

Preparing your app for Google Play Instant

Droidcon London 2018







IMMOBILIEN
SCOUT 24



Google Play





Friction



IMMOBILIEN
SCOUT 24

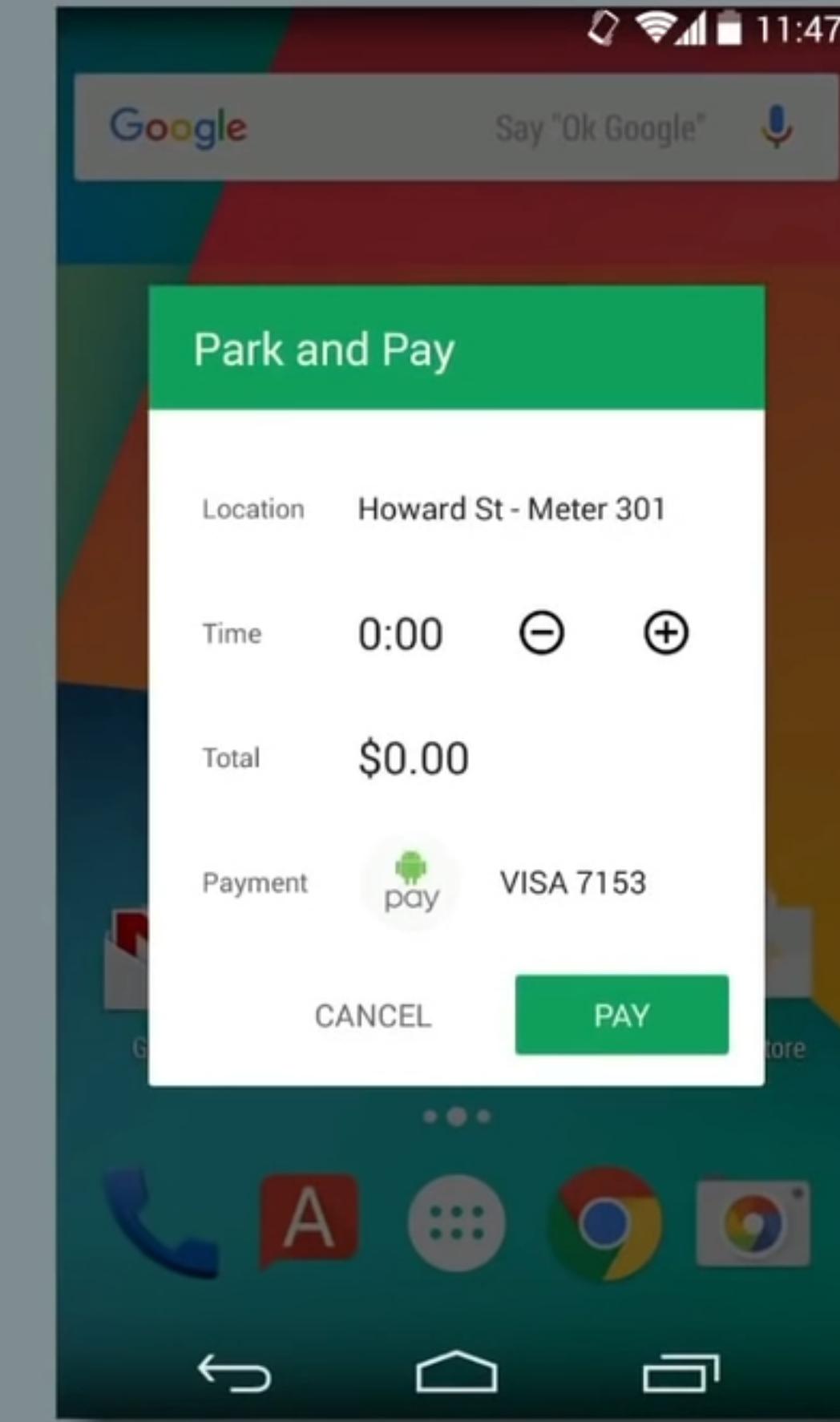
youtube.com/MakStudios

HUFF
POST



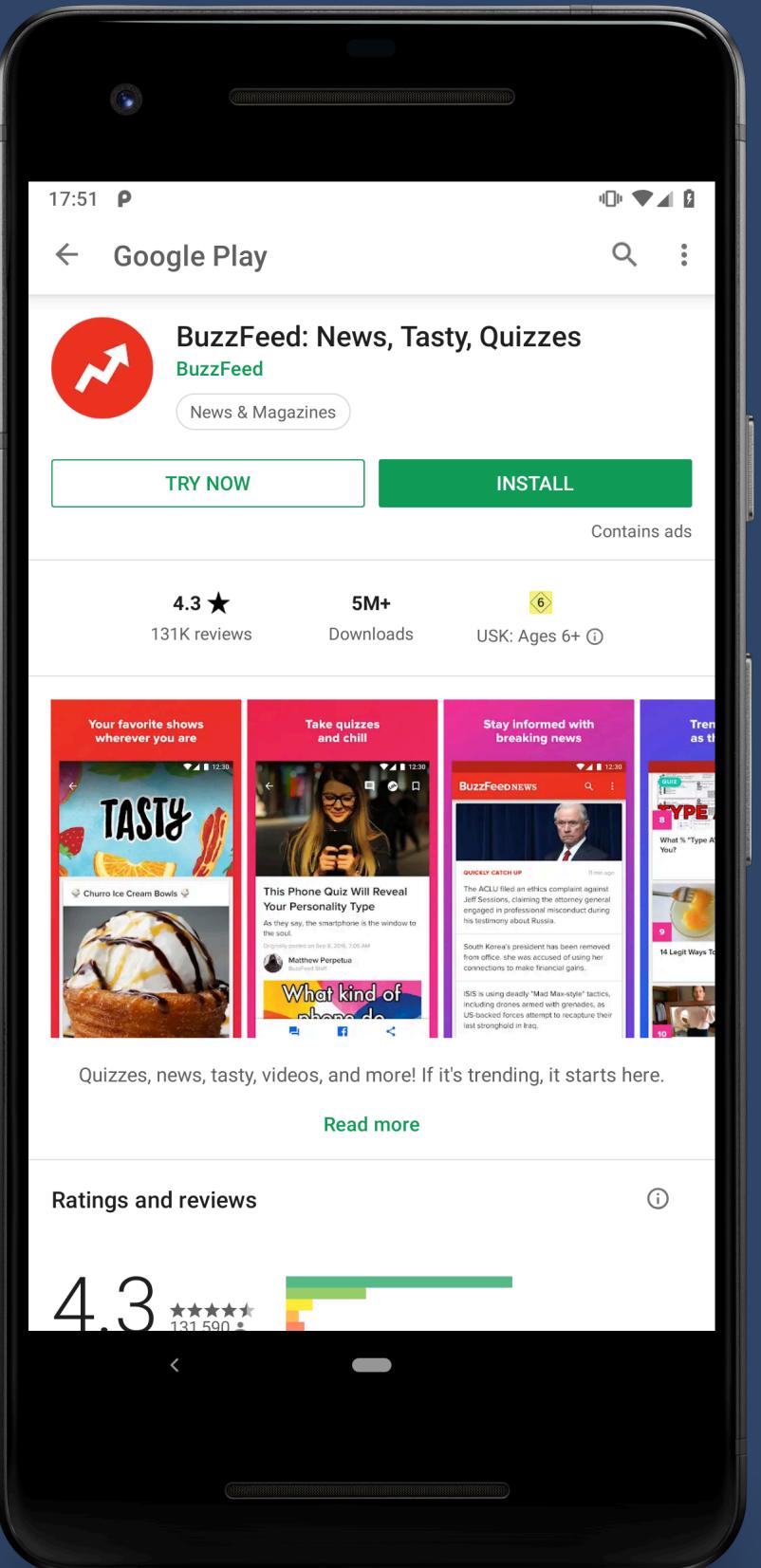


Google Play Instant



< a href />

Try Now





Performance

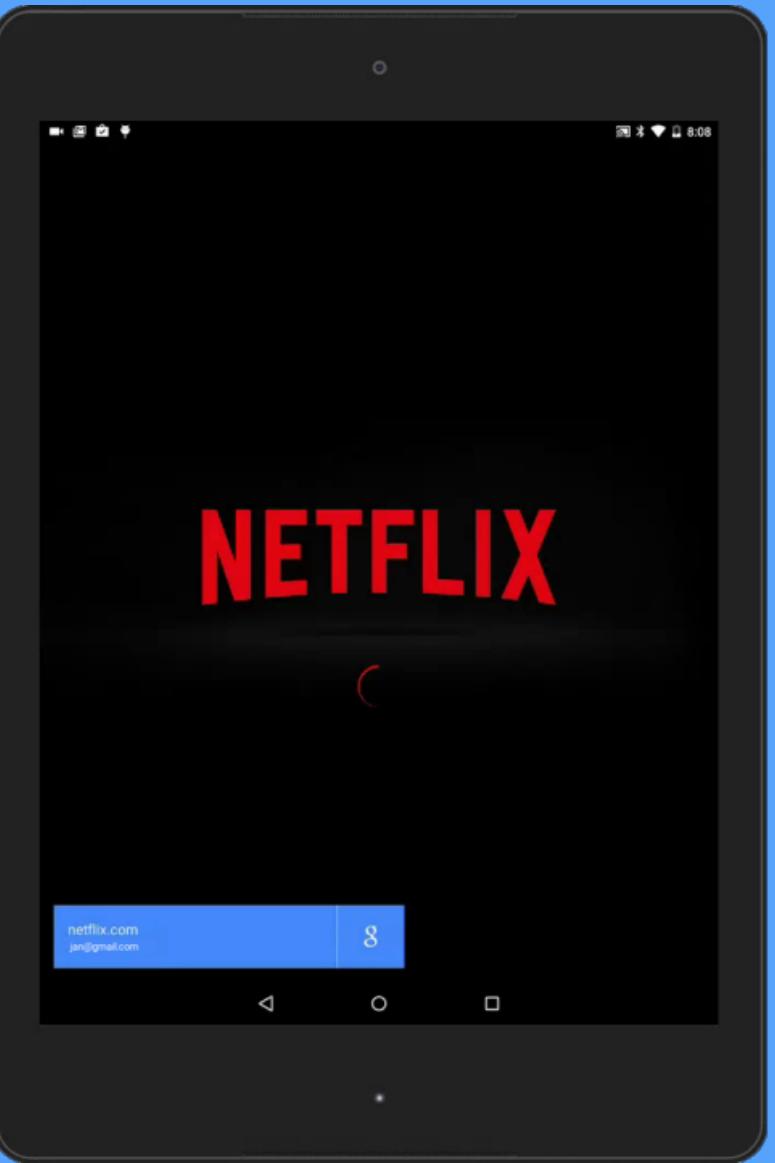


Security



HTTPS

Smart Lock





Play Billing Library

Runtime Permissions

- ACCESS_COARSE_LOCATION
- ACCESS_FINE_LOCATION
- ACCESS_NETWORK_STATE
- BILLING (Deprecated as of Play Billing Library 1.0)
- CAMERA
- INSTANT_APP_FOREGROUND_SERVICE (Only in Android Oreo)
- INTERNET
- READ_PHONE_NUMBERS (Only in Android Oreo)
- RECORD_AUDIO
- VIBRATE

