



A Native screenshot blocking library for React-Native developer, with background customizable after captured. Screenshot detector are also supported.

MIT license


Code of conduct

50 stars 5 forks 2 watching Activity

Public repository

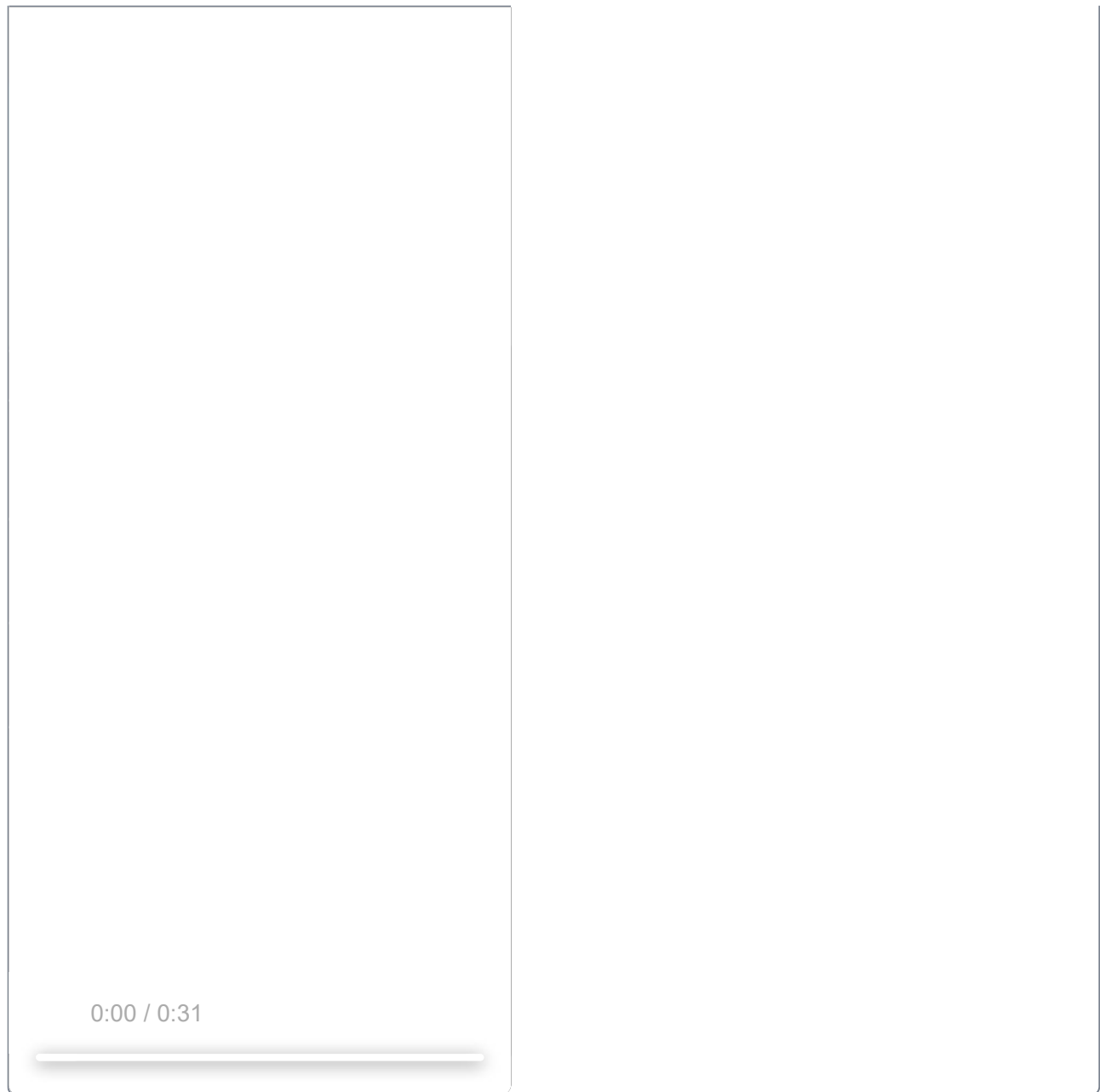
master

Branches Tags

gbumps Update App.tsx ... 3 days ago 67 

[View code](#)

The screenshot shows a GitHub repository page for 'react-native-background-color-screenshot'. At the top, there are badges indicating it is 'styled with prettier', has a 'License MIT', and uses 'ESLint'. Below these, the repository name 'react-native-background-color-screenshot' is displayed in a large, bold font. Underneath the name, the description reads: 'A Native library for blocking screenshot for react native developer, with background color screenshot customizable.' Further down, there is a video player section with a video titled 'avnzod.mp4'.



