

PermissionsAndroid

Project with Native Code Required

The following section only applies to projects with native code exposed. If you are using the managed Expo workflow, see the guide on [Permissions](#) in the Expo documentation for the appropriate alternative.

PermissionsAndroid provides access to Android M's new permissions model. The so-called "normal" permissions are granted by default when the application is installed as long as they appear in `AndroidManifest.xml`. However, "dangerous" permissions require a dialog prompt. You should use this module for those permissions.

On devices before SDK version 23, the permissions are automatically granted if they appear in the manifest, so `check` should always result to `true` and `request` should always resolve to `PermissionsAndroid.RESULTS.GRANTED`.

If a user has previously turned off a permission that you prompt for, the OS will advise your app to show a rationale for needing the permission. The optional `rationale` argument will show a dialog prompt only if necessary - otherwise the normal permission prompt will appear.

Example

PermissionsAndroid Example

```
import React from 'react';
import {
  Button,
  PermissionsAndroid,
  StatusBar,
  StyleSheet,
  Text,
  View,
} from 'react-native';

const requestCameraPermission = async () => {
  try {
    const granted = await PermissionsAndroid.request(
      PermissionsAndroid.PERMISSIONS.CAMERA,
      {
        title: 'Cool Photo App Camera Permission',
        message:
          'Cool Photo App needs access to your camera ' +
          'so you can take awesome pictures.',
        buttonNeutral: 'Ask Me Later',
        buttonNegative: 'Cancel',
        buttonPositive: 'OK',
      },
    );
    if (granted === PermissionsAndroid.RESULTS.GRANTED) {
      console.log('You can use the camera');
    }
  }
}
```

Download [Expo Go](#) and scan the QR code to get started.



Connected devices 0

[Log in](#) or set a [Device ID](#) to open this Snack from Expo Go on your device or simulator.

Preview



My Device

Android

Permissions that require prompting the user

Available as constants under `PermissionsAndroid.PERMISSIONS`:

- `READ_CALENDAR`: `'android.permission.READ_CALENDAR'`
- `WRITE_CALENDAR`: `'android.permission.WRITE_CALENDAR'`
- `CAMERA`: `'android.permission.CAMERA'`
- `READ_CONTACTS`: `'android.permission.READ_CONTACTS'`
- `WRITE_CONTACTS`: `'android.permission.WRITE_CONTACTS'`
- `GET_ACCOUNTS`: `'android.permission.GET_ACCOUNTS'`
- `ACCESS_FINE_LOCATION`: `'android.permission.ACCESS_FINE_LOCATION'`
- `ACCESS_COARSE_LOCATION`: `'android.permission.ACCESS_COARSE_LOCATION'`

