

C . e . S . A . R
centro de estudos e sistemas
avançados do recife



THE DEVELOPER'S CONFERENCE



**37% das 1+ milhão de
perguntas em
Android estão sem
resposta aceita!**

O que fazer quando
a **sua** é uma delas?



Que tal navegar pelo Android Studio?

Stubs!



```
6 package android.app.job;
7
8 import java.util.List;
9
10 @SystemService(value=android.content.Context.JOB_SCHEDULER_SERVICE) public abstract class JobScheduler {
11     public static final int RESULT_FAILURE = 0;
12     public static final int RESULT_SUCCESS = 1;
13
14     public JobScheduler() { throw new RuntimeException("Stub!"); }
```

Olhar pela
10^a vez os
docs da
Google?



E agora?



Pq navegar no AOSP vai te levar ao próximo nível?

Palestra TDC SP 19 - Trilha Android

17.07.19



THE DEVELOPER'S
CONFERENCE

Palestrantes

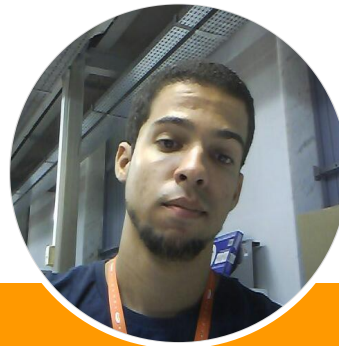


Rodrigo Perazzo
Tech Lead

rodrigo.perazzo 

@rodrigoperazzo 

@PerazzoZe 



Victor Rattis
Eng SW

victorrattis 

@victorrattis 

@victorRattis 

5+ anos em Android Embarcado



KitKat 4.4



Lollipop 5.0



Marshmallow 6.0



Nougat 7.0



Oreo 8.0



Pie 9.0



Cada vez mais Restrições

Dangerous Permissions

Android M

Background Limits

Android O

Background Optimizations

Android N

Non-SDK Interfaces

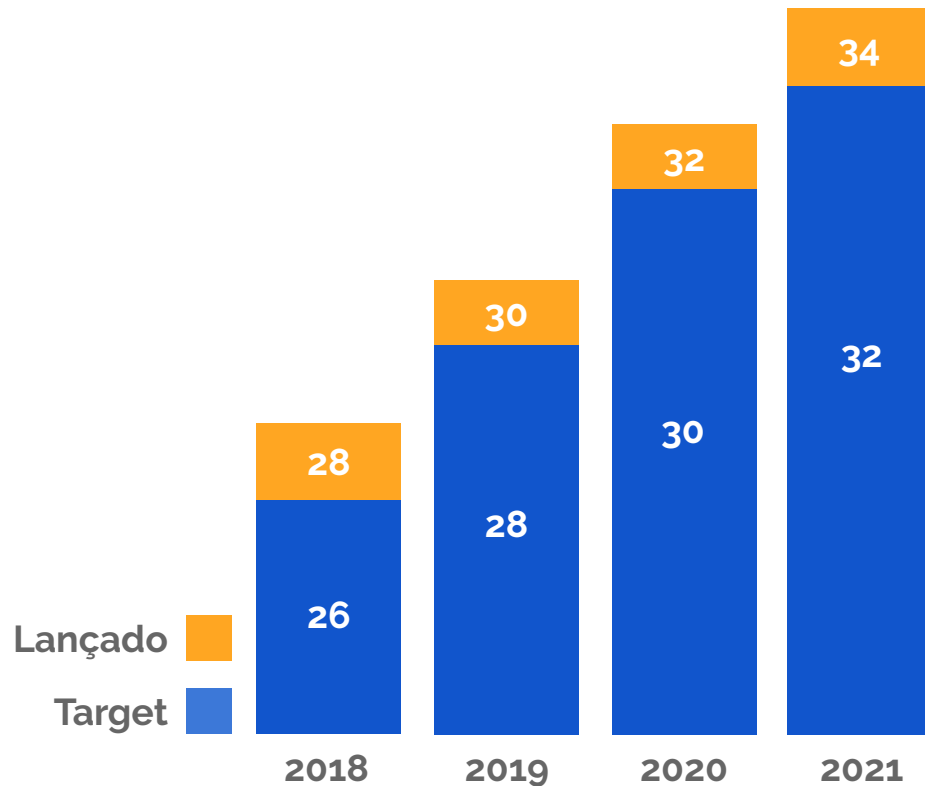
Android P

Target SDK na Google Play

Apps devem atualizar seu SDK

1º de agosto de 2019 novos apps
devam ter o target SDK pelo menos **Android 9.0**

1º de novembro de 2019 atualização
de apps devem ter target SDK para **Android 9.0**



AOSP

אנדרואיד
open source project





Jake Wharton ✓

@JakeWharton

Following

So Q branched from **internal master** on April 4th which means if I landed a new API into master on April 5th its source wouldn't be visible in **AOSP** for ~16 months. If you wanted to just file feedback on the API you wouldn't even see it for a year.

