Helion

SMASHING MAGAZINE

Juhani Lehtimäki

PODRĘCZNIK DLA PROJEKTANTÓW Android™ UI

SMASHING MAGAZINE

Klucz do sukcesu Twojej aplikacji!

Tytuł oryginału: Smashing Android UI: Responsive Android UI and Design Patterns for Phones and Tablets

Tłumaczenie: Mikołaj Szczepaniak

ISBN: 978-83-246-6859-5
This edition first published 2013

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:
ftp://ftp.helion.pl/przyklady/andrui.zip

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/andrui
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

6 INTENCJE ANDROIDA

SYSTEM INTENCJI ANDROIDA jest prawdopodobnie najważniejszym mechanizmem oferowanym przez tę platformę. Intencje umożliwiają wewnętrzne i zewnętrzne wiązanie aplikacji. System intencji umożliwia programistom wywoływanie zarówno funkcji platformy Android, jak i funkcji wszystkich pozostałych zainstalowanych aplikacji. Ten sam system umożliwia aplikacjom udostępnianie funkcji innym aplikacjom.

W tym rozdziale wyjaśnię, czym są intencje Androida i gdzie są stosowane.

Celem tego rozdziału jest szczegółowe opisanie wpływu mechanizmu intencji na sposób projektowania interfejsu użytkownika aplikacji budowanych dla systemu Android. Rozdział zawiera co prawda kilka przykładów, ale nie obejmuje pełnej specyfikacji intencji ani wszystkich przypadków użycia. Czytelników zainteresowanych szczegółami technicznymi zachęcam do zapoznania się z dokumentacją Androida (patrz strona *http://developer.android.com/reference/android/content/Intent.html*).

INTENCJE UMOŻLIWIAJĄ APLIKACJOM WZAJEMNĄ WSPÓŁPRACĘ

Intencja przypomina trochę technicznie i formalnie zdefiniowany komunikat wysyłany do komponentu aplikacji. Taki komunikat może być wysyłany albo wewnątrz aplikacji, albo przekazywany pomiędzy różnymi aplikacjami bądź nawet pomiędzy systemem operacyjnym a aplikacjami. Aplikacja może na przykład wysłać do systemu operacyjnego komunikat o potrzebie wybrania jakiegoś numeru telefonu.

Największą zaletą mechanizmu intencji jest możliwość współpracy różnych aplikacji i wzajemnego udostępniania funkcji w łatwy i bezproblemowy sposób. Każda aplikacja może zażądać od platformy identyfikacji pozostałych aplikacji udostępniających określone funkcje, po czym użyć jednej z tych aplikacji lub umożliwić użytkownikowi wybór właściwej.

Jednym z najbardziej popularnych przypadków użycia intencji jest udostępnianie przez aplikację jakiegoś zasobu, na przykład obrazu. Niezależnie od tego, czy chodzi o edytor zdjęć, aplikację obsługującą aparat fotograficzny, aplikację graficzną, czy dowolny inny program, istnieje możliwość poinformowania systemu Android o istnieniu obrazu do udostępnienia. System operacyjny „wie", które spośród pozostałych zainstalowanych aplikacji mogą obsłużyć to żądanie.

Przeanalizujmy teraz konkretny przykład. Poniższa sekwencja obrazów ilustruje ciąg czynności wykonywanych przez użytkownika, który za pomocą czterech różnych aplikacji robi zdjęcie i publikuje je w serwisie społecznościowym. Na rysunku 6.1 pokazano pierwszy krok, polegający na zrobieniu zdjęcia za pomocą aplikacji aparatu dostępnej w Androidzie. Następnie użytkownik otwiera to zdjęcie w aplikacji edytora (Skitch), gdzie umieszcza na zdjęciu prosty napis (patrz rysunek 6.2). I wreszcie użytkownik udostępnia gotowe (przerobione) zdjęcie na Twitterze (patrz rysunek 6.3).

Rysunek 6.1. Użytkownik robi zdjęcie za pomocą aplikacji aparatu fotograficznego Androida
Źródło: Android

Rysunek 6.2. Użytkownik umieszcza na zdjęciu tekst w aplikacji Skitch
Źródło: Skitch, copyright 2012 Evernote Corporation

Rysunek 6.3. Użytkownik udostępnia gotowe zdjęcie na Twitterze
Źródło: Twitter

W opisanej sekwencji na szczególną uwagę zasługuje brak konieczności zapisywania zdjęcia w galerii czy systemie plików — użytkownik może osiągnąć swój cel bez zapisywania obrazu. Plik obrazu jest automatycznie przenoszony przez system operacyjny Android bez wiedzy i ingerencji użytkownika. Ostatni krok (udostępnienie zdjęcia na Twitterze) jest wykonywany przez aplikację Twitter, zatem użytkownik nie musi się dodatkowo logować w celu sprawdzenia, czy jego wpis ze zdjęciem rzeczywiście jest gotowy do publikacji.

KORZYSTANIE Z SERWISÓW SPOŁECZNOŚCIOWYCH I UDOSTĘPNIANIE ZASOBÓW

Aplikacje mobilne i serwisy społecznościowe wprost doskonale do siebie pasują. Ludzie kochają swoje telefony i uwielbiają się dzielić w wybranych serwisach społecznościowych dosłownie wszystkim, co widzą, słyszą i jedzą. Na niektórych platformach integracja z Facebookiem i Twitterem jest tylko jednym ze sloganów i narzędzi wykorzystywanych przez marketingowców. Urządzenia z systemem Android oferują możliwość udostępniania

zasobów ze wszystkich aplikacji we wszystkich serwisach społecznościowych, w tym na Google+, LinkedIn, Orkut i oczywiście na Facebooku i Twitterze. Ewentualne nowe serwisy społecznościowe muszą tylko przygotować wersje swoich aplikacji dla platformy Android, stosując odpowiednie filtry intencji (to zagadnienie zostanie wyjaśnione w dalszej części tego rozdziału). Po zainstalowaniu aplikacji użytkownicy mogą udostępniać w nowej sieci społecznościowej zasoby bezpośrednio z galerii, wszystkich odpowiednio zaimplementowanych aplikacji do robienia zdjęć, aplikacji pocztówkowych, edytorów graficznych, edytorów tekstu. Z tego samego powodu użytkownicy nie są ściśle przywiązani do oficjalnych aplikacji klienckich. Na rysunku 6.4 pokazano proces udostępniania obrazu przez użytkownika. Warto zwrócić uwagę na aplikacje Seesmic, Plume, Tweet Lanes, TweetDeck i Twitter — wszