

Introduction to Android

- Krótkie przedstawienie
- Czym jest Android
- Scenariusz prezentacji
- Q/A session

Krótkie przedstawienie

- Kim jestem?
- Zainteresowania
- Dlaczego Android?

Czym jest Android

- Bazuje na Linuxie
- Języki w jakich możecie pisać (Java, Groovy, Scala, C i pochodne, Java script(phonegap))
- Market share, Gartner - "do 2012 Android stanie się drugim najpopularniejszym systemem dla smartphonów."
- MIC - "w 2013, 31.8 millionów telefonów na Androidzie i 126 million urządzeń na Androidzie (tablety etc.)."

Scenariusz prezentacji

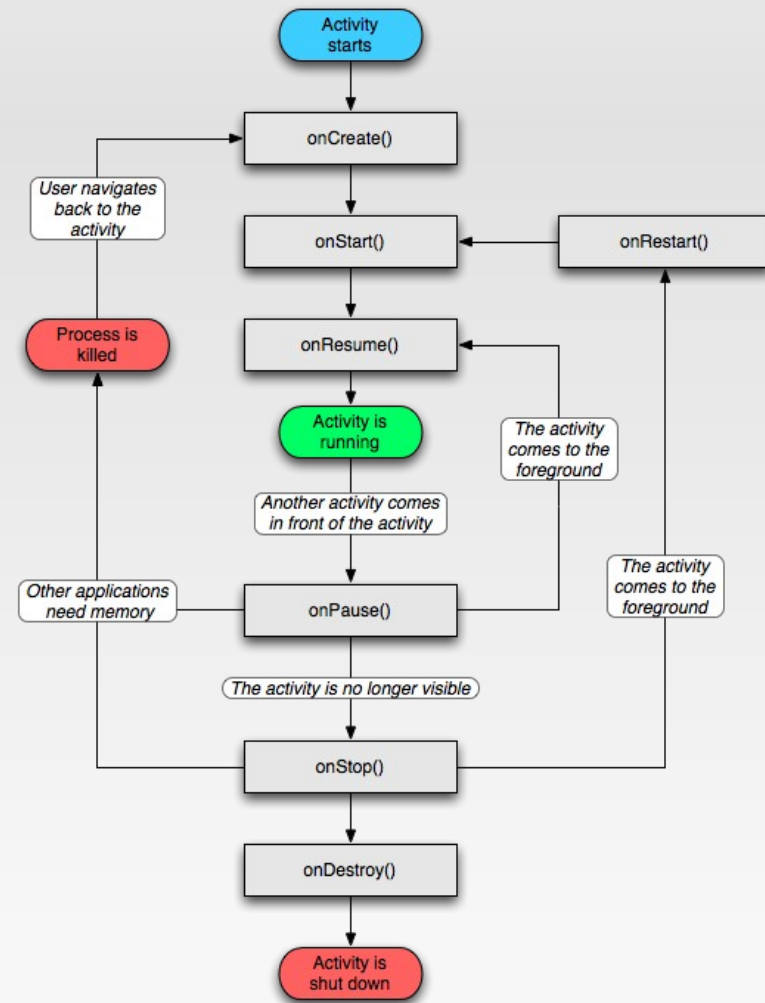
- Komponenty aplikacji Androida
- Cykl życia Activity
- Cykl życia Service (usługi)
- Narzędzia do feedback'owania użytkownika
- Przechowywanie danych
- Budowa View
- Q/A