The Android **"Open Source"** Project

5:59 PM - 5 Jun 2019

29 Retweets 181 Likes



7

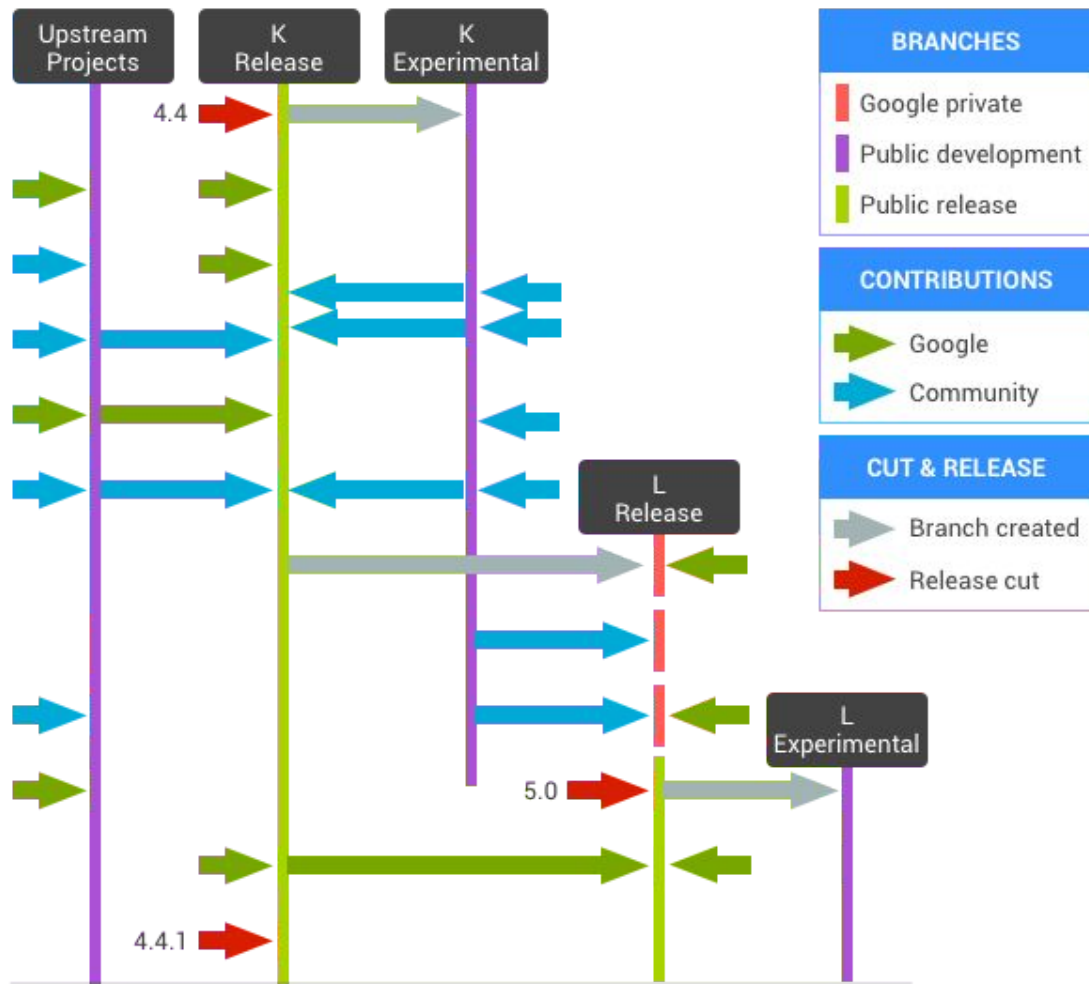


29



181





Você pode baixá-lo, mas...

