoM (Firebase)

```
dependencies {
  implementation platform('com.google.firebase:firebase-bom:31.2.0')

  implementation 'com.google.firebase:firebase-analytics-ktx'
  // implementation 'com.google.firebase:firebase-auth-ktx'
  // implementation 'com.google.firebase:firebase-firestore-ktx'
}
```

Chris Banes' Compose BoM

```
dependencies {
    api(platform("dev.chrisbanes.compose:compose-bom:2023.02.00-alpha01"))

    // Use whichever Compose artifacts you need without a version number
    implementation("androidx.compose.foundation:foundation")
    implementation("androidx.compose.material:material")
    implementation("androidx.compose.material3:material3")
}
```

Lesson #6:
Be prepare for long-time versions

Release

- ❏ A suitable SDK probably will not be changed
- ❏ Automate release
- ❏ Dogfooding
- ❏ Promotion

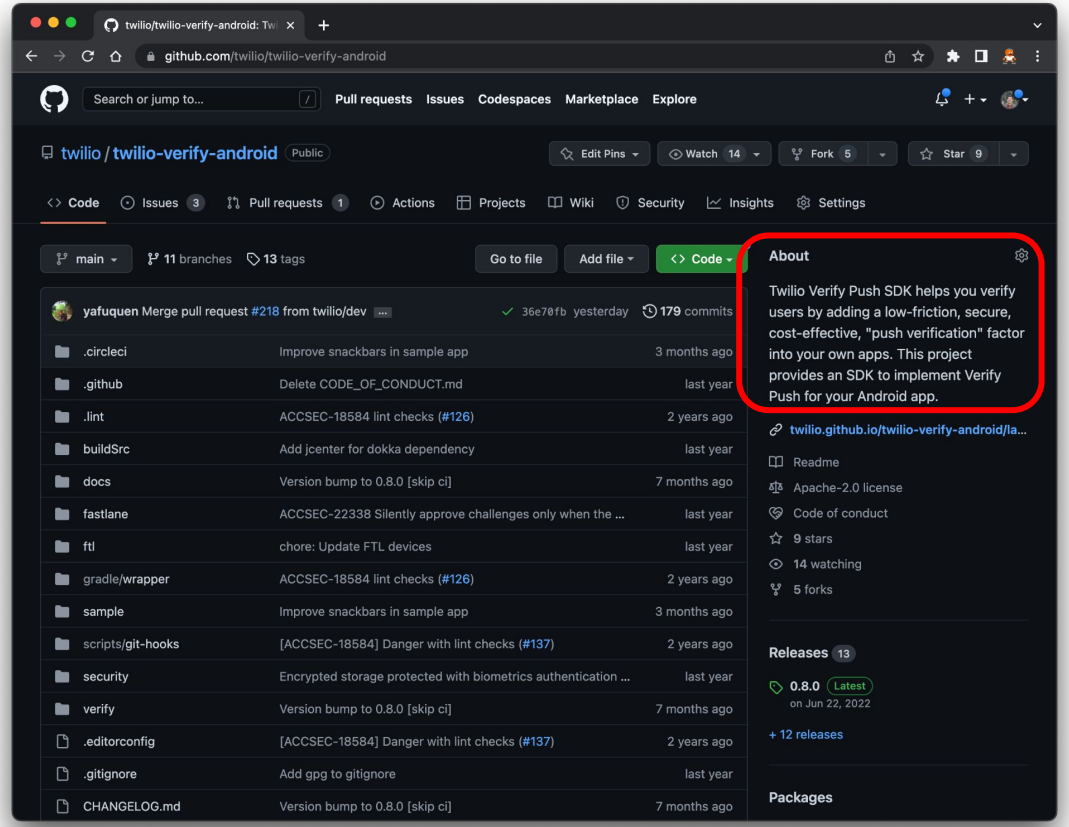
Lesson #7:
Create an open source project is also create community

Open sourcing

About section

Fill the about section.

Help others to know what the repo contains.



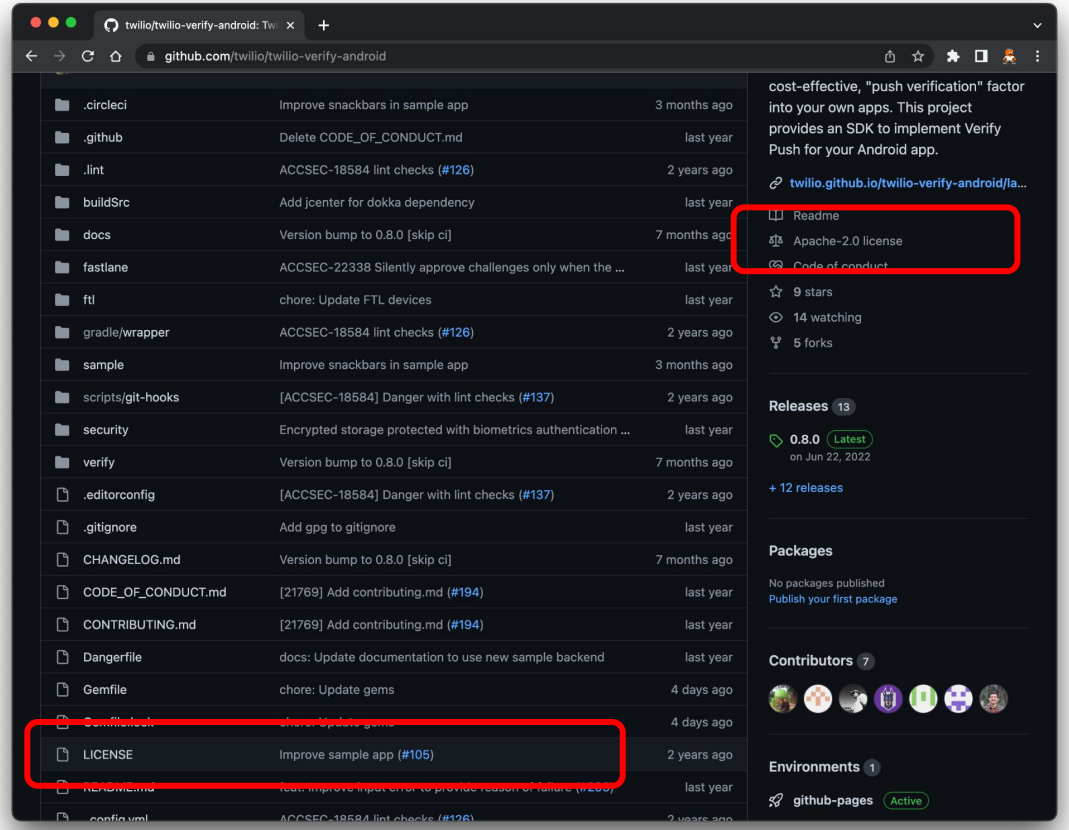
Licensing

There are different license types:

- MIT
- Apache License v2.0

Choose the type that fits the project better.

