Logi aplikacji

dr Artur Bartoszewski

Logi systemowe

System logów w Android SDK to narzędzie służące do rejestrowania i wyświetlania informacji diagnostycznych podczas działania aplikacji.
Te informacje mogą obejmować komunikaty o błędach, ostrzeżenia, informacje debugowania, a także dowolne inne dane, które deweloper zdecyduje się zarejestrować.

Główne komponenty systemu logów:

- Logcat: Okno w Android Studio, które wyświetla logi w czasie rzeczywistym. Możesz filtrować logi według poziomu ważności (np. błędy, ostrzeżenia), tagów, nazwy aplikacji itp.
- Klasa Log: Klasa w Android SDK, która udostępnia metody do rejestrowania logów, takie jak Log.d() (debug), Log.e() (błąd), Log.w() (ostrzeżenie) itp. Jak działa system logów:

Logi systemowe

- Rejestrowanie logów – programista dodaje instrukcje ***Log*** w kodzie aplikacji, aby zarejestrować ważne informacje.
- Przechwytywanie logów - Android przechwytuje te logi i przekazuje je do ***Logcat***.
- Wyświetlanie logów - ***Logcat*** wyświetla logi w czasie rzeczywistym, umożliwiając programiście monitorowanie działania aplikacji i identyfikowanie potencjalnych problemów.

Rodzaje logów

- Log.d (..) - DEBUG
- Log.i (..) - INFO
- Log.w (..) - WARN (warning)
- Log.e (..) - ERROR
- Log.v (..) - VERBOSE
- Log.wtf(..) - log asercji

Logi debugowania są kompilowane, ale usuwane w czasie wykonywania.
Dzienniki Logi, ostrzeżeń i informacji są zawsze przechowywane.

Logi systemowe

- Logi debugowania (Log.d()): Służą do rejestrowania informacji pomocnych w debugowaniu aplikacji, takich jak wartości zmiennych, przebieg wykonywania kodu itp.

```
int liczba = 10;
Log.d("MojaAplikacja", "Wartość liczby: "+ liczba);
```

- Logi informacyjne (Log.i()): Służą do rejestrowania ogólnych informacji o działaniu aplikacji, takich jak uruchomienie aktywności, zakończenie zadania itp.

```
Log.i("MojaAplikacja", "Uruchomiono aktywność Ustawienia");
```

- Logi ostrzegawcze (Log.w()): Służą do rejestrowania potencjalnych problemów, które nie są jeszcze błędami, ale mogą do nich prowadzić.

```
if (polaczenieSiecioweNiedostepne) {
   Log.w("MojaAplikacja", "Brak połączenia z Internetem");
}
```

Logi systemowe

- Logi błędów (Log.e()): Służą do rejestrowania błędów, które uniemożliwiają poprawne działanie aplikacji

```
try {
    // Kod, który może wygenerować wyjątek
} catch (Exception e) {
    Log.e("MojaAplikacja", "Wystąpił błąd: ", e);
}
```

- Logi asercji (Log.wtf()): Służą do rejestrowania krytycznych błędów, które nigdy nie powinny wystąpić. Skrót "wtf" oznacza "What a Terrible Failure".

```
if (warunekKrytycznyNieJestSpełniony) {
    Log.wtf("MojaAplikacja", "Krytyczny błąd: warunek nie jest spełniony");
}
```