Limitations



Background Services

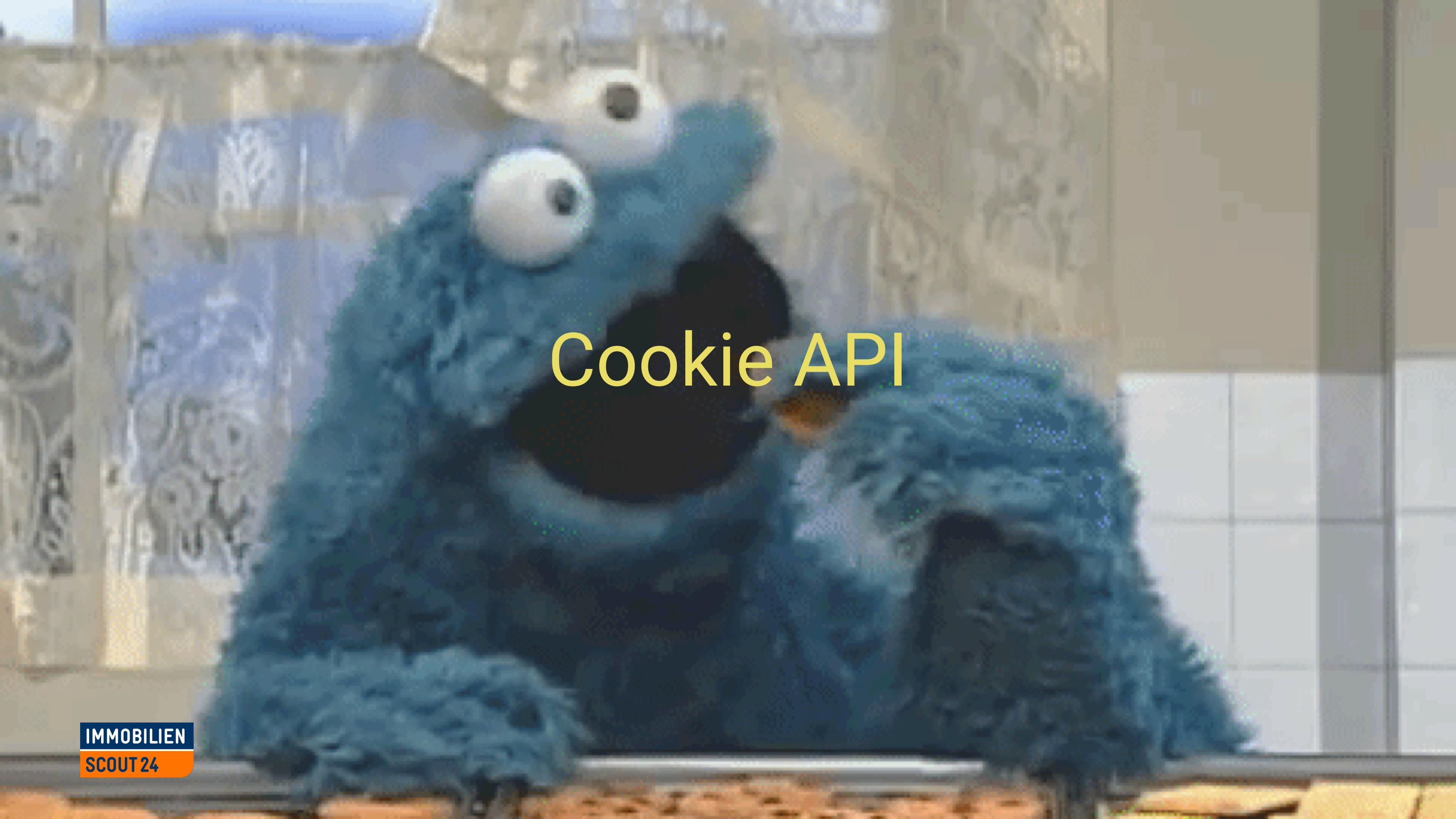
Push Notifications

External Storage



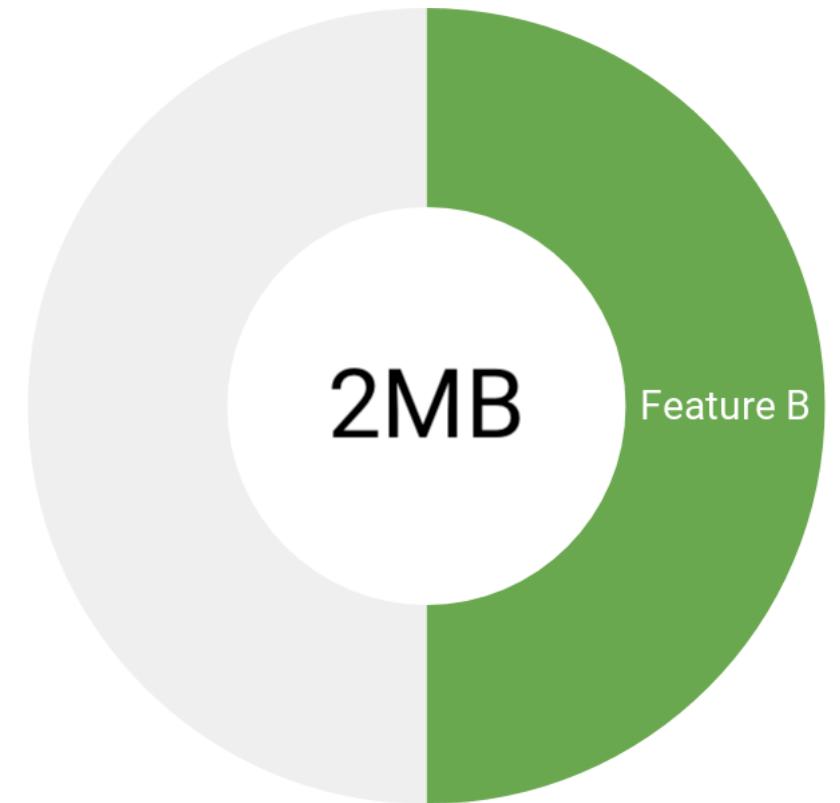
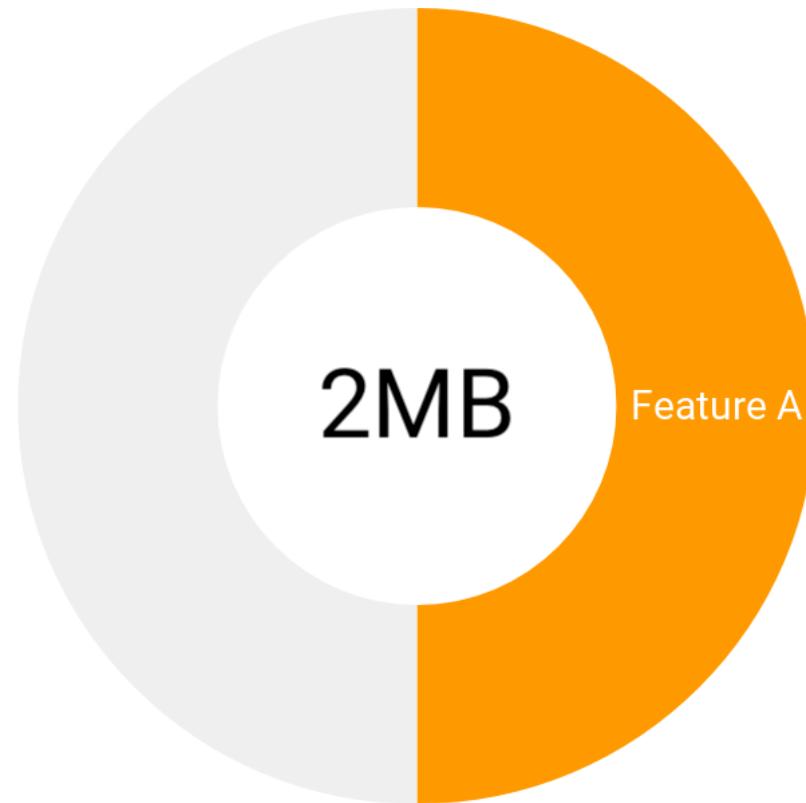
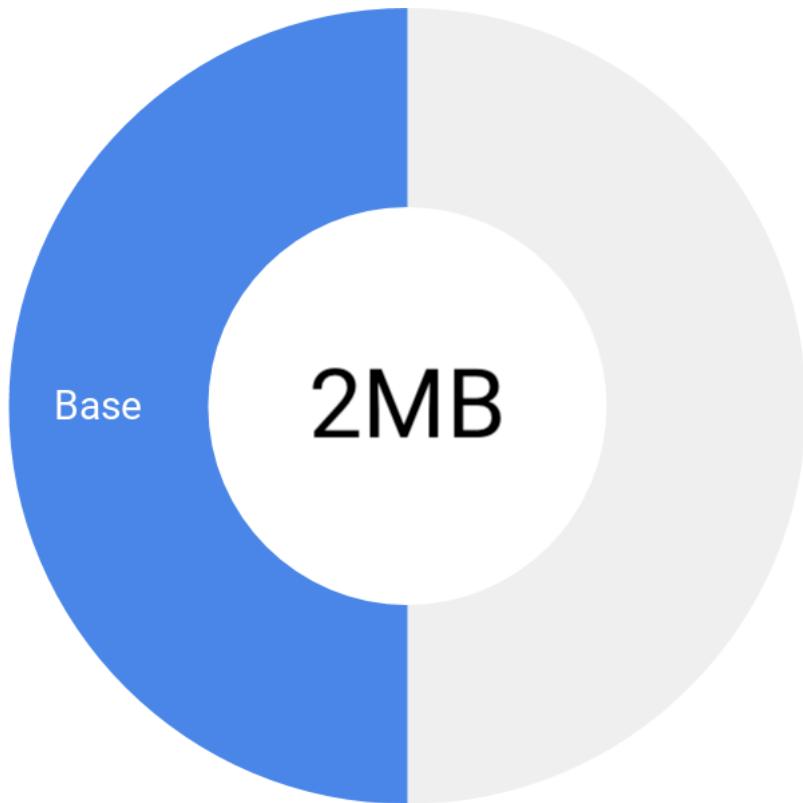
Local Storage