Visit: https://choosealicense.com/

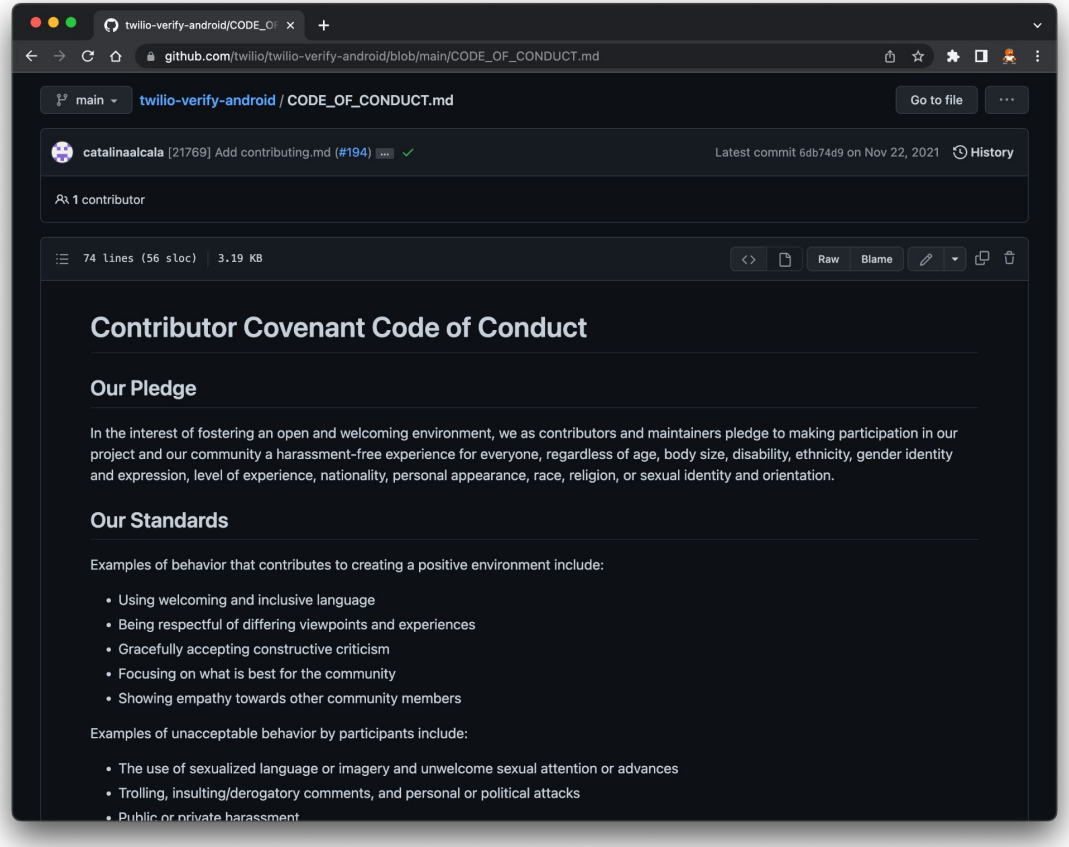


Code of conduct

Adopt a code of conduct to define community standards.

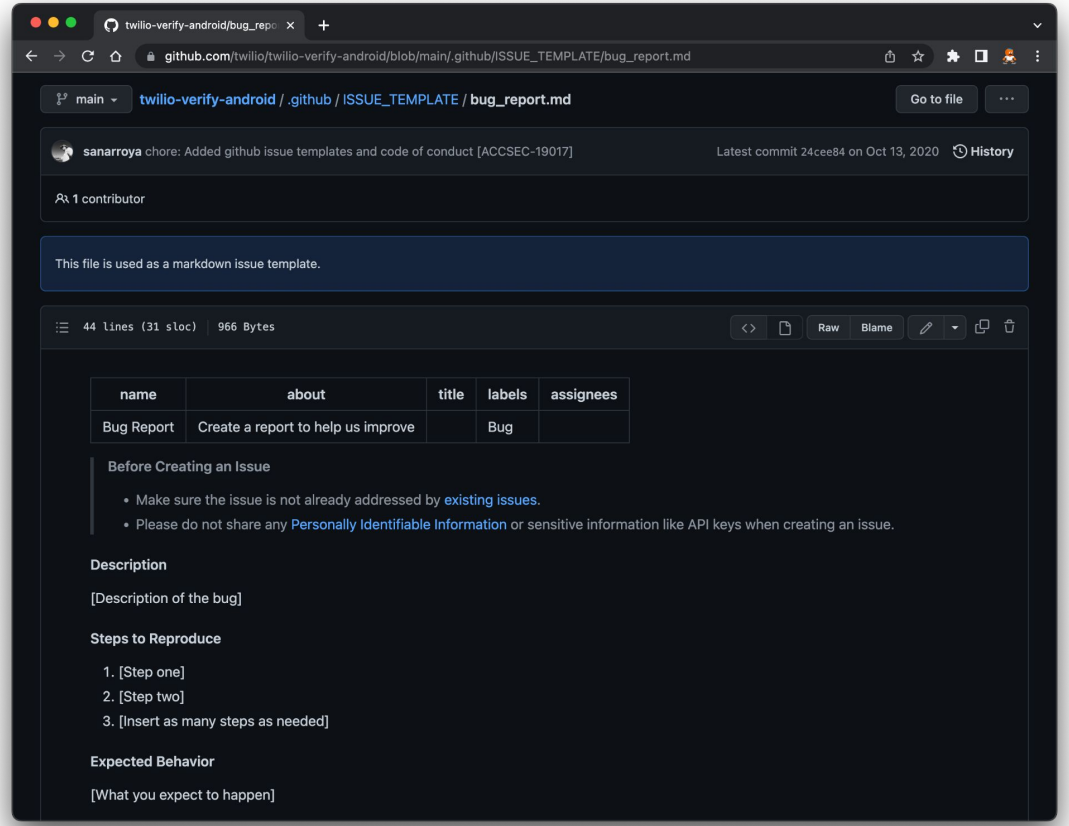
Signal a welcoming and inclusive project.