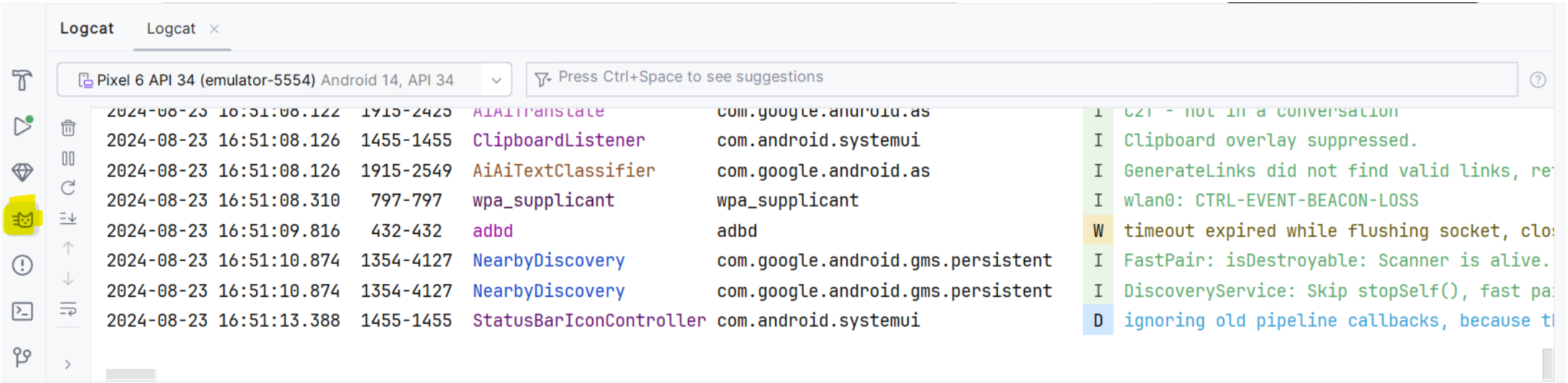
Logi systemowe

- Log.V(..) - służy do rejestrowania logów o poziomie verbose (szczegółowy).
  - Bardzo szczegółowe debugowanie działania aplikacji (wartości wszystkich zmiennych w pętli, śledzenie przebiegu wykonywania kodu krok po kroku itp.)
  - Śledzenie przepływu danych, np. jak dane są przekazywane między różnymi komponentami.
  - Monitorowanie wydajności i identyfikacja potencjalne wąskiego gardła.

```
for (int i = 0; i < 10; i++) {
    Log.v("MojaAplikacja", "Iteracja pętli: " + i);
    // ... inny kod
}
```

UWAGA: Logi verbose generują dużą ilość danych, co może wpłynąć na wydajność aplikacji. Używaj ich tylko wtedy, gdy są absolutnie niezbędne i usuwaj przed publikacją apki.

Logi systemowe



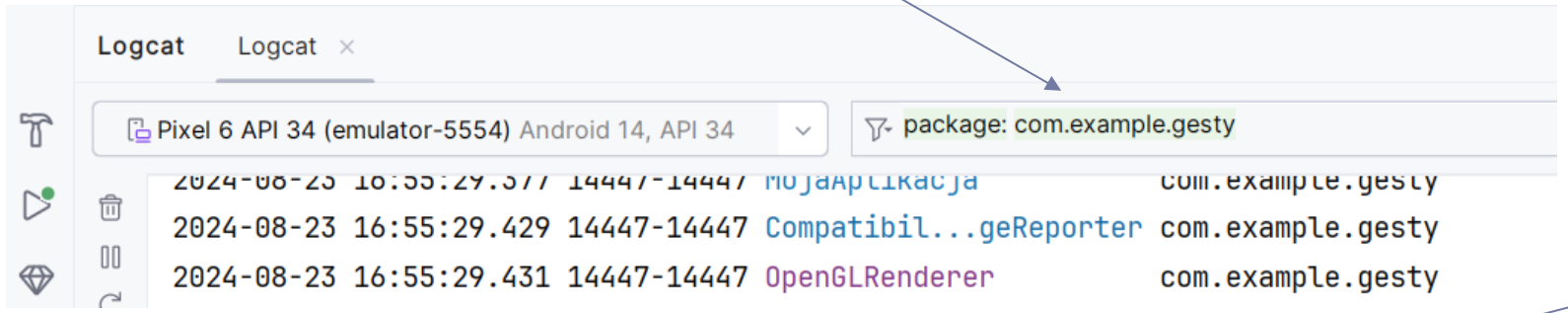
Logcat to narzędzie w Android Studio które służy do wyświetlania logów generowanych przez aplikację oraz system.

Jest to okno, które pokazuje w czasie rzeczywistym informacje diagnostyczne, takie jak komunikaty o błędach, ostrzeżenia, informacje itp.

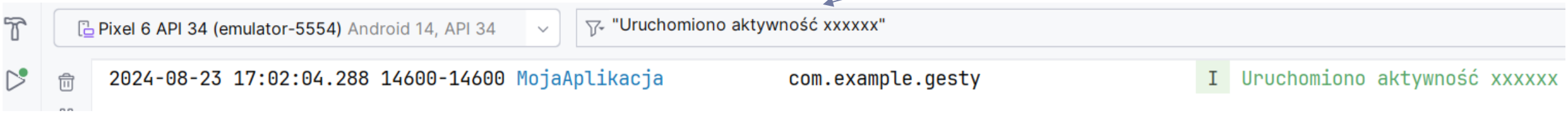
Logi systemowe

Logcat pokazuje wszystko, co dzieje się w systemie. Aby coś znaleźć w tym natłoku danych należy je odfiltrować.