- `ACCESS_BACKGROUND_LOCATION` :
'android.permission.ACCESS_BACKGROUND_LOCATION'
- `RECORD_AUDIO` : 'android.permission.RECORD_AUDIO'
- `READ_PHONE_STATE` : 'android.permission.READ_PHONE_STATE'
- `CALL_PHONE` : 'android.permission.CALL_PHONE'
- `READ_CALL_LOG` : 'android.permission.READ_CALL_LOG'
- `WRITE_CALL_LOG` : 'android.permission.WRITE_CALL_LOG'
- `ADD_VOICEMAIL` : 'com.android.voicemail.permission.ADD_VOICEMAIL'
- `USE_SIP` : 'android.permission.USE_SIP'
- `PROCESS_OUTGOING_CALLS` : 'android.permission.PROCESS_OUTGOING_CALLS'
- `BODY_SENSORS` : 'android.permission.BODY_SENSORS'
- `SEND_SMS` : 'android.permission.SEND_SMS'
- `RECEIVE_SMS` : 'android.permission.RECEIVE_SMS'
- `READ_SMS` : 'android.permission.READ_SMS'
- `RECEIVE_WAP_PUSH` : 'android.permission.RECEIVE_WAP_PUSH'
- `RECEIVE_MMS` : 'android.permission.RECEIVE_MMS'
- `READ_EXTERNAL_STORAGE` : 'android.permission.READ_EXTERNAL_STORAGE'
- `WRITE_EXTERNAL_STORAGE` : 'android.permission.WRITE_EXTERNAL_STORAGE'
- `BLUETOOTH_CONNECT` : 'android.permission.BLUETOOTH_CONNECT'
- `BLUETOOTH_SCAN` : 'android.permission.BLUETOOTH_SCAN'
- `BLUETOOTH_ADVERTISE` : 'android.permission.BLUETOOTH_ADVERTISE'
- `ACCESS_MEDIA_LOCATION` : 'android.permission.ACCESS_MEDIA_LOCATION'
- `ACCEPT_HANDOVER` : 'android.permission.ACCEPT_HANDOVER'
- `ACTIVITY_RECOGNITION` : 'android.permission.ACTIVITY_RECOGNITION'
- `ANSWER_PHONE_CALLS` : 'android.permission.ANSWER_PHONE_CALLS'
- `READ_PHONE_NUMBERS` : 'android.permission.READ_PHONE_NUMBERS'
- `UWB_RANGING` : 'android.permission.UWB_RANGING'
- `BODY_SENSORS_BACKGROUND` : 'android.permission.BODY_SENSORS_BACKGROUND'
- `READ_MEDIA_IMAGES` : 'android.permission.READ_MEDIA_IMAGES'
- `READ_MEDIA_VIDEO` : 'android.permission.READ_MEDIA_VIDEO'

- `READ_MEDIA_AUDIO`: `'android.permission.READ_MEDIA_AUDIO'`
- `POST_NOTIFICATIONS`: `'android.permission.POST_NOTIFICATIONS'`
- `NEARBY_WIFI_DEVICES`: `'android.permission.NEARBY_WIFI_DEVICES'`
- `READ_VOICEMAIL`: `'com.android.voicemail.permission.READ_VOICEMAIL'`,
- `WRITE_VOICEMAIL`: `'com.android.voicemail.permission.WRITE_VOICEMAIL'`,

Result strings for requesting permissions

Available as constants under `PermissionsAndroid.RESULTS`:

- `GRANTED`: `'granted'`
- `DENIED`: `'denied'`
- `NEVER_ASK_AGAIN`: `'never_ask_again'`

Reference

Methods

`check()`

```
static check(permission: Permission): Promise<boolean>;
```

Returns a promise resolving to a boolean value as to whether the specified permissions has been granted.

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
permission	string	Yes	The permission to check for.

request()

```
static request(  
  permission: Permission,  
  rationale?: Rationale,  
): Promise<PermissionStatus>;
```

Prompts the user to enable a permission and returns a promise resolving to a string value (see result strings above) indicating whether the user allowed or denied the request or does not want to be asked again.

If `rationale` is provided, this function checks with the OS whether it is necessary to show a dialog explaining why the permission is needed (<https://developer.android.com/training/permissions/requesting.html#explain>) and then shows the system permission dialog.

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
permission	string	Yes	The permission to request.
rationale	object	No	See <code>rationale</code> below.

Rationale:

NAME	TYPE	REQUIRED	DESCRIPTION
title	string	Yes	The title of the dialog.
message	string	Yes	The message of the dialog.
buttonPositive	string	Yes	The text of the positive button.
buttonNegative	string	No	The text of the negative button.
buttonNeutral	string	No	The text of the neutral button.

requestMultiple()


```
static requestMultiple(  
  permissions: Permission[],  
): Promise<{[key in Permission]: PermissionStatus}>;
```

Prompts the user to enable multiple permissions in the same dialog and returns an object with the permissions as keys and strings as values (see result strings above) indicating whether the user allowed or denied the request or does not want to be asked again.

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
permissions	array	Yes	Array of permissions to request.

Is this page useful?  

 Edit this page

Last updated on **Jun 21, 2023**