

GOOGLE PLAY CALL LOG PERMISSION GUIDELINES

The Permissions Declaration process is outlined by Google in detail [here](#). Of note, you may need to provide app review instructions, a video demonstration of your app, and/or test account credentials.

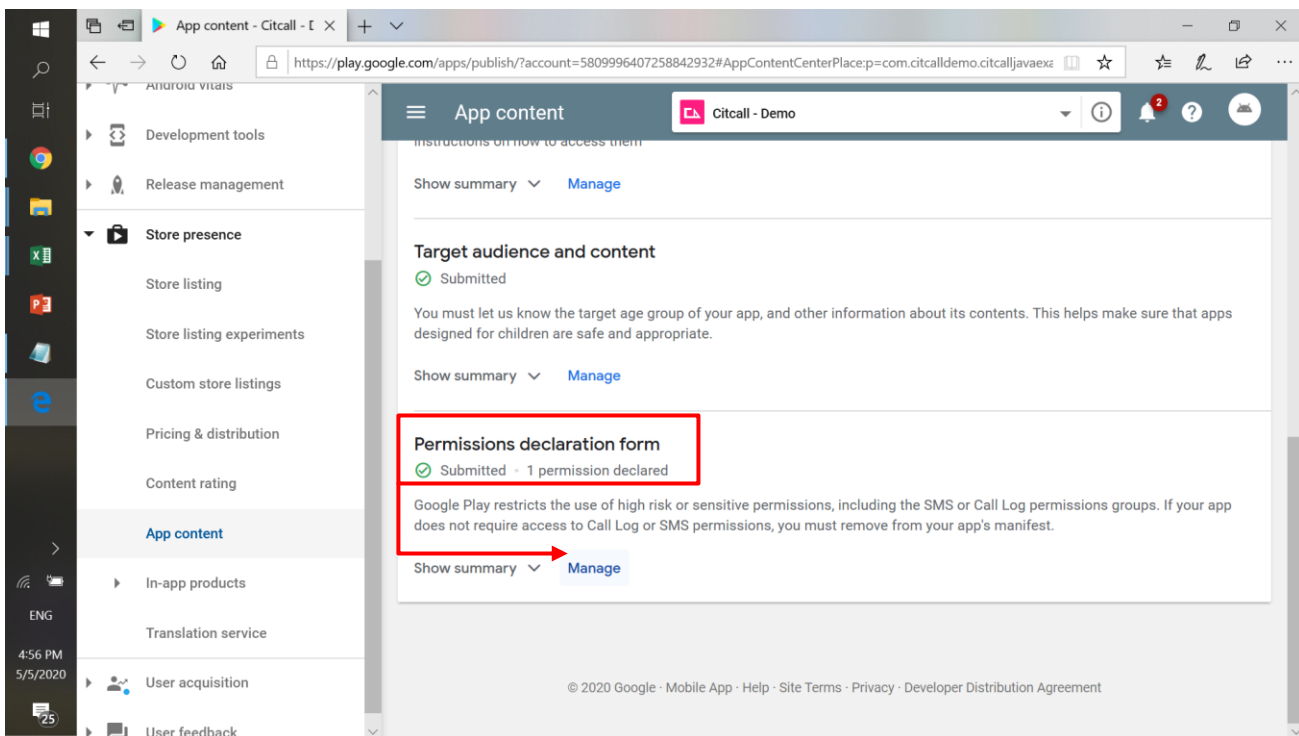
We recommend having these ready *before* beginning the Permissions Declaration process.

Steps to filling out the Permission Declaration Form

1. Choose app release and create a new release as you normally would at <https://play.google.com/apps/publish>
2. Usually, You will automatically see the Permission Declaration Form if your Android Manifest includes any of Google's restricted permissions.

you also can access Permission Declaration Form here (in case it not showed automatically)

Google Play Console > Store Presence > App Content > Permission declaration form > manage



3. Under “Compliance status” choose “Yes, this release meets the SMS and Call Log permissions policy”

Permissions Declaration Form * Required

Compliance status *

Does this release meet the SMS and Call Log permissions policy? If you choose no, you can submit the form without selecting any core functionalities.