Możemy zawęzić wyświetlanie do logów naszej aplikacji

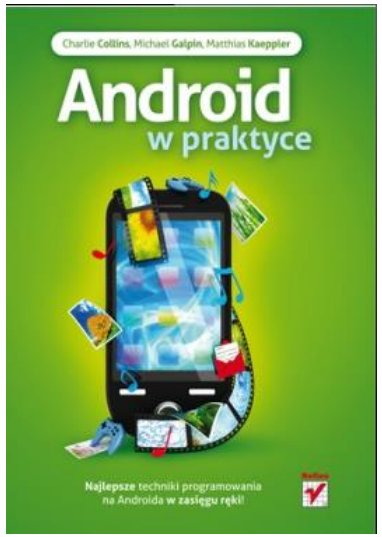


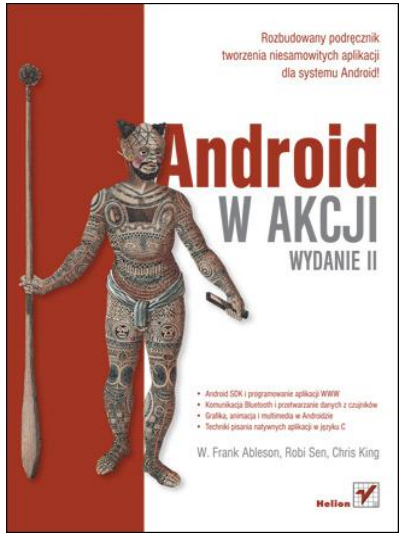
Albo wyszukać log po jego opisie

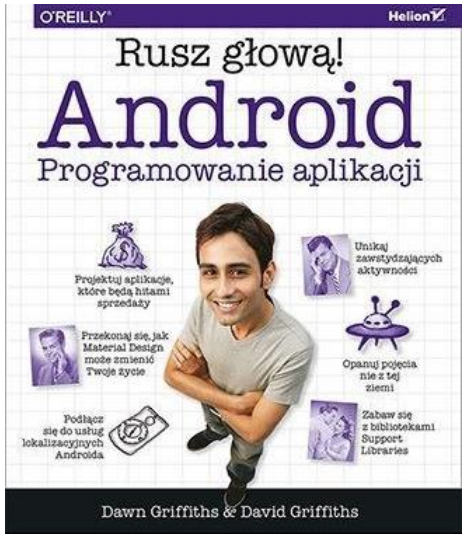


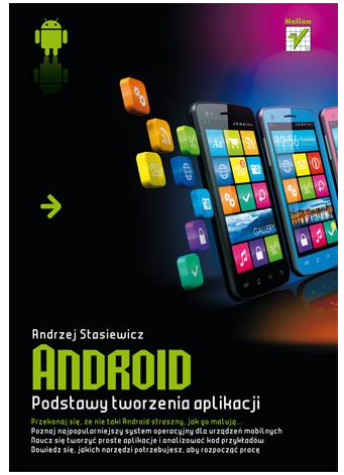
Możliwości wyszukiwania są oczywiście dużo większe…

Literatura


Charlie Collins, Michael Galpin, Matthias Kaeppler
Android
w praktyce
Najlepsze techniki programowania na Androida w zasięgu ręki!


Rozbudowany podręcznik tworzenia niesamowitych aplikacji dla systemu Android!
Android
W AKCJI
WYDANIE II
W. Frank Ableson, Robi Sen, Chris King
Helion


O'REILLY
Helion
Rusz głową!
Android
Programowanie aplikacji
Projektuj aplikacje, które będą hitami sprzedaży
Unikaj zawstydzających aktywności
Przekonaj się, jak Material Design może zmienić Twoje życie
Opanuj pojęcia nie z tej ziemi
Podłącz się do usług lokalizacyjnych Androida
Zabaw się z bibliotekami Support Libraries
Dawn Griffiths & David Griffiths


Helion
Andrzej Stasiewicz
ANDROID
Podstawy tworzenia aplikacji