



# Mobile\_logs

Логи - это текстовые файлы, в которых записываются все действия пользователя. Например, какие кнопки он нажимает в приложении и как на это оно реагирует в ответ. Записи в логах формируются в хронологическом порядке. Самая свежая — внизу.

## Уровни логирования

Логи на мобильных устройствах бывают нескольких уровней:

- ERROR,
- WARN,
- INFO,
- DEBUG,
- VERBOSE.

Они представлены по уровню важности — от самого высокого к самому низкому, — и каждый следующий уровень включает в себя предыдущий. Например, VERBOSE содержит в себе логи всех остальных.

**Примечание:** уровни более применимы к логам на Android, потому что именно там такое разделение встречается чаще.

### Error (ERROR)

На этом уровне информируются ошибки работы системы. Записи этого уровня требуют быстрого вмешательства разработчика — на такие ошибки нужно реагировать максимально быстро.

Как пример, такая запись в логе:

*“ SpannableStringBuilder: SPAN\_EXCLUSIVE\_EXCLUSIVE spans cannot have a zero length ”*

Это ошибка, в которой говорится, что строковый элемент span не может быть пустым.

Или вот:

*“ [ZeroHung]zrhung\_get\_config: Get config failed for wp[0x0008] ”*

Эта системная ошибка сообщает, что происходит утечка памяти при взаимодействии с каким-то элементом или приложением.

## Warning (WARN)

На этом уровне отображаются записи, сообщающие о каком-то неожиданном поведении, требующем внимания, или о ситуации, которая незнакома системе.

Например, сообщение ниже — запись из тестового приложения:

```
“ [OMX.hisi.video.decoder.avc] setting nBufferCountActual to 16 failed: -2147483648 “
```

Мы пытаемся декодировать запись в какой-то формат, но его нет. Ошибка сообщает о неуспешной попытке настройки видеоплеера в нужном формате.

Ещё пример:

```
“ BroadcastQueue: Permission Denial: broadcasting Intent ”
```

Эта системная ошибка говорит о сбое в работе одного из виджетов на устройстве.

## Info (INFO)

На этот уровень приходят записи информационного характера, например о работе системы. Допустим, такое сообщение об уровне заряда батареи на устройстве:

```
“ APwBatteryMonitor: screen off start battery: 100 ”
```

А это сообщение говорит о том, что экран устройства был выключен:

```
“ HwBatteryService: intent = Intent { act=android.intent.action.SCREEN_OFF flg=0x58200010 } ”
```

Ещё в логи этого уровня входят запросы от клиента на сервер: хедеры, тело запросов, которые отправляет клиент, и ответы сервера.

```
“ okhttp.OkHttpClient: <-- 200 https://domainname/api/v1/smith/deals (1691ms)
```

```
okhttp.OkHttpClient: server: nginx/1.15.9
```

```
okhttp.OkHttpClient: date: Thu, 23 Sep 2021 19:41:17 GMT
```

```
okhttp.OkHttpClient: content-type: application/json
```

```
okhttp.OkHttpClient: vary: Accept-Encoding
```

```
okhttp.OkHttpClient: strict-transport-security: max-age=15724800; includeSubDomains
```

```
okhttp.OkHttpClient: {"key":{"key":value,"name":""},"key":value,"key":value}
```

```
okhttp.OkHttpClient: <-- END HTTP ”
```

Такие записи могут помочь вам в понимании какого-то бага или в разборе задачи при условии, что вы не можете перехватить трафик или не знаете, какие запросы отправляются на бэкэнд.

## Debug (DEBUG)

Это уровень сообщений, в которых передаётся информация о процессах отладки или шагах работы крупных процессов. Например, в записи ниже сказано, что пользователь нажимал на кнопку уменьшения или увеличения громкости:

```
“ MediaSessionService: dispatchVolumeKeyEvent ”
```

Сначала мы видим запись о самом факте нажатия на кнопку, далее оно расшифровывается подробнее:

```
{ action=ACTION_DOWN, keyCode=KEYCODE_VOLUME_UP }
```

Ещё пример: если ваше приложение использует сокет-сессию, то на уровне DEBUG мы можем увидеть, когда сессия начинается и заканчивается:

```
“ b$b: WebSocket connected ”
```

## Verbose (VERBOSE)

Сообщения такого уровня уточняют или раскрывают действия. Например, у нас есть служба управления окнами на экране приложения. И на уровне Verbose мы можем увидеть подробности её работы.

Открытие окна:

```
WindowManager: addWindow
```

Закрытие окна:

*WindowManager: Removing Window*

На этом уровне мы можем посмотреть системные подробности наших действий. Например, при включении геолокации в записи отобразится текущая геолокация.

*GnssLocationProvider: reportLocation Location [...]*

А меняя звук на устройстве, мы увидим, как растёт или падает значение:

*AudioManager: getStreamVolume streamType: 3 volume: 10*

Каждое нажатие, то есть изменение звука, будет отражаться новым сообщением.

Verbose - уровень самого низкого приоритета. Выбирая такой уровень отображения логов, мы будем видеть записи и со всех предыдущих уровней.

**Примечание:** разработчики приложения самостоятельно покрывают действия логами, определяют уровни, а также какие сообщения какому из них соответствуют.

## Группы логов

- **Crash logs** — файл, в котором хранятся записи только об ошибках экстренного завершения программы — по-простому, когда приложение крашнулось.
- **Logs** — простые логи, или журнал событий. Это файл, в котором хранятся системные записи и ответы устройства на действие пользователя.

## Программы для сбора логов с мобильных устройств

### Программы для снятия логов: Android

Android Studio

Logcat

SDK Platform Tools

### Инструменты снятия логов: iOS

xCode

iMazing

### Структуры файла логово мобильного устройства

Структура зависит от устройства и инструмента для снятия логов, но в основном это:

- Дата
- Время
- Тип ошибки/уровень лога
- Название девайса
- Системы, отправляющие информацию в лог
- Текст лога ошибки