Instant <-> Installed

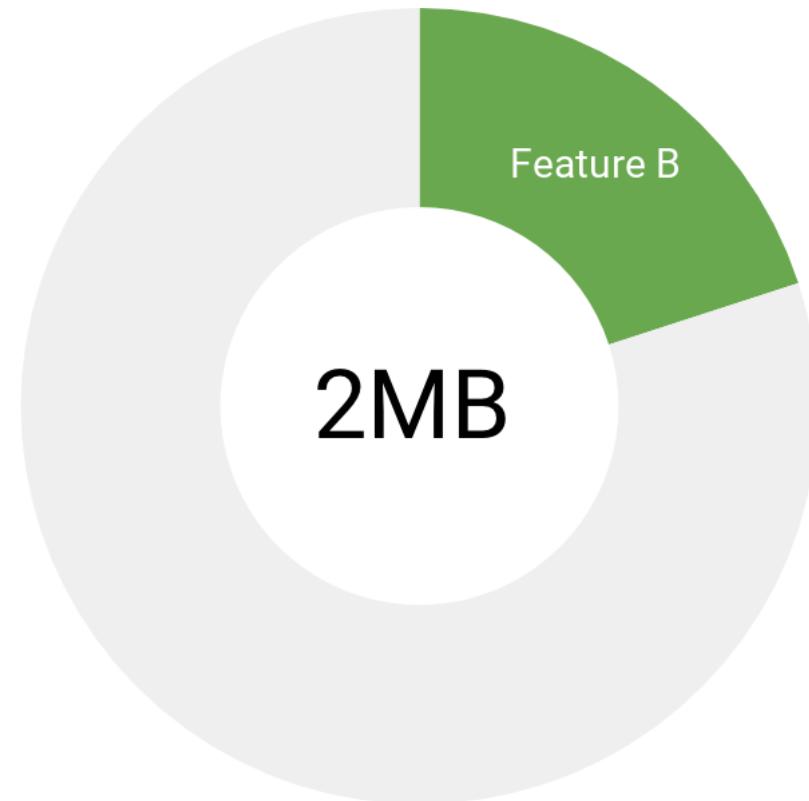
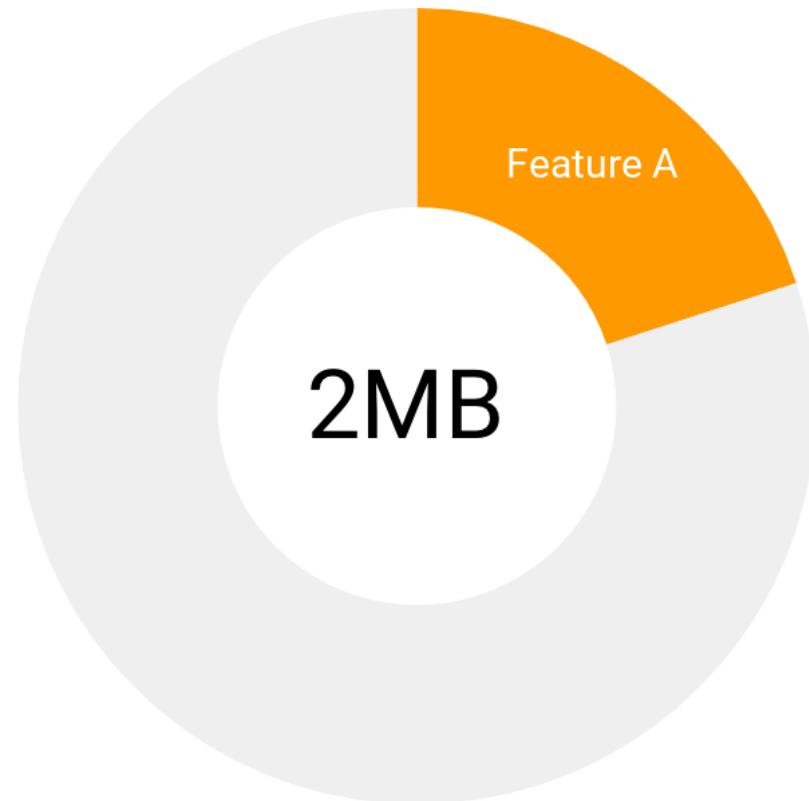
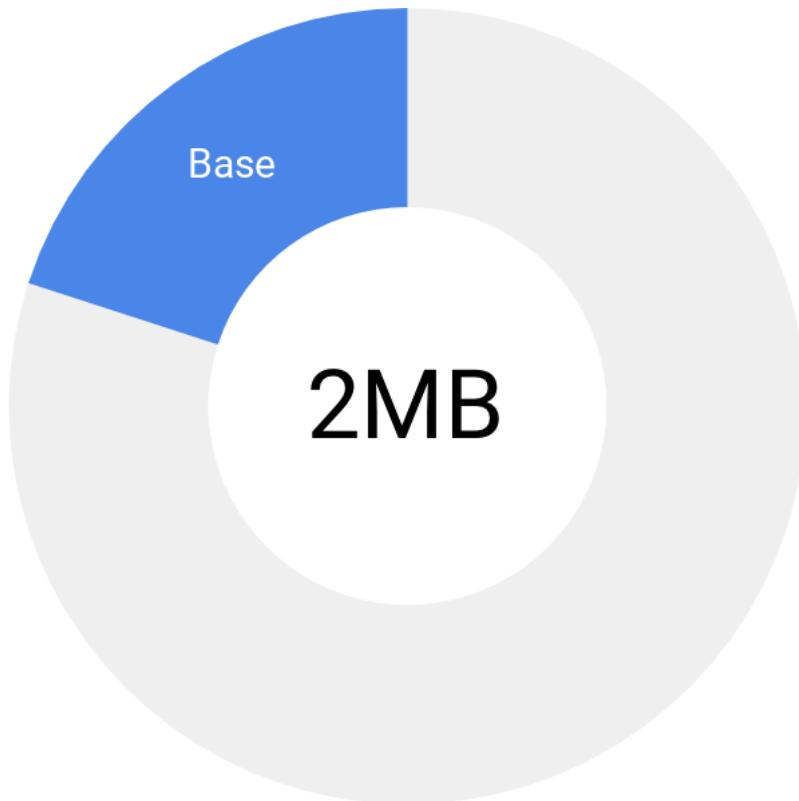