Ambiente

Ferramenta **Repo**
para gerenciar os
600+ repositórios Git

Tamanho

60 GB de código
fonte

Navegação

Navegação limitada
ao terminal do Linux

Android **xRef** Pie 9.0.0_r3

> OPERSYS  TM Android android-9.0.0_r45

ANDROID SOURCE

Pie - 9.0.0_r3
Oreo - 8.1.0_r33
Oreo - 8.0.0_r4
Nougat - 7.1.2_r36
Nougat - 7.1.1_r6
Nougat - 7.0.0_r1
Marshmallow - 6.0.1_r10
Marshmallow - 6.0.0_r5
Marshmallow - 6.0.0_r1
Lollipop - 5.1.1_r6
Lollipop - 5.1.0_r1
Lollipop - 5.0.0_r2
KitKat - 4.4.4_r1
KitKat - 4.4.3_r1.1
KitKat - 4.4.2_r2
KitKat - 4.4.2_r1
KitKat - 4.4
JellyBean - 4.3
JellyBean - 4.2.2
JellyBean - 4.2
JellyBean - 4.1.2
JellyBean - 4.1.1
ICS - 4.0.4
ICS - 4.0.3
Gingerbread - 2.3.7
Gingerbread - 2.3.6
Froyo - 2.2.3
Eclair - 2.1
Donut - 1.6

News

- **2018-08-11** - New Index: Pie - 9.0.0_r3
- **2018-06-19** - New Index: Oreo - 8.1.0_r33
- **2017-12-22** - New Index: Nougat - 7.1.2_r36
- **2017-09-19** - New Index: Oreo - 8.0.0_r4
- **2016-12-22** - New Index: Nougat - 7.1.1_r6
- **2016-08-24** - New Index: Nougat - 7.0.0_r1
- **2016-01-08** - New Index: Marshmallow - 6.0.1_r10
- **2016-01-07** - New Index: Marshmallow - 6.0.0_r5
- **2015-10-07** - New Index: Marshmallow - 6.0.0_r1
- **2015-07-03** - New Index: Lollipop - 5.1.1_r6
- **2015-03-13** - New Index: Lollipop - 5.1.0_r1
- **2014-09-06** - New Index: Lollipop - 5.0.0_r2
- **2014-07-14** - New Index: KitKat - 4.4.4_r1
- **2014-06-06** - New Index: KitKat - 4.4.3_r1.1
- **2014-03-25** - New Index: KitKat - 4.4.2_r2
- **2014-01-08** - New Index: KitKat - 4.4.2_r1
- **2013-11-01** - New Index: KitKat - 4.4
- **2013-10-31** - New Index: Gingerbread - 2.3.7
- **2013-09-30** - New Index: ICS - 4.0.3
- **2013-07-25** - New Index: JellyBean - 4.3
- **2013-04-22** - New Index: JellyBean - 4.1.2
- **2013-25-02** - New Index: JellyBean - 4.2.2
- **2012-11-21** - New Index: JellyBean - 4.2
- **2012-08-22** - New Index: Froyo - 2.2.3
- **2012-08-22** - New Index: Eclair - 2.1
- **2012-08-22** - New Index: Donut - 1.6
- **2012-07-18** - New Index: JellyBean - 4.1.1
- **2012-06-13** - New Index: ICS - 4.0.4
- **2011-09-02** - New Index: Gingerbread - 2.3.6

KERNEL SOURCE

Kernel - 3.18
Kernel - 3.14
Kernel - 3.10
Kernel - 3.4
Kernel - 3.3
Kernel - 3.0
Kernel - 2.6.39



CONTACT

Rodrigo Chioffi
[contact\(at\)androidxref.com](mailto:contact(at)androidxref.com)

AndroidXRef Pie 9.0.0_r3

Full Search

Definition

Symbol

File Path

History

In Project(s)

dalvik
developers
development
device
external
frameworks

Source Tree : [xref/](#)

Textos. Ideal para logs de erro!

