

Alarm Service

Служба **Alarm Service** используется для отправки пользователю разовых или повторяющихся сообщений в заданное время. Таким образом вы сможете создавать различные планировщики, будильники, реализовать выполнение регулярных сетевых запросов, запуска трудоёмких или дорогих операций, запланированных на определенное время и другие приложения, которые должны срабатывать по расписанию.

Важно помнить, что механизм сигнализации не зависит от конкретного приложения, это значит, что прописанные намерения сработают в определённое время или с заданным интервалом. Так как сигнализация устанавливается за пределами вашего приложения, она может быть использована для вызова событий, принадлежащих приложению, или для запуска самого приложения, даже если оно закрыто. Сигнализация может стать мощным инструментом в сочетании с приёмниками широковещательных намерений.

Таким образом, сигнализация чрезвычайно полезна, когда речь идет о снижении требований к ресурсам, особенно в случае с приложениями, которые работают в фоновом режиме, позволяя останавливать сервисы и отключать таймеры, сохраняя возможность выполнять запланированные действия.

Для планирования операций, которые работают только на протяжении жизненного цикла вашего приложения, вместо механизма сигнализации лучше использовать класс `Handler` в сочетании с таймерами и потоками. Это позволит `Android` лучше контролировать системные ресурсы.

Сигнализация в `Android` остается активной даже тогда, когда устройство находится в режиме ожидания и при необходимости может его «пробудить», однако она отменяется каждый раз, когда устройство перезагружается.

Менеджер оповещений `AlarmManager`

Доступ к службе **Alarm Service** осуществляется при помощи объекта **`AlarmManager`** следующим образом:

```
AlarmManager manager = (AlarmManager) getSystemService(
    Context.ALARM_SERVICE);
```

Методы

- **`cancel()`** - удаляет все сигнализации
- **`setTime()`** - устанавливает системное время
- **`setTimeZone()`** - устанавливает временную зону
- **`set()`** - задаёт одноразовую сигнализацию
- **`setRepeating()`** - задаёт повторяющиеся сигнализации с фиксированным временным интервалом
- **`setInexactRepeating()`** - устанавливает повторяющиеся сигнализации без строгого требования к точности периода повторения

Чтобы создать сигнализацию, которая сработает всего один раз, используйте метод **`set()`**, укажите тип сигнализации, время срабатывания и ожидающее намерение, которое должно запуститься. Если время срабатывания, которое вы указали для сигнализации, уже прошло, указанное намерение запустится немедленно.

Параметры

Методы **`set()`**, **`setRepeating()`**, **`setInexactRepeating()`** используют следующие параметры:

- **typeOne** - тип используемого времени (системное или всемирное время UTC), который определяется константами
 - **ELAPSED_REALTIME** - запускает ожидающее намерение, основываясь на времени, которое прошло с момента загрузки устройства, но не с момента выхода из режима ожидания. Это время включает любой временной промежуток, в котором устройство находилось в данном режиме. Обратите внимание, что прошедшее время вычисляется на основании того, когда устройство было загружено. Используется системное время
 - **ELAPSED_REALTIME_WAKEUP** - по прошествии указанного промежутка времени с момента загрузки выводит устройство из спящего режима и запускает ожидающее намерение. Используется системное время
 - **RTC** - запускает ожидающее намерение в указанное время, но не выводит устройство из режима ожидания. Используется всемирное время UTC
 - **RTC_WAKEUP** - выводит устройство из режима ожидания для запуска ожидающего намерения в указанное время. Используется всемирное время UTC
- **triggerTime** - время работы оповещения
- **interval** - интервал между отправкой повторных сигнализаций в миллисекундах. Также можно использовать константы
 - **INTERVAL_DAY**
 - **INTERVAL_HALF_DAY**
 - **INTERVAL_HOUR**
 - **INTERVAL_HALF_HOUR**
 - **INTERVAL_FIFTEEN_MINUTES**
- **operation** - объект **PendingIntent**, определяющий действие, выполняемое при запуске сигнализации. Можно получить через методы **getBroadcast()** или **getService()**

Служба

Для установки сигнализации вам придётся создать собственную службу, наследуясь от базового класса **Service** (либо через приёмник **BroadcastReceiver**, об этом в другом примере):

```
public class AlarmService extends Service {...}
```

Запуск и управление службой происходит при помощи объекта **Intent**.

Установка времени для сигнализаций

Для задания времени работы оповещения необходимо установить его время запуска и добавить к нему длительность работы этого оповещения. Например, нам необходимо, чтобы оповещение отрабатывало 5 секунд после запуска:

```
long time = SystemClock.elapsedRealTime() + 5000;
```

Также можно использовать объект **Calendar**. Например, мы хотим, чтобы продолжительность сигнала оповещения была 10 секунд, а период повторения оповещения был один час:

```
Calendar calendar = Calendar.getInstance();
calendar.setTimeInMillis(System.currentTimeMillis());
calendar.add(Calendar.SECOND, 10);
long time = calendar.getTimeInMillis();
```

```
manager.setRepeating(AlarmManager.RTC_WAKEUP, time,  
    AlarmManager.INTERVAL_HOUR, pendingIntent);
```

Примеры с [Alarm Service](#)