Cookie API

Size Limit (4 Mb)



Size Limit ~~(4 Mb)~~ (10 Mb) 💪



History

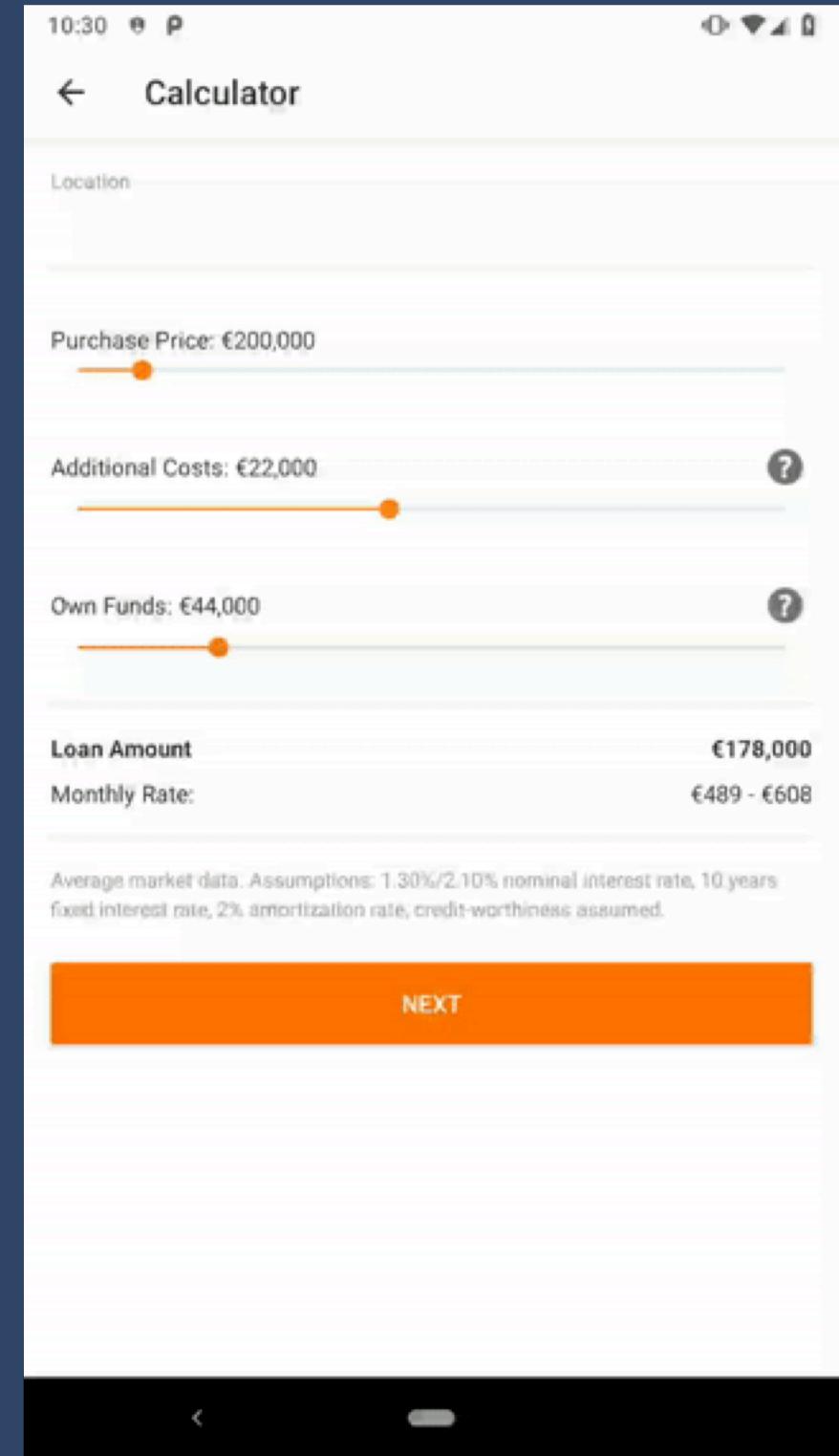


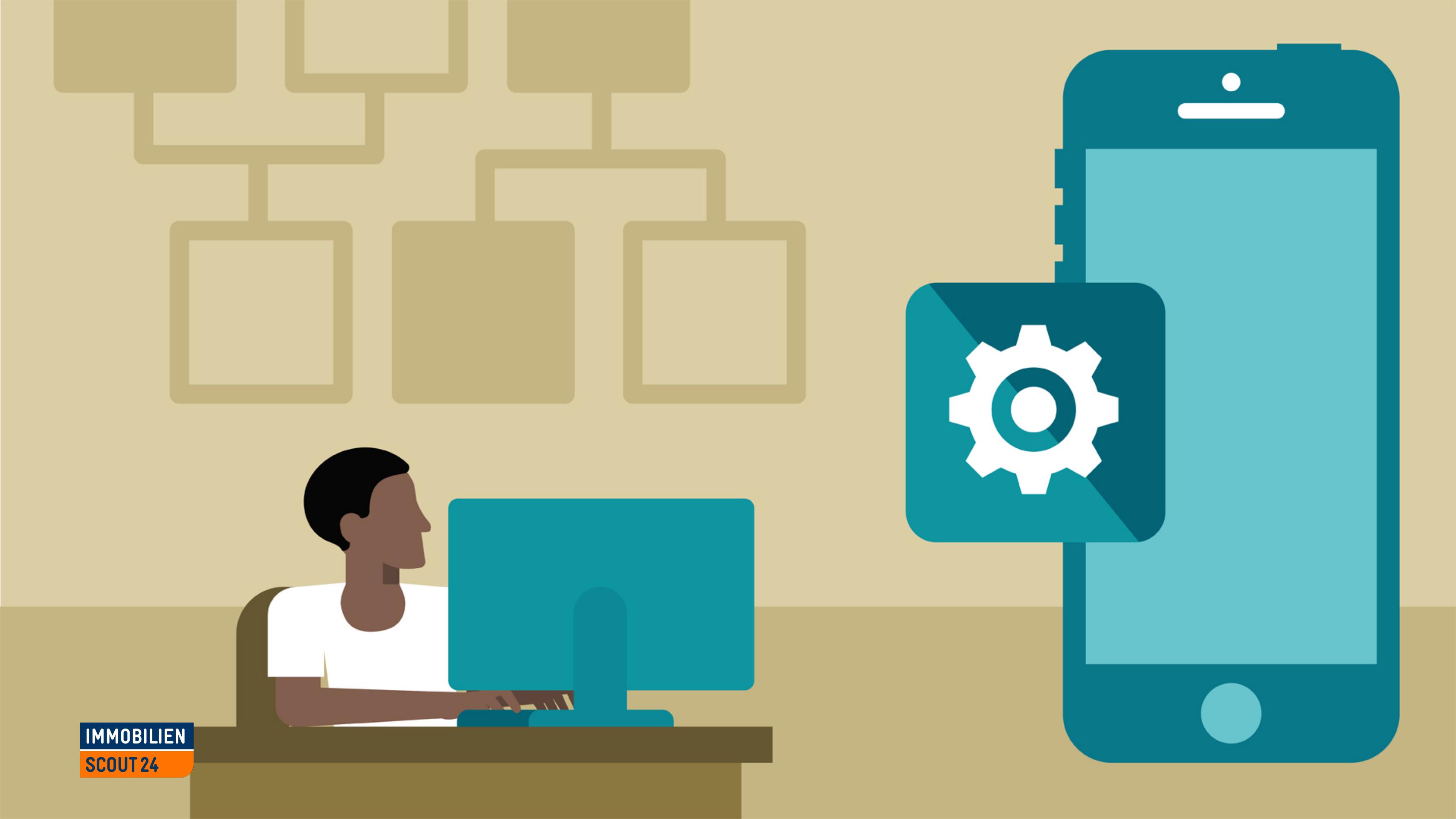
**IMMOBILIEN
SCOUT24**



The Monolith

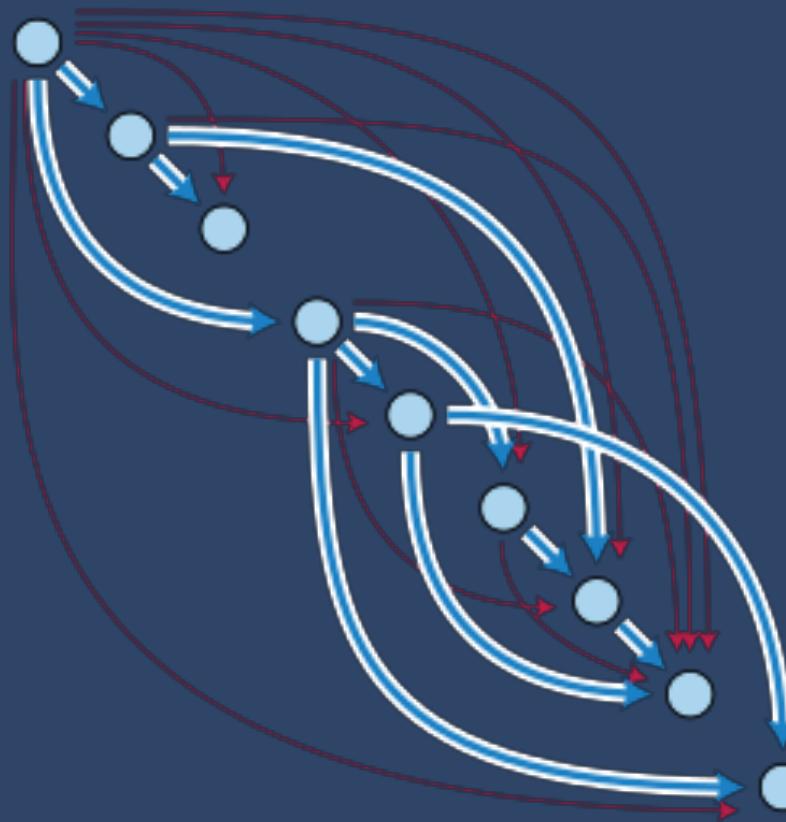
Location Input





IMMOBILIEN
SCOUT24

Dependency Graph

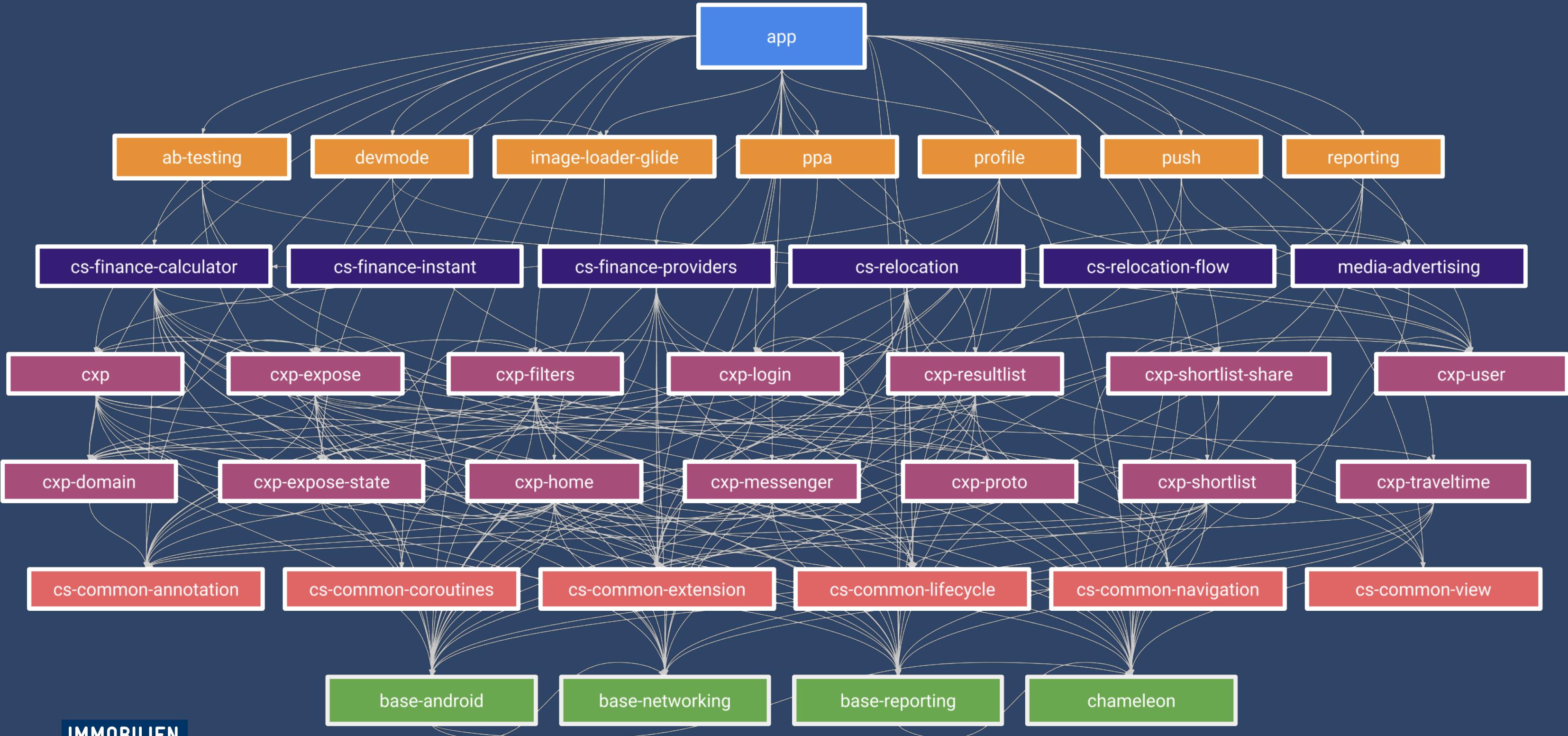


@Component

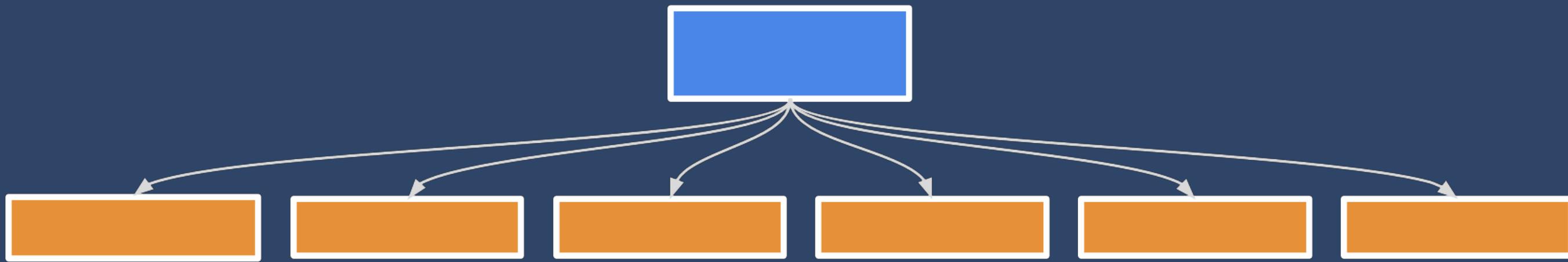
app



search



Flat Hierarchy

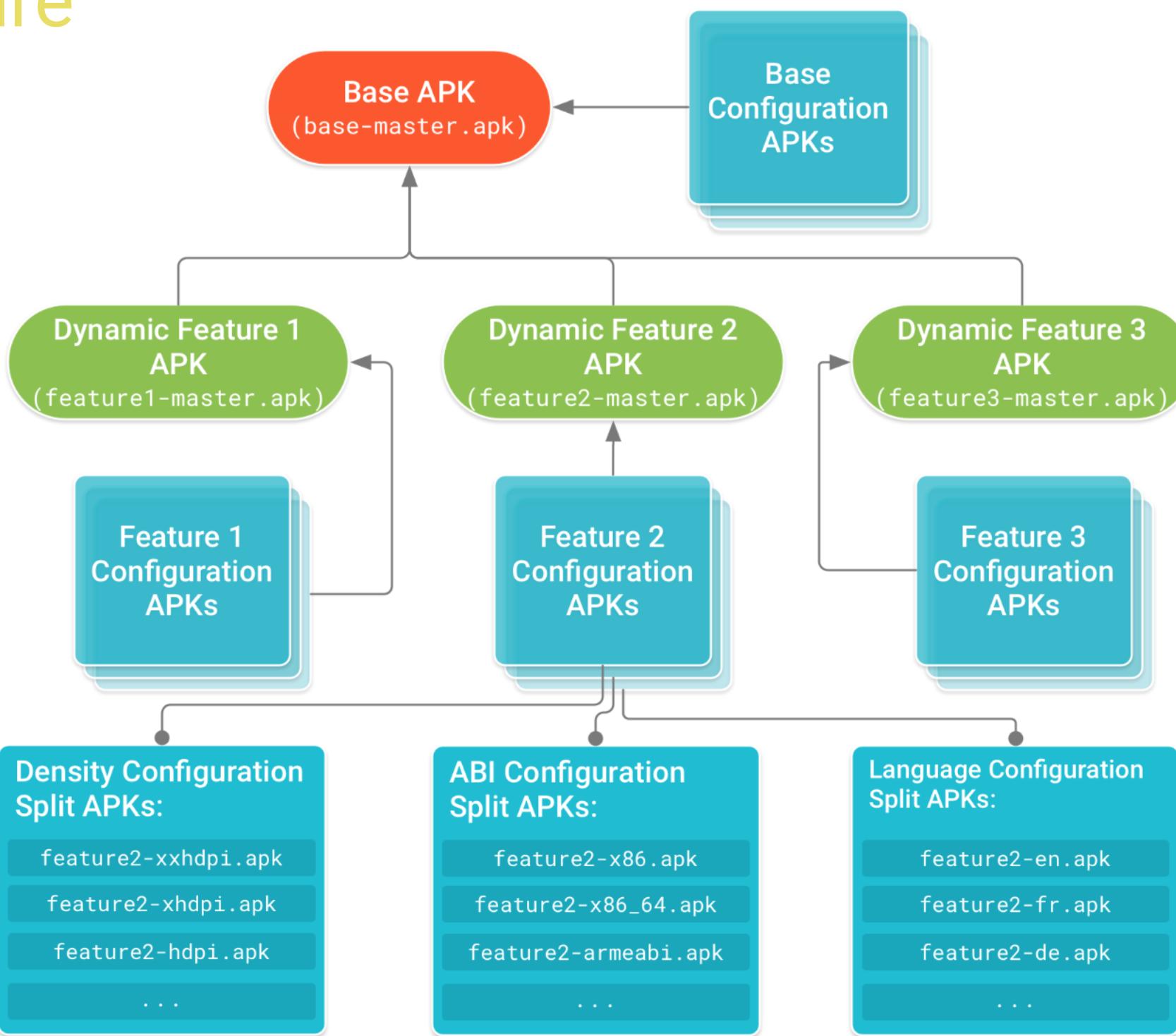






Android App Bundle

Dynamic Feature





IMMOBILIEN
SCOUT 24

Optimisations

- Code minification (`minifyEnabled`)
- Resource shrinking (`shrinkResources`)
- Optimised SVG assets (bit.ly/2SeZePQ)
- Configuration splits (`splits { }`)
- Third party SDK's 😱

android [~/Projects/scout/android] - .../cs-finance-instant/build/outputs/apk/release/cs-finance-instant-release.zip

Project: de.is24.mobile.finance (version ?)

Zip file size: 4.5 MB

Compare with previous APK...

File

	Raw File Size	Download Size	% of Total Download size
cs-finance-calculator-release-unsigned.apk	5 MB	4.5 MB	100%
classes.dex	3.3 MB	3.3 MB	73.3%
res	745.2 KB	730.1 KB	15.8%
classes2.dex	209.3 KB	209.3 KB	4.5%
resources.arsc	592.3 KB	127.2 KB	2.7%
kotlinx	66.9 KB	66.9 KB	1.4%
kotlin	63.4 KB	63.3 KB	1.4%
okhttp3	33.2 KB	33.2 KB	0.7%
META-INF	5.1 KB	5.1 KB	0.1%
AndroidManifest.xml	1.4 KB	1.4 KB	0%
play-services-basement.properties	53 B	53 B	0%
play-services-tasks.properties	51 B	51 B	0%