AndroidXRef Pie 9.0.0_r3

Full Search

Definition

Symbol

File Path

History

In Project(s)

- dalvik
- developers
- development
- device
- external
- frameworks

Source Tree : [xref/](#)

Classes, métodos e variáveis

AndroidXRef Pie 9.0.0_r3

Full Search

Definition

Symbol

File Path

History

In Project(s)

- dalvik
- developers
- development
- device
- external
- frameworks

Source Tree : [xref/](#)

Diretórios e tipos de arquivo

AndroidXRef Pie 9.0.0_r3

Full Search

Definition

Symbol

File Path

History

In Project(s)

- dalvik
- developers
- development
- device
- external
- frameworks

Source Tree : [xref/](#)

Commit messages

AndroidXRef Pie 9.0.0_r3

Full Search

Definition

Symbol

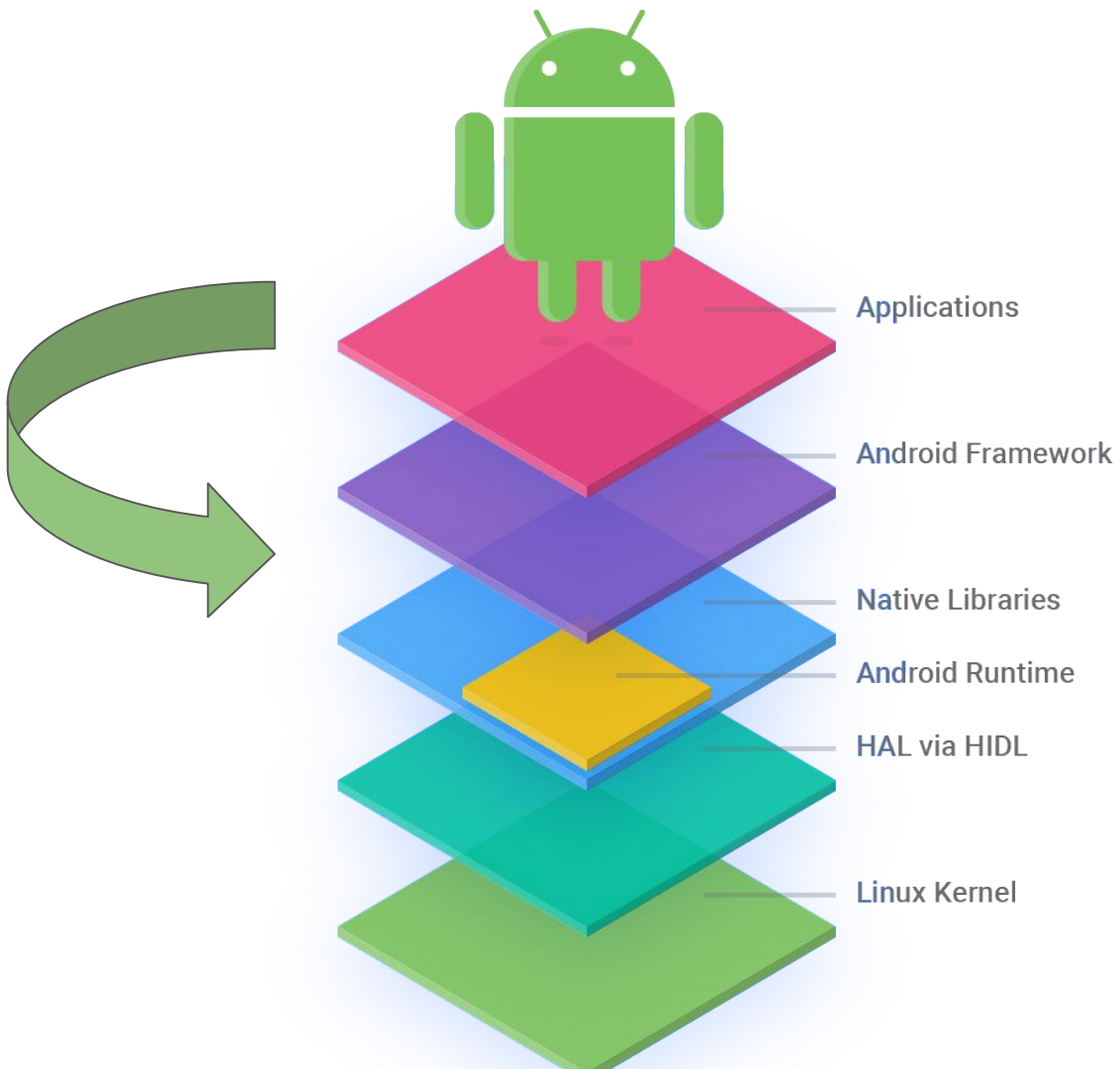
File Path

History

In Project(s)

- dalvik
- developers
- development
- device
- external
- frameworks

Source Tree : [xref/](#)