Komponenty aplikacji

- Activities
- Services
- Broadcast i Intent Receivers
- Content Providers

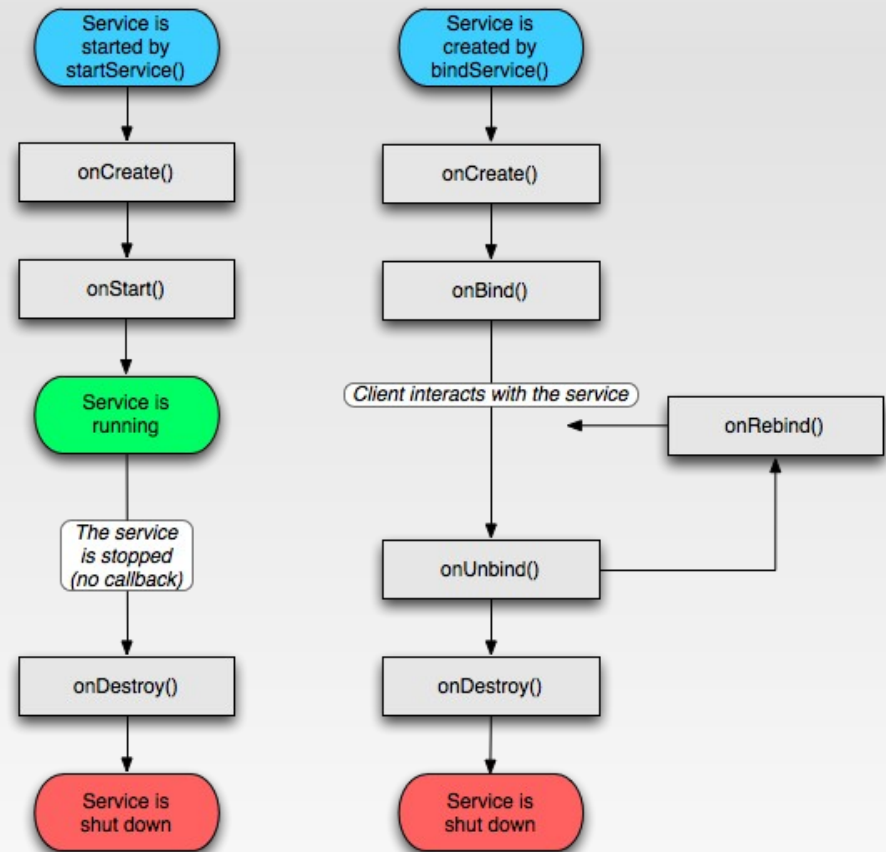
Cykl życia Activity

- onCreate
- onStart
- onResume
- onPause
- onStop
- onDestroy



Cykl życia Service

- Różnice między onCreate i onStart
- onResume, onPause, onStop nie są wymagane
- onBind
- onDestroy



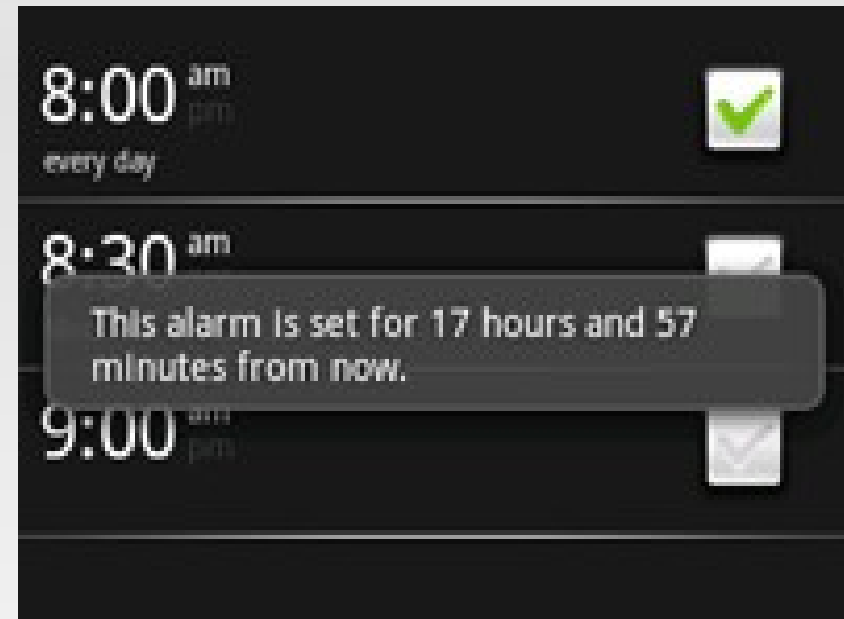
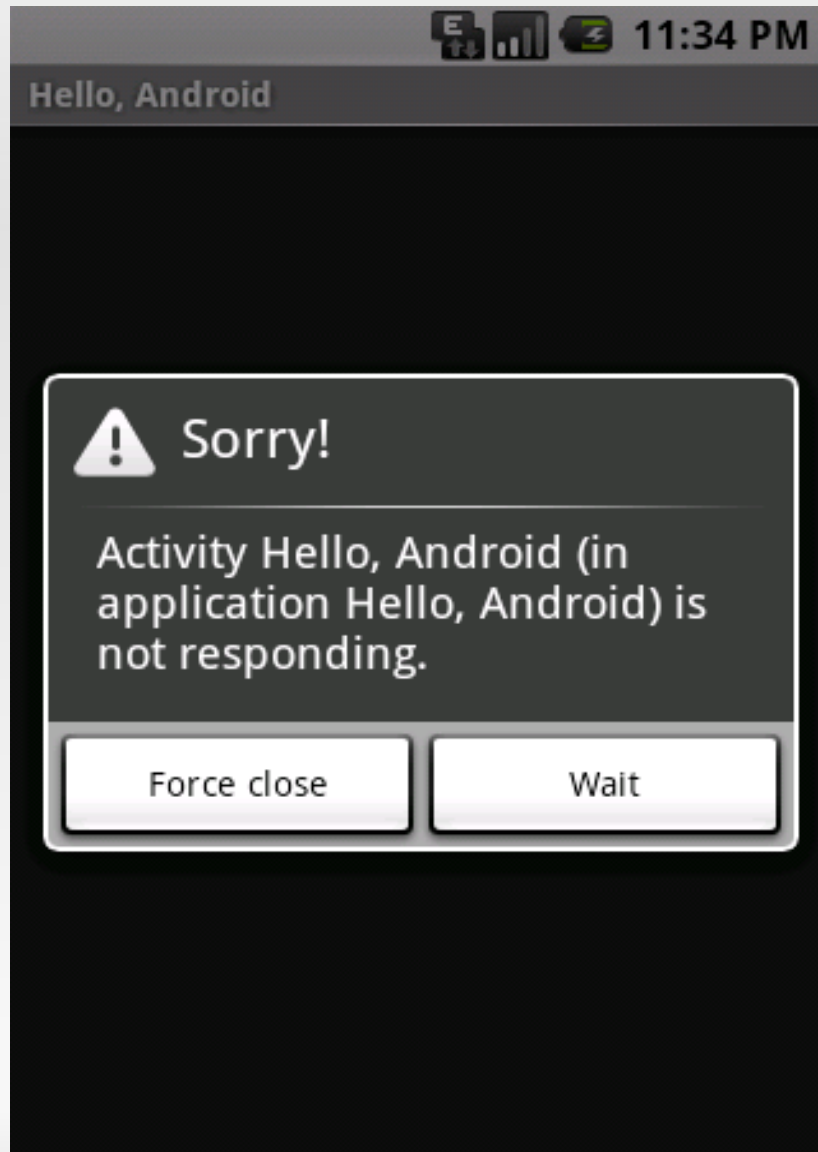
Narzędzia do feedback'owania użytkownika