Outline procedures for handling abuse.



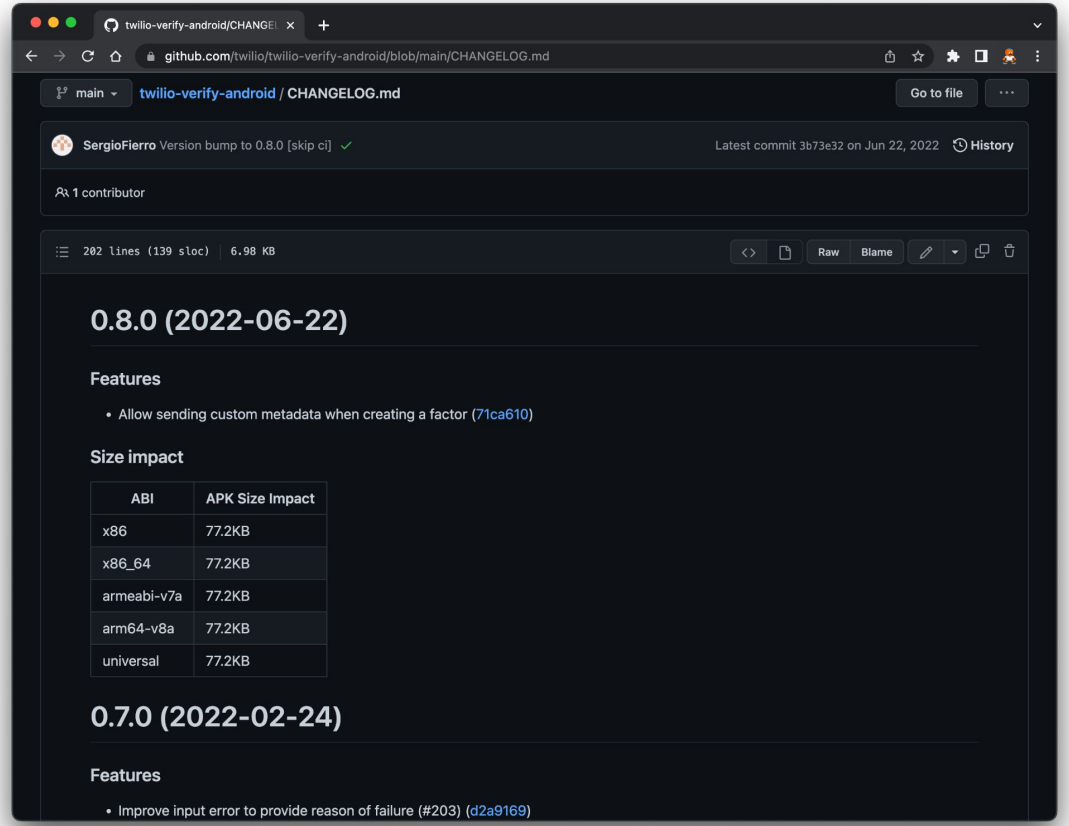
Issue report and feature request

Customize and standardize the information that you want contributors to include when they open issues and pull requests in your repository.



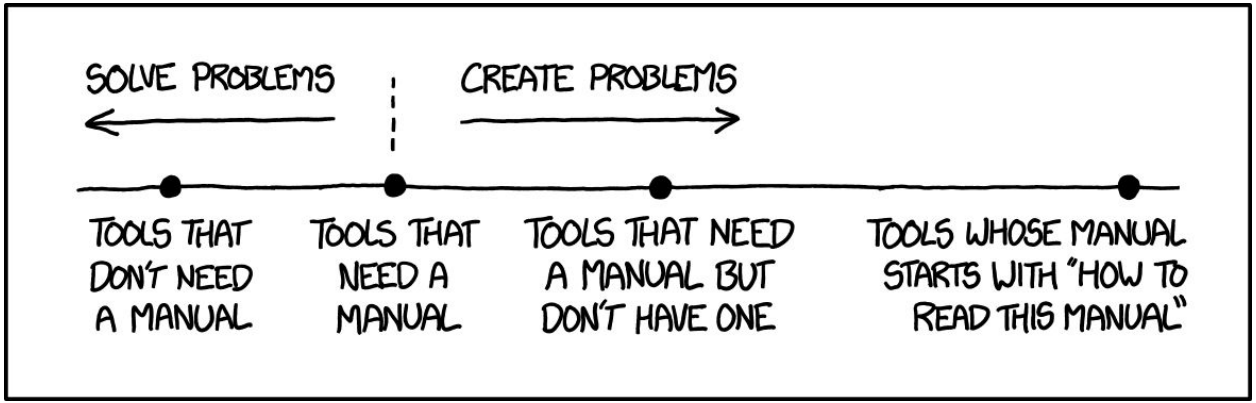
CHANGELOG file

Make it easier for users and contributors to see precisely what notable changes have been made between each release (or version) of the project.