Visível

App



SDK



Código
Interno

Invisível

Visível

App

SDK

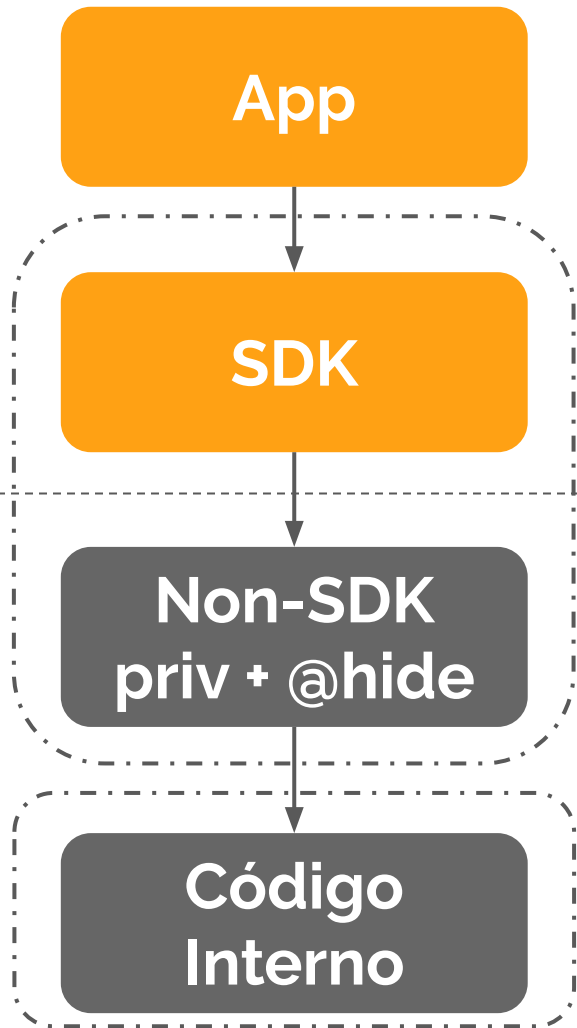
framework.jar

Invisível

Non-SDK
priv + @hide

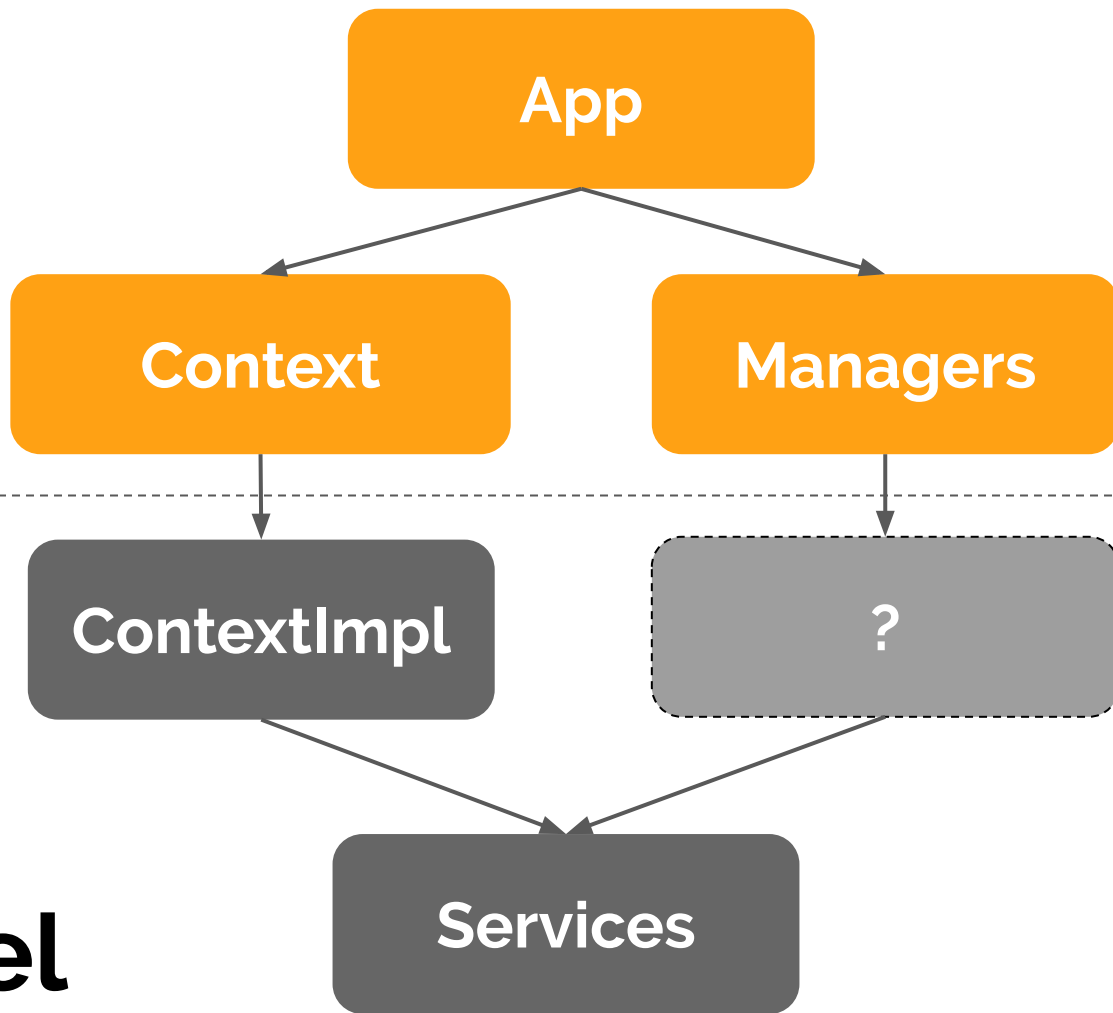
Código
Interno

services.jar
libandroid*.so



```
166
167 /**
168  * Broadcast intent action indicating that the GPS has either been
169  * enabled or disabled. An intent extra provides this state as a boolean,
170  * where {@code true} means enabled.
171  * @see #EXTRA_GPS_ENABLED
172  *
173  * @hide
174  */
175 public static final String GPS_ENABLED_CHANGE_ACTION =
176     "android.location.GPS_ENABLED_CHANGE";
```

Visível



Invisível

```
429  /**
430   * Returns a list of the names of all known location providers.
431   * <p>All providers are returned, including ones that are not permitted to