play-services-maps.properties	50 B	50 B	0%
play-services-base.properties	50 B	50 B	0%

Load Proguard mappings...

This dex file defines 8748 classes with 55160 methods, and references 65503 methods.

Class	Defined Methods	Referenced Methods	Size
android	19339	25601	2.5 MB
support	18238	20314	2.2 MB
v4	7085	7615	837.3 KB
v7	5951	6843	724.1 KB
design	2905	3351	364.7 KB
constraint	784	910	150 KB
transition	763	808	97.5 KB
graphics	318	335	43.8 KB
customtabs	192	211	24.9 KB
v13	57	57	5.6 KB
annotation	34	35	5.3 KB
asynclayoutinflater	12	12	3.5 KB
compat	12	12	3.5 KB
coordinatorlayout	12	12	3.9 KB
coreui	12	12	3.9 KB
coreutils	12	12	3.5 KB

Device File Explorer

IMMOBILIEN Build SonarLint Logcat Version Control Terminal TeamCity Event Log

finished in 11 s 664 ms (53 minutes ago) Git: develop Context: <no context>

SCOUT24 494 of 3005M

Third Party SDKs



Navigation

Intent(context, CalculatorActivity.class.java)

Navigation

~~Intent(context, CalculatorActivity.class.java)~~

App Links

```
<activity android:name=".calculator.CalculatorActivity">
    <intent-filter>

        <action android:name="android.intent.action.VIEW"/>

        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.BROWSABLE"/>

        <data android:host="www.immobilienscout24.de"
              android:path="/baufinanzierung/finanzierungsrechner/"
              android:scheme="https"/>

    </intent-filter>
</activity>