Yes, this release meets the SMS and Call Log permissions policy

No, this release does not meet the SMS and Call Log permissions policy

4. Under “Core functionality” choose the options that best match your use case.

The screenshot shows the Google Play Console interface for 'App releases'. The 'Permissions Declaration Form' is expanded, showing the 'Core functionality' section. The form includes a warning about sensitive permissions and a list of checkboxes for core functionalities. The 'Previously declared permissions' section shows one permission is already declared.

Declare sensitive permissions

All apps requesting to use sensitive permissions, such as SMS or Call Log permission groups, must complete the Permissions Declaration Form. Failing to report any app functionality that uses these permissions can lead to app suspension.

We don't allow these permissions to be used for contact prioritization, affinity profiles, or social graphs.

If you're using SMS or Call Log permissions for account verification, sharing content, or initiating a text message or phone call, you can use alternative methods to get the same functionality. [Learn more](#)

Previously declared permissions (1 permission)

Permissions Declaration Form * Required

Core functionality What core functionalities in your app require SMS and Call Log permissions? ⓘ

- Default SMS handler
- Default Phone handler
- Default Assistant handler
- Transactional backup and restore for users and archive for enterprise (time-limited/non-continuous)
- Enterprise archive, CRM, and device management
- Caller ID, spam detection, and blocking
- Connected device companion apps (for example, smartwatch or

Tick on the Call-based OTP account verification to use auto-read for misscall OTP

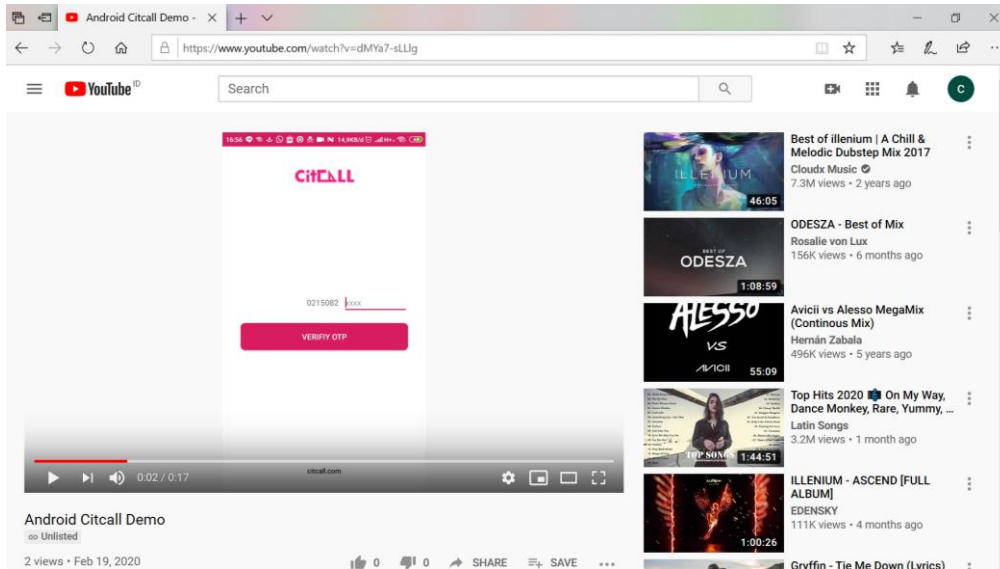
The screenshot shows the 'Core functionality' section of the 'Permissions Declaration Form'. The checkbox for 'Call-based OTP account verification' is checked. A yellow arrow points to this checkbox. Below the list is a section titled 'Don't see your use case?' with a link to 'REPORT NEW USE CASE' and 'LEARN MORE'.

- Physical safety / emergency alert apps (e.g., senior safety)
- Call-based OTP account verification
- Using SMS_CB_RECEIVED for customer communications (e.g., Smart Zone Cast service)
- Write and Show Call History in Dialer
- In-vehicle hands-free use or projected display
- Anti-SMS Phishing

Don't see your use case?
If your app's core functionality requires a new use case, you can report it below. Only apps with a use case listed on the form will be approved.