432   * be accessed by the calling activity or are currently disabled.
433   *
434   * @return list of Strings containing names of the provider
435   */
436  public List<String> getAllProviders() {
437      try {
438          return mService.getAllProviders();
439      } catch (RemoteException e) {
440          throw e.rethrowFromSystemServer();
441      }
442  }
```

Visível

App



Classe do
SDK

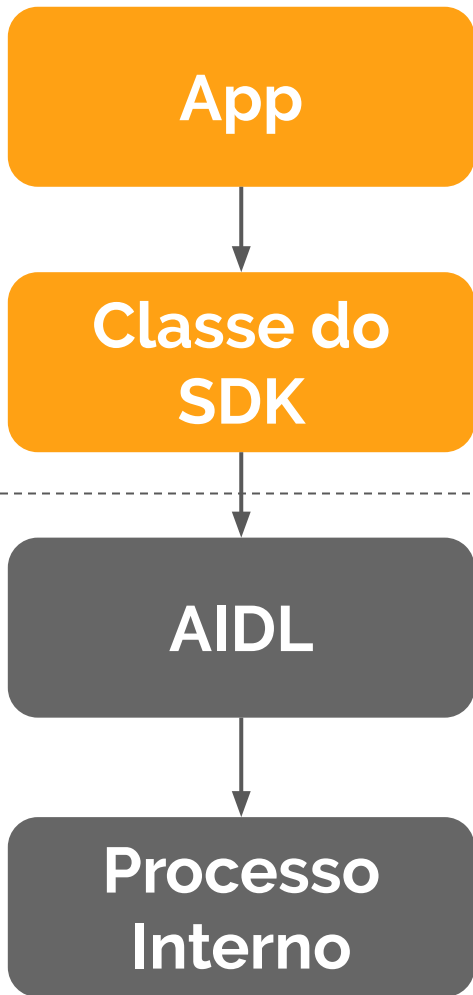


AIDL



Processo
Interno

Invisível



Android Interface Definition Language

- Arquivos `.aidl`
- Nomes: `IManager` / `IService` / ...
- Implementado com: `extends <NOME>.Stub`



Case Real (SDK 28):
“Porque o
JobScheduler não
está funcionando?!”

O que o app estava fazendo:

- Agendava um job num BroadcastReceiver para executar assim que possível.
- O Job demorava muito para ser executado (horas!)