Lesson #8:
Treat the documentation like the source code

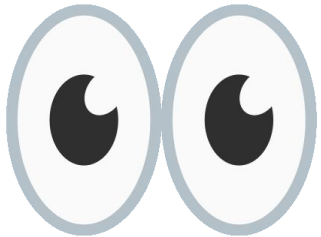
Documentation

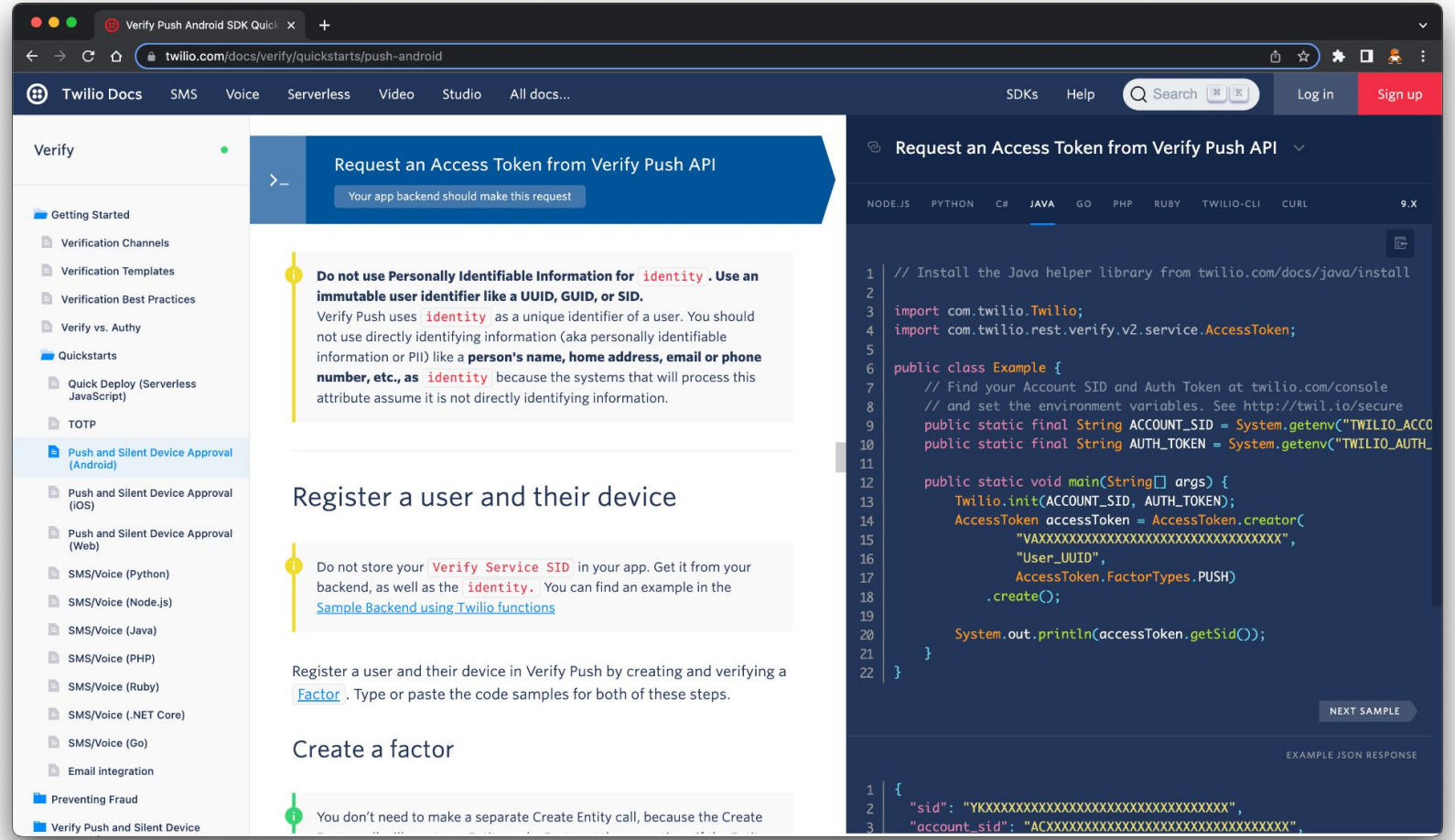


https://xkcd.com/1343/

Documenting

- Few people will read carefully.
- Include an index.
- Use headings and highlights.
- Ensure the code snippets work.
- When documenting steps:
  - Check the initial state.
  - Provide a final state verification.
- Think about the usage from a beginner’s perspective.
- Write first. Then, read.




Verify Push Android SDK Quick
twilio.com/docs/verify/quickstarts/push-android
Twilio Docs
SMS
Voice
Serverless
Video
Studio
All docs...
SDKs
Help
Search
Log in
Sign up
Verify
Getting Started
Verification Channels
Verification Templates
Verification Best Practices
Verify vs. Authy
Quickstarts
Quick Deploy (Serverless JavaScript)
TOTP
Push and Silent Device Approval (Android)
Push and Silent Device Approval (iOS)
Push and Silent Device Approval (Web)
SMS/Voice (Python)
SMS/Voice (Node.js)
SMS/Voice (Java)
SMS/Voice (PHP)
SMS/Voice (Ruby)
SMS/Voice (.NET Core)
SMS/Voice (Go)
Email integration
Preventing Fraud
Verify Push and Silent Device
Request an Access Token from Verify Push API
Your app backend should make this request
Do not use Personally Identifiable Information for identity. Use an immutable user identifier like a UUID, GUID, or SID.
Verify Push uses identity as a unique identifier of a user. You should not use directly identifying information (aka personally identifiable information or PII) like a person's name, home address, email or phone number, etc., as identity because the systems that will process this attribute assume it is not directly identifying information.
Register a user and their device
Do not store your Verify Service SID in your app. Get it from your backend, as well as the identity. You can find an example in the Sample Backend using Twilio functions
Register a user and their device in Verify Push by creating and verifying a Factor. Type or paste the code samples for both of these steps.
Create a factor
You don't need to make a separate Create Entity call, because the Create
Request an Access Token from Verify Push API
NODE.JS
PYTHON
C#
JAVA
GO
PHP
RUBY
TWILIO-CLI
CURL
9.X
// Install the Java helper library from twilio.com/docs/java/install
import com.twilio.Twilio;
import com.twilio.rest.verify.v2.service.AccessToken;
public class Example {
// Find your Account SID and Auth Token at twilio.com/console
// and set the environment variables. See http://twil.io/secure
public static final String ACCOUNT_SID = System.getenv("TWILIO_ACCO
public static final String AUTH_TOKEN = System.getenv("TWILIO_AUTH_
public static void main(String[] args) {
Twilio.init(ACCOUNT_SID, AUTH_TOKEN);
AccessToken accessToken = AccessToken.creator(
"VAXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
"User_UUID",
AccessToken.FactorTypes.PUSH)
.create();
System.out.println(accessToken.getSid());
}
}
NEXT SAMPLE
EXAMPLE JSON RESPONSE
{
"sid": "YKXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
"account_sid": "ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",

Closing thoughts