Get started

- Installation
 - 1.Install
 - stable
 - beta
 - 2.Linking
- Usage
 - register
 - registerWithoutScreenguard
 - registerWithBlurView (beta)

- [registerWithImage \(beta\)](#)
- [unregister](#)
- [Limitation](#)
- [Contributing](#)
- [License](#)

Installation

1. Install

Stable

- For protecting from being screenshoted and screen recording captured only, stable version is enough.

```
$ npm install react-native-screenguard --save
```



```
$ yarn add react-native-screenguard
```



Beta

- If you want more customization over the screen protector filter like `registerWithBlurView` and `registerWithImage`, install this version.

```
$ npm install react-native-screenguard@beta --save
```



```
$ yarn add react-native-screenguard@beta
```



Note : Remember to `pod install` on ios and `gradle build` on Android again to take effect.

2. Linking

- React-native 0.60 and higher: just `cd ios && pod install`, no additional requirements.
- React-native 0.59 and lower: Please do manual installation as follow

iOS

- i. In XCode, in the project navigator, right click `Libraries` → `Add Files to [your project's name]`
- ii. Go to `node_modules` → `react-native-screenguard` and add `ScreenGuard.xcodeproj`
- iii. In XCode, in the project navigator, select your project. Add `libScreenguard.a` to your project's `Build Phases` → `Link Binary With Libraries`

Android

- i. Open up `android/app/src/main/java/[...]/MainActivity.java`
 - o Add `import com.screenguard.ScreenGuardPackage;` to the imports at the top of the file
 - o Add `new ScreenGuardPackage()` to the list returned by the `getPackages()` method
- ii. Append the following lines to `android/settings.gradle` :

```
include ':react-native-screenguard'
project(':react-native-screenguard').projectDir = new
File(rootProject.projectDir, '../node_modules/react-native-
screenguard/android')
```



- iii. Insert the following lines inside the dependencies block in `android/app/build.gradle` :

```
compile project(':react-native-screenguard')
```



For Expo user: First, you need to eject Expo or `npx expo prebuild` in order to use this library, check Expo docs below:

<https://docs.expo.dev/workflow/prebuild/>



Post-installation setting for Android important

On Android, from `v0.1.4+` , remember to add a little more options as it won't work as expected.

1. Open up [your_project_path]/android/app/src/main/AndroidManifest.xml and add activity com.screenguard.ScreenGuardColorActivity like below

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
  <application .....>
    <activity
      android:name=".MainActivity" .....>
      .....
    </activity>

    <activity android:name="com.screenguard.ScreenGuardColorActivity"
      android:theme="@style/Theme.AppCompat.Translucent"
      />
    </application>
  </manifest>
```



2. Open up [your_project_path]/android/app/src/main/res/values/styles.xml and add style Theme.AppCompat.Translucent like below

```
<resource>

<style name="AppTheme">your current app style theme.....</style>

<style name="Theme.AppCompat.Translucent">
  <item name="android:windowNoTitle">true</item>
  <item name="android:windowBackground">@android:color/transparent</item>
  <item name="android:colorBackgroundCacheHint">@null</item>
  <item name="android:windowIsTranslucent">true</item>
  <item name="android:windowAnimationStyle">@null</item>
  <item name="android:windowSoftInputMode">adjustResize</item>
</style>
</resource>
```



Usage

Note : All features below contain a callback method after a screenshot or screen recording has been taken.

1. register

- (iOS + Android) : Activate the screenguard with your custom background color layout.
- Android will receive the background color when app in background or inactive state.