- Toast messages
- AlertBoxes
- Notification

Toasts i Alert notifications

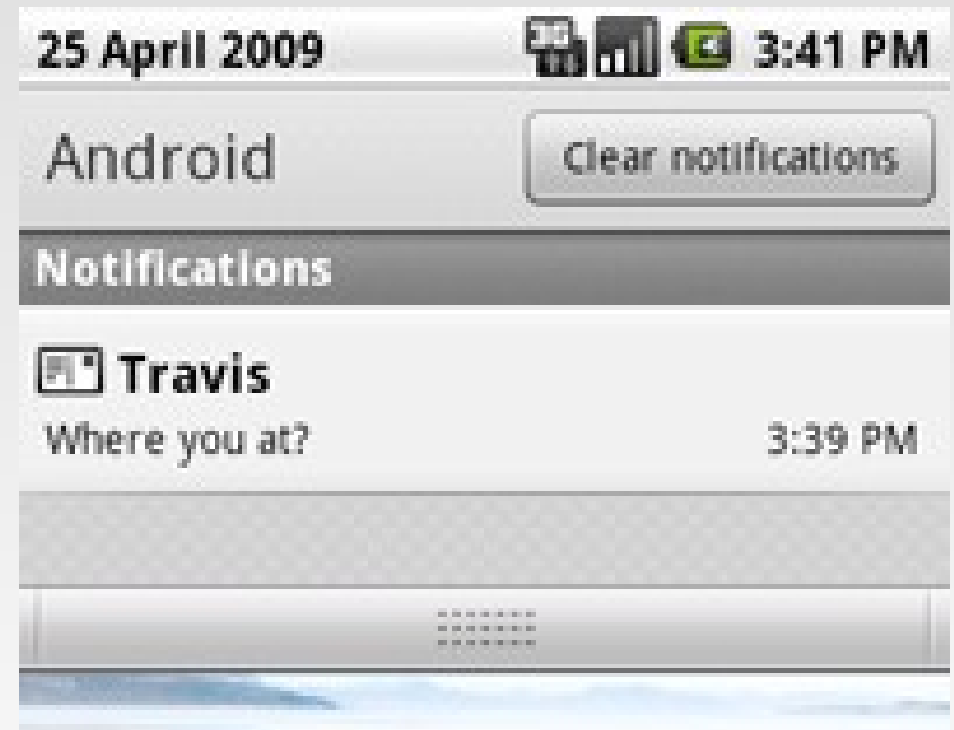
- Toasts:
 - Tylko informowanie użytkownika bez możliwości interakcji
 - Którki czas trwania
 - Nie przerywają przepływu aplikacji (nieinwazyjny UX)
- Alerty
 - Możliwość (konieczność!) ingerencji użytkownika
 - Możliwość dodania obrazków/tytułu etc.
 - Trwają aż użytkownik podejmie akcję
 - Przerywają przepływ aplikacji (inwazyjny UX)

