Scheduling Background Work with JobScheduler

Jim Wilson
MOBILE SOLUTIONS DEVELOPER & ARCHITECT

@hedgehogjim   blog.jwhh.com

What to Expect from This Module

Job Scheduler Overview

Creating Job Implementation Class

Job Information, Criteria, and Scheduling

Launching Background Work

When Work Needs to Stop

Choosing Between Services & Job Scheduler

Background Work Challenges

Background work is a common need

- Long running work
- Don’t want that work to affect user’s foreground experience

Presents significant challenges

- Challenges for the system
- Challenges for the developer

Background Work System Challenges

Many apps now run background work

- Can affect user’s foreground experience

Can create system-wide impact

- Excessive memory usage
- High CPU usage
- Rapid battery drain

Background Work Developer Challenges

Background work often has run criteria

- Device is connected to network
- Device is plugged in
- Run at regular time intervals

Not always easy to get right

- Need to determine if currently available
- May need to wait until available
- Detect when no longer available

Background Work Challenges and Job Scheduler

Job scheduler

- Addresses background work challenges

Addresses system challenges

- Gives system more control of when background work is run

Addresses developer challenges

- Handles run criteria details

Job Scheduler

Introduced in API 21

- Android 5.0 and newer
- Supported by majority of devices

Preferred way to do background work

- Allows system to manage resource use
- Limits impact on user experience
- Limits impact on device

Useful in many common service scenarios

- Caveat: work may not start immediately

Job Scheduler

Work is handled as a "job"

- A job is created in steps

Implement the job

- Component that handles doing the work

Build information about the job

- Includes job run criteria

Schedule the job

- Pass job information to job scheduler

Job Implementation

Jobs implemented as a special service

- Must extend the JobService class

Override onStartJob method

- Called to indicate job should start

Override onStopJob method

- Called to indicate job should stop
- Usually means that criteria is no longer being met

Job Implementation

Service must appear in manifest

- Mostly the same as other services

Must be marked with a special permission

- Use android:permission attribute
- android.permission.BIND_JOB_SERVICE

Job Information

JobInfo class

- An app-defined job ID
- The job implementation component
- Job criteria
- Job-defined data

Created using the builder pattern

- Use JobInfo.Builder class

Job Information

Constructing JobInfo.Builder

- App-defined job ID
- Job implementation component

Job run criteria

- Supports a variety of criteria
- Must set at least one

Job Information

Network criteria

- Needs a network connection
- Metered/unmetered connection

Power criteria

- Device is charging
- Battery not low

Device state criteria

- Device is idle
- Storage is not low

Job Information

Timing criteria

- Delay starting
- Run at regular intervals

Override deadline

- Maximum time to wait
- Will run even if other criteria not met

Job Information

Can include job-defined data

- Set as extras
- Store in a PersistableBundle
- Implementation component can retrieve

Scheduling the Job

Get a reference to JobScheduler

- Job scheduler is system service
- Use Context.getSystemService
- Pass JOB_SCHEDULER_SERVICE

Call JobScheduler.schedule method

- Pass in JobInfo
- Job will run after criteria is met
- Not necessarily as soon as criteria is met

Performing Job Work

onStartJob method

- Called to indicate work should begin

onStartJob runs on main app thread

- Perform no-long running work

Dispatch work to a different thread

- Use AsyncTask
- Send to a Handler on a different thread

Performing Job Work

Coordinating work with job scheduler

- Indicate background work was started
- Indicate when work is done

onStartJob return value

- Return to true to indicate that work is being performed on another thread

jobFinished method

- Call to indicate work is done
- Can optionally have job rescheduled

Performing Job Work

Job configuration and identification data

- JobParameters class
- Job scheduler passes to onStartJob
- Includes job extras

When Work Needs to be Stopped

Work may need to stop before complete

- Usually because criteria is no longer met

onStopJob method

- Called to indicate job needs to stop
- Return true to have job rescheduled

Stopping work

- Details of stoppage are job specific
- Do not call jobFinished method

Choosing Between Services and JobScheduler

Services

- App handles run criteria details
- App controls execution
- Start immediately

Job scheduler

- System handles run criteria
- System controls execution
- System decides when job starts

Choosing Between Services and JobScheduler

Prefer job scheduler

- Works well for most scenarios
- Allows Android to provide better management of system resources

Android 8 (API 26)

- Puts limits on execution of services

Summary

Job scheduler

- Supported on Android 5.0 and newer
- Preferred way to do background work
- Allows system to better manage resource use

Manages details of job criteria

- Starts job only after criteria met
- Stops job if criteria no longer being met

Summary

JobInfo class

- App defined job ID
- Job implementation component
- Job criteria
- Job-defined data

Constructed using builder pattern

- JobInfo.Builder class

Summary

Jobs implemented as a special service

- Must extend the JobService class
- Must be marked in manifest with BIND_JOB_SERVICE permission

Summary

Override onStartJob method

- Called to indicate work should begin
- Runs on main app thread
- Must dispatch work to different thread

Override onStopJob method

- Called to indicate work should stop

jobFinished method

- Call to indicate work is complete
- But don’t call when work stopped by onStopJob method