[REPORT NEW USE CASE](#) [LEARN MORE](#)

5. "Video Instructions" is important, given that a reviewer would need a particular SIM card this could be a good way to show them how your app works via youtube like picture below. **(Make sure the video link is not private & accessible to public)**
(Citcall can help upload video of your test apps on youtube if needed)



6. in instruction for app review :
Provide step to test the account (all steps until the misscall OTP verification or login step)
& **mention the OTP verification is using citcall.com to help google identify the usage**
Example
step 1 – open apps, go to log in
step 2, 3, & so on
We use Citcall.com as our vendor that need read_call_log for the otp verification

Instructions for app review

Is any additional information needed to review your app?

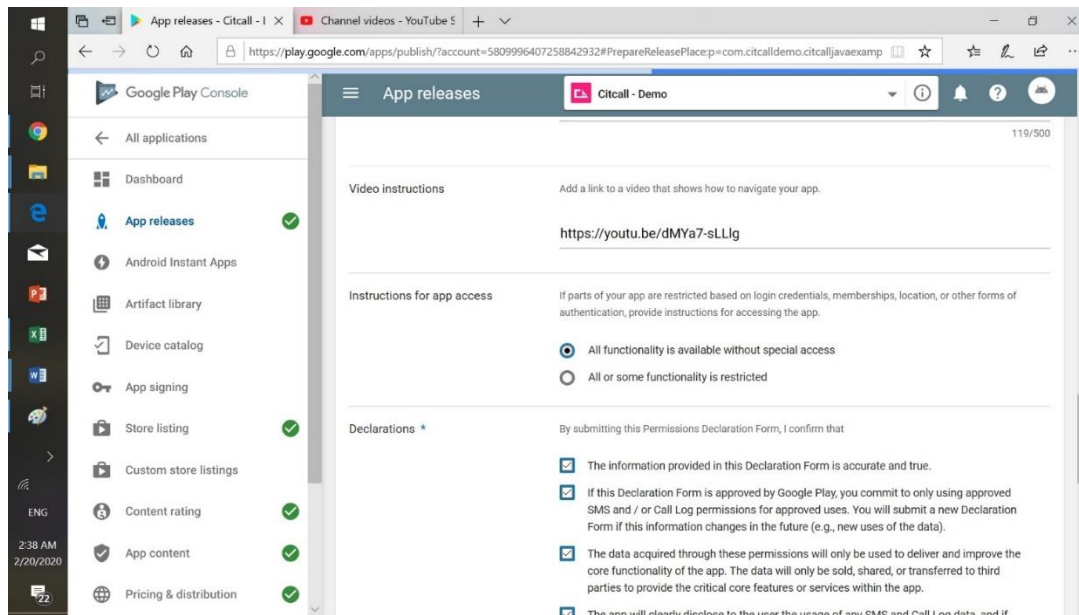
Details

0/500

Video instructions

Add a link to a video that shows how to navigate your app.

7. "Test account" gives Google test credentials to log into your app if your app requires them. (optional if available will be better)

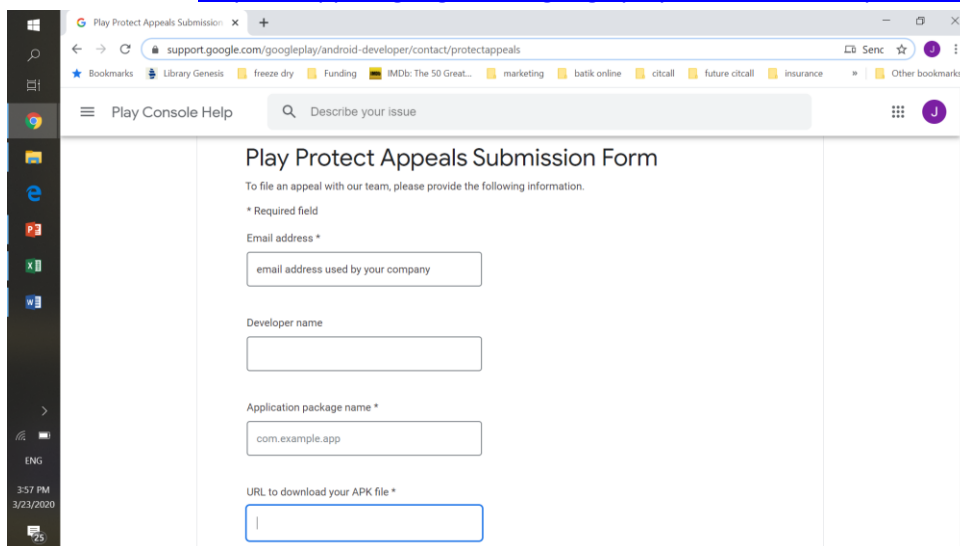


8. Read and agree to the “Declarations” section.
9. Complete the rest of the usual app release fields, then click SAVE. When you are ready to release click REVIEW.

GOOGLE PLAY APPEAL

In case you get rejected after following our guide, don't worry you can still appeal to google (appeal process normally +- 2days)

1. check your rejection email from google & fill in this links <https://support.google.com/googleplay/android-developer/contact/protectappeals>



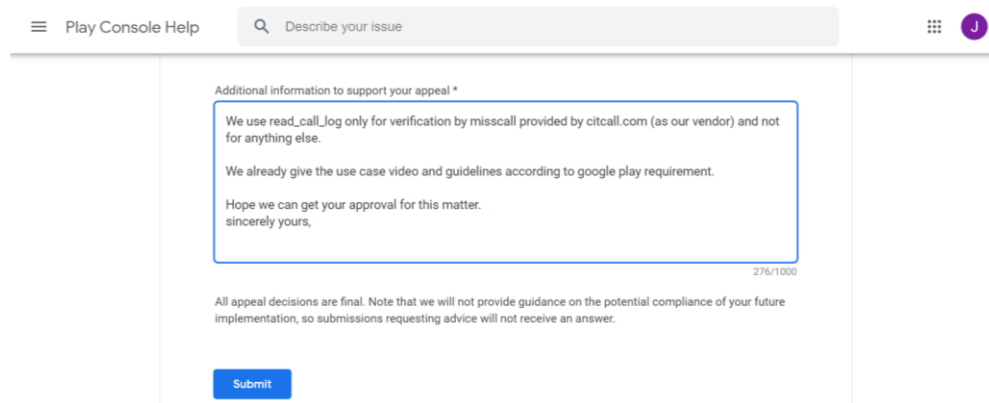
2. for the last box : Additional information to support your appeal *, you can copy paste the text below

We use read_call_log only for verification by misscall provided by citcall.com (as our vendor) and use read_call_log not for anything else beside that matter.

We already give the use case video and guidelines according to google play requirement.

Hope we can get your approval for this matter.

sincerely yours,



The screenshot shows the 'Additional information to support your appeal' form in the Play Console. The form contains the following text:

We use read_call_log only for verification by misscall provided by citcall.com (as our vendor) and not for anything else.

We already give the use case video and guidelines according to google play requirement.

Hope we can get your approval for this matter.

sincerely yours,

276/1000

All appeal decisions are final. Note that we will not provide guidance on the potential compliance of your future implementation, so submissions requesting advice will not receive an answer.

Submit

3. submit the form and wait for google reply (usually +-2 working days)

We've accepted your appeal and included details below about what additional steps you'll need to take to get your app to appear on Google Play.

Action required: Submit your app

In order to show your app on Google Play, please submit your app again:

1. Sign in to your [Play Console](#).
2. Select your app.
3. Select **Store presence** > **Store listing**.
4. Click **Submit update** or **Resubmit app**.

If the submit button is grayed out, you can make a minor change to your store listing to activate the button. For example, add a space after your app title and then delete it. Once the button turns blue, you can submit your update.

Before republishing your apps, you may also want to double check that your app complies with all [Developer Program Policies](#). If you need additional help, you can also visit the [Play Console Help Center](#) or the [Android Developers website](#).

Thanks for your continued support of Google Play!

4. After you get email from google that accepted your appeal then you can submit again update store listing and publish the latest version that rejected before