```

```
<activity android:name=".calculator.CalculatorActivity">
    <intent-filter>

        <action android:name="android.intent.action.VIEW"/>

        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.BROWSABLE"/>

        <data android:host="www.immobilienscout24.de"
              android:path="/baufinanzierung/finanzierungsrechner/"
              android:scheme="https"/>

    </intent-filter>
</activity>
```

Intent Extras

```
val intent = Intent(context, CalculatorActivity::class.java)
intent.putExtra("location", "Berlin")
```

!

Query Params

```
intent.data?.getQueryParameter("location")
```

?

Query Params

```
val geocode = intent.data?.getQueryParameter("location")
val label = intent.data?.getQueryParameter("label")
```

```
val region = if (geocode != null && label != null) Region(geocode, label)
else null
```

```
val region = FinanceRegion(  
    intent.data?.getQueryParameter("location"),  
    intent.data?.getQueryParameter("label")  
)  
  
data class Region(val geocode: String, val label: String) {  
  
    companion object {  
  
        operator fun invoke(  
            geocode: String?,  
            label: String?  
        ) : Region(  
            geocode ?: "1276001010",  
            label ?: "Berlin - Mitte (Mitte)"  
        )  
    }  
}
```

Verify Site Associations

```
keytool -list -v -keystore release.keystore
```

```
[ {  
    "relation": ["delegate_permission/common.handle_all_urls"],  
    "target": {  
        "namespace": "android_app",  
        "package_name": "de.is24.android",  
        "sha256_cert_fingerprints": [...]  
    }  
} ]
```



Assistant

App Links Assistant



Gradle

!!! Assistant

App Links Assistant

Android App Links enable your users to launch directly into your app when they click on URLs that your app supports and they can also make your app content searchable.

The App Links Assistant will walk you through how to implement Android App Links below.

① Add URL intent filters

Use the URL Mapping editor to easily add URL intent filters to your Activities.

[Open URL Mapping Editor](#)

② Add logic to handle the intent

When the system starts the activity through the intent filter, you can use the data provided by the intent to determine your app's response.

Select each URL-mapped activity and insert the template codes. You can then add your own logic to handle the intent as appropriate.

[Select Activity](#)

③ Associate website

Associate your app with your website through a Digital Asset Links file.

[Open Digital Asset Links File Generator](#)

④ Test on device or emulator

Test your implementation of Android App Links by simulating launching a URL on a device or emulator.

[Test App Links](#)

Emulator Support



Topeka

github.com/googlesamples/android-topeka

Plaid

github.com/nickbutcher/plaid

Next Steps

Cheers! 🍻