```
import ScreenGuardModule from 'react-native-screenguard';

ScreenGuardModule.register(
  //insert any hex color you want here, default black if null or empty
  '#0F9D58',
  (_) => {
    .....do anything you want after the screenshot
  });
```



iOS



Android



2. registerWithoutScreenguard

- (iOS + Android) Activate without screenguard, if you just want to detect and receive event callback only.

```
import ScreenGuardModule from 'react-native-screenguard';
```

```
ScreenGuardModule.registerWithoutScreenguard(  
  (_) => {  
    .....do anything you want after the screenshot  
  });
```



3. registerWithBlurView

- Beta version only. See how to install [here](#)
- Activate screenguard with a blurred effect view after captured.
- Blurview on Android using [Blurry](#).
- Accepted a JS object with following parameters:
 - **radius (required)**: blur radius value number in between [15, 50] (Explain below) , throws warning if smaller than 15 or bigger than 50, exception if smaller than 1 or not a number.
 - **timeAfterResume (Android only)**: A small amount of time (in milliseconds) for the blur view to disappear before jumping back to the main application view, default 1000ms

```
import ScreenGuardModule from 'react-native-screenguard';
```



```
const data = {  
  radius: 35,  
  timeAfterResume: 2000,  
};
```

```
//register with a blur radius of 35  
ScreenGuardModule.registerWithBlurView(data, (_) => {  
  .....do anything you want after the screenshot  
});
```

Explain : Set blur radius smaller than 15 won't help much, as content still look very clear and easy to read. Same with bigger than 50 but content will be shrinked and vanished inside the view, blurring is meaningless. So, between 15 and 50 is enough.

📺 screenguard_blur.mp4 ▾



4. registerWithImage

- Beta version only. See how to install [here](#)
- Activate screenguard with a custom image view and background color.
- ImageView using [SDWebImage](#) on iOS and [Glide](#) on Android for faster loading and caching.
- Accepted a JS object with following parameters:
 - width : width of the image
 - height : height of the image

- `uri` (**required**): uri of the image, accept all kinds of image (jpg|jpeg|png|gif|bmp|webp|svg), throws warning if uri is not an image uri;
- `backgroundColor` : background color behind the image, just like `register` .
- `timeAfterResume` (**Android only**): A small amount of time (in milliseconds) for the blur view to disappear before jumping back to the main view, default 1000ms

```
import ScreenGuardModule from 'react-native-screenguard';

const data = {
  height: 150,
  width: 200,
  uri: 'https://www.icegif.com/wp-content/uploads/2022/09/icegif-386.gif',
  backgroundColor: color,
  alignment: 5 // Alignment.centerRight
},
//register with an image
ScreenGuardModule.registerWithImage(
  data,
  (_) => {
    .....do anything you want after the screenshot
  });
```



Warning : This feature is still in experimental on Android, so please use with caution as some unexpected behaviour might occurs.

5. unregister

- (iOS + Android) Deactivate the screenguard.

```
ScreenGuardModule.unregister();
```



Limitation

- This library support blocking screenshot for iOS 13+, Android 5+ only.
- The protection filter is already activated until you call `unregister` . So remember to call a function only **ONCE** for limiting errors and unexpected problems might happened during testing.

- Lib does not support combine feature together yet. (For example you want to use `registerWithBlurView` combine with `register` to have a blur view with color behind,.....)
- On Android, if you want to use callback, consider using `registerWithoutScreenguard` instead, as you may not receive any event after a screenshot has been triggered if using with `register` .


Contributing

All contributions are welcome! Please open an issue if you get stuck and bugs, or a PR if you have any feature idea, improvements and bug fixing. I'm very appreciate !

License

MIT

Releases 8

 0.2.2 - stable Latest
3 days ago

Contributors 3



gbumps Goosebump



ddikodroid Ahmad Syarifuddin Randiko



SaeedZhiany SaeedZhiany

● TypeScript 30.9% ● JavaScript 26.2% ● Java 25.2% ● Objective-C++ 11.5% ● Ruby 3.1%
● Objective-C 3.1%

Report repository