Abhishek Jain

FOLLOW

Technology Enthusiast | Android Developer | Software Developer

Share data between your Android apps securely

Published Jul 15, 2017

There may be certain scenarios where one app needs to share some sensitive data to another app. The requirement is pretty simple, there should a simple interface for one app to request data that is exclusive to another app. The important part of this requirement is that no other app should be able to get hold of that sensitive data. This post is about how you can achieve this.

There is a well defined and documented way in which you can share data between apps. Android framework provides a very straight-forward way to share data among different apps using ContentProviders. The idea is simple, the app that intents to share data with other apps declares a content provider with its corresponding contract. A contract is what tells other apps on how to request data from the defined content provider. Content provider can expose all types of CRUD operations. The implementation of those operations is hidden behind the content provider implementation. This works great and serves the first part of the requirement perfectly. We can share the data from one app to another seamlessly. Here begins the interesting part of the story, how do we fulfill the second half of the requirement, the exclusive sharing of the data to only a limited set of apps. Well here is how..

Android provides us a way to define the permissions on the content provider we define. These are the permissions the consumer of the content provider must hold in order to be able to request and receive the data from the provider.

```
<provider
   android:name="CertificateProvider"
   android:permission="custom.permission.CONTENT_PROVIDER"
   android:authorities="package_name.CertificateProvider"
   android:exported="true" />
```

You would also need to define a custom permission:

```
<permission
   android:name="custom.permission.CONTENT_PROVIDER"
   android:protectionLevel="signature" />
```

Notice the `protectionLevel` attribute of this permission. The protection level for this permission is set to `signature`. Signature here denotes the certificate or key-store used to sign while generating an apk. What this means is that the app requesting data from the content providers defined above must hold this permission. And since the protection level of this permission is set to signature, the app holding the permission must also be signed with the same certificate/key-store. If all these conditions are not met, an app trying to access data through the content provider will very gently crash! 😇

So back to our initial goal. Since both the apps are provided by us. We can sign both the apps with the same certificate so that both have the same signature. This way only the app signed with the certificate that we hold can access the data through our content provider. Mission accomplished!! 😎

Android | Content providers | Share | App | Data

Enjoy this post? Give **Abhishek Jain** a like if it's helpful.

♡

8

SHARE

Abhishek Jain

Technology Enthusiast | Android Developer | Software Developer

I am an experienced Android developer with over 3 years of experience. My areas of expertise include writing well-designed and highly testable code. I am a tech enthusiast, new technology excite me!

FOLLOW