AndroidXRef Pie 9.0.0_r3

Home Sort by: [last modified time](#) | **relevance** | [path](#)

Full Search

Definition 

Symbol

File Path

History

[Search](#)

[Clear](#)

[Help](#)

In Project(s)

[select all](#)

[invert selection](#)

dalvik
developers
development
device
external
frameworks

Searched **defs:JobScheduler** (Results 1 - 1 of 1) sorted by relevance

[/frameworks/base/core/java/android/app/job/](#)

H A D	JobScheduler.java	40	* JobScheduler with {@link #schedule(JobInfo)}. When the criteria declared are met, the
		49	* can be run at any moment depending on the current state of the JobScheduler 's internal queue.
		61	public abstract class JobScheduler { <i>class</i>

```
53 * duration of the job.
54 * </p>
55 * <p>You do not
56 * instantiate this class directly; instead, retrieve it through
57 * {@link android.content.Context#getSystemService
58 * Context.getSystemService(Context.JOB_SCHEDULER_SERVICE)}.
59 */
60 @SystemService(Context.JOB_SCHEDULER_SERVICE) ←
61 public abstract class JobScheduler {
62     /** @hide */
63     @IntDef(prefix = { "RESULT_" }, value = {
64         RESULT_FAILURE,
65         RESULT_SUCCESS,
66     })
67     @Retention(RetentionPolicy.SOURCE)
68     public @interface Result {}
69
```

AndroidXRef Pie 9.0.0_r3

Home Sort by: last modified time | **relevance** | path

Full Search

Definition

Symbol ←

File Path

History

Search

Clear

Help

In Project(s)

select all

invert selection

dalvik
 developers
 development
 device
 external
 frameworks

/frameworks/base/core/java/android/app/

HAD [SystemServiceRegistry.java](#) 816 registerService(Context.**JOB_SCHEDULER_SERVICE**, JobScheduler.class,
 820 IBinder b = ServiceManager.getServiceOrThrow(Context.**JOB_SCHEDULER_SERVICE**);

/frameworks/base/services/core/java/com/android/server/content/

HAD [SyncManager.java](#) 519 Context.**JOB_SCHEDULER_SERVICE**);

/frameworks/base/services/core/java/com/android/server/job/

HAD [JobSchedulerService.java](#) 1203 publishBinderService(Context.**JOB_SCHEDULER_SERVICE**, mJobSchedulerStub);

```
2533 @Override
2534 public int schedule(JobInfo job) throws RemoteException {
2535     if (DEBUG) {
2536         Slog.d(TAG, "Scheduling job: " + job.toString());
2537     }
2538     final int pid = Binder.getCallingPid();
2539     final int uid = Binder.getCallingUid();
2540     final int userId = UserHandle.getUserId(uid);
2541
2542     enforceValidJobRequest(uid, job);
2543     if (job.isPersisted()) {
2544         if (!canPersistJobs(pid, uid)) {
2545             throw new IllegalArgumentException("Error: requested job be persisted without"
2546                 + " holding RECEIVE_BOOT_COMPLETED permission.");
2547         }
2548     }
2549
2550     validateJobFlags(job, uid);
2551
2552     long ident = Binder.clearCallingIdentity();
2553     try {
2554         return JobSchedulerService.this.scheduleAsPackage(job, null, uid, null, userId,
2555             null);
2556     } finally {
2557         Binder.restoreCallingIdentity(ident);
2558     }
2559 }
```

```
827
828 public int scheduleAsPackage(JobInfo job, JobWorkItem work, int uId, String packageName,
829     int userId, String tag) {
830     try {
831         if (ActivityManager.getService().isAppStartModeDisabled(uId,
832             job.getService().getPackageName())) {
833             Slog.w(TAG, "Not scheduling job " + uId + ":" + job.toString()
834                 + " -- package not allowed to start");
835             return JobScheduler.RESULT_FAILURE;
836         }
837     } catch (RemoteException e) {
838     }
839 }
```



```
885     }
886     startTrackingJobLocked(jobStatus, toCancel);
887     StatsLog.write_non_chained(StatsLog.SCHEDULED_JOB_STATE_CHANGED,
888         uId, null, jobStatus.getBatteryName(),
889         StatsLog.SCHEDULED_JOB_STATE_CHANGED__STATE__SCHEDULED,
890         JobProtoEnums.STOP_REASON_CANCELLED);
891 }
```



```
1268     */
1269 private void startTrackingJobLocked(JobStatus jobStatus, JobStatus lastJob) {
1270     if (!jobStatus.isPreparedLocked()) {
1271         Slog.wtf(TAG, "Not yet prepared when started tracking: " + jobStatus);
1272     }
1273     jobStatus.enqueueTime = sElapsedRealtimeClock.millis();
1274     final boolean update = mJobs.add(jobStatus);
1275     if (mReadyToRock) {
1276         for (int i = 0; i < mControllers.size(); i++) {
1277             StateController controller = mControllers.get(i);
1278             if (update) {
1279                 controller.maybeStopTrackingJobLocked(jobStatus, null, true);
1280             }
1281             controller.maybeStartTrackingJobLocked(jobStatus, lastJob);
1282         }
1283     }
1284 }
1285 }
```