Przykłady Toast'a i Alert Box'a



Notyfikacje

- Z reguły startowane z Services
- Możliwość dodania własnej grafiki, dźwięku, pracy diody
- Dwa typy ongoing i isntant



Przechowywanie danych

- SQLite
- Content Providers
- Różnice w kontekstach użycia
- O czym należy pamiętać gdy korzystamy z SQLite zamiast Content Provider'ów

SQLite

- Self-contained (serwer nie jest wymagany)
- Transactional engine
- Unikalny dla konkretnej aplikacji

Content Provider

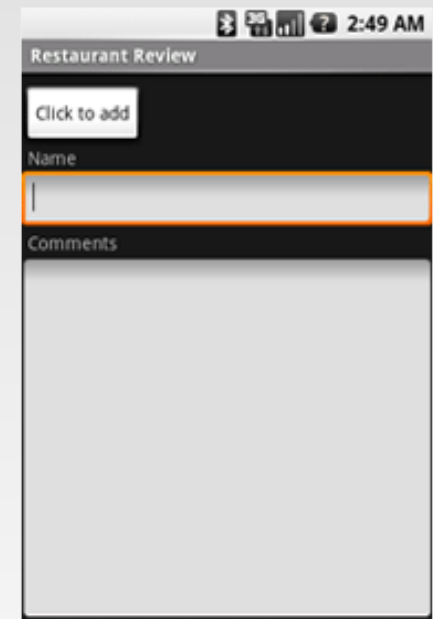
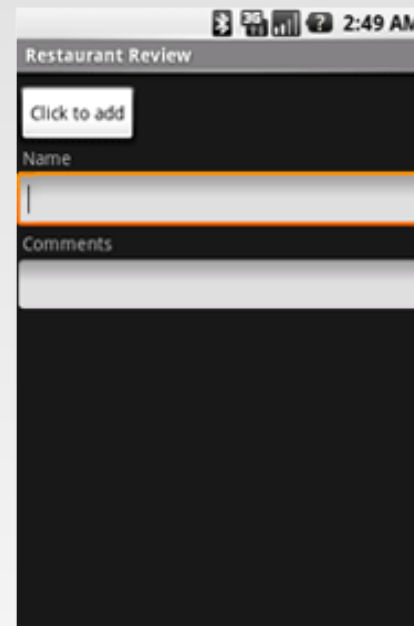
- Implementacja content providera kroki:
 - Rozszerzenie ContentProvider class
 - Zdefiniowanie CONTENT_URI
 - Zdefiniowanie przestrzeni do trzymania danych (przeważnie baza danych)
 - Stworzenie nazw kolumn dla komunikacji z klientem
 - Definicja procesu za pomocą którego dane binarne są zwracane do klienta
 - Deklaracja public static String której klient użyje aby odnaleźć kolumny
 - Implementacja metod CRUD Kursora
 - Update AndroidManifest.xml z deklaracją <provider>
 - Definicja typów MIME dla nowych typów danych

Budowa View

- Nowy system do zarządzania GUI dla Java ecosphere
- Model-View-Controller
- Model – guts aplikacji
- View – feedback aplikacji skierowany do użytkownika
- Controller – część aplikacji która odpowiada na akcje użytkownika

Przykładowy View z hierarchią

- Hierarchia widoków:
 - LinearLayout (horizontal)
 - Button(width:wrap_content)
 - TextView (width:fill_parent)
 - EditText(width:fill_parent)
 - TextView(width:fill_parent)
 - EditText(width:fill_parent)/EditText(height:fill_parent)



Sesja Q/A i podziękowania