Regras de agendamento

```
1136
1137 // Create the controllers.
1138 mControllers = new ArrayList<StateController>();
1139 mControllers.add(new ConnectivityController(this));
1140 mControllers.add(new TimeController(this)); ←
1141 mControllers.add(new IdleController(this));
1142 mBatteryController = new BatteryController(this);
1143 mControllers.add(mBatteryController);
1144 mStorageController = new StorageController(this);
1145 mControllers.add(mStorageController);
1146 mControllers.add(new BackgroundJobsController(this)); ←
1147 mControllers.add(new ContentObserverController(this));
1148 mDeviceIdleJobsController = new DeviceIdleJobsController(this);
1149 mControllers.add(mDeviceIdleJobsController);
1150
```

```
67 @Override
68 public void maybeStartTrackingJobLocked(JobStatus jobStatus, JobStatus lastJob) {
69     updateSingleJobRestrictionLocked(jobStatus, UNKNOWN);
70 }
```



```
189 boolean updateSingleJobRestrictionLocked(JobStatus jobStatus, int activeState) {
190
191     final int uid = jobStatus.getSourceUid();
192     final String packageName = jobStatus.getSourcePackageName();
193
194     final boolean canRun = !mAppStateTracker.areJobsRestricted(uid, packageName,
195         (jobStatus.getInternalFlags() & JobStatus.INTERNAL_FLAG_HAS_FOREGROUND_EXEMPTION)
196         != 0);
197
198     final boolean isActive;
199     if (activeState == UNKNOWN) {
200         isActive = mAppStateTracker.isUidActive(uid);
201     } else {
202         isActive = (activeState == KNOWN_ACTIVE);
203     }
204     boolean didChange = jobStatus.setBackgroundNotRestrictedConstraintSatisfied(canRun);
205     didChange |= jobStatus.setUidActive(isActive);
206     return didChange;
207 }
```

Estado do app?

AndroidXRef Pie 9.0.0_r3

xref: /frameworks/base/services/core/java/com/android/server/j

Home **History** | Annotate | Line# | Navigate | Download

Date	Author	Comments (Show)
01-Mar-2018	Christopher Tate <ctate@google.com>	Jobs in 'active' apps are not subject to standby Along the way we turn down the "app idle" tracking in the jbo scheduler: it's now redundant with standby bookkeeping, and was muddying the waters. Bug: 74017233 Test: atest CtsJobSchedulerTestCases Change-Id: I3220e6ef97f153b3f718e267a0dc4fbd3b926316

App Standby Bucket

- Feature de gerenciamento de bateria integrada no Android P
- Prioriza requisições por frequência de uso:
 - Active: App is currently being used or was very recently used
 - Working set: App is in regular use
 - Frequent: App is often used, but not every day
 - Rare: App is not frequently used

Solução!

- Explicar a restrição e mudar a UX do aplicativo!
 1. Pedir para o **usuário manualmente colocar** o app na **Power Save**
 2. Executar o processamento como um **Foreground Service**

Nem todo código vai resolver seu problema.

```
5129 public int getSimCount() {  
5130     // FIXME Need to get it from Telephony Dev Controller when that gets implemented!  
5131     // and then this method shouldn't be used at all!  
5132     if(isMultiSimEnabled()) {  
5133         return 2;  
5134     } else {  
5135         return 1;  
5136     }  
5137 }
```

Porque navegar no AOSP?

+ Velocidade

Para atualizar o app às novas versões

+ Autonomia

Para resolver problemas “sem solução”
Para aprender novas features do Android

+ Facilidade

Para contribuir com a comunidade

Dúvidas?



Obrigado!

Palestra TDC SP 19 - Trilha Android

17.07.19



THE DEVELOPER'S
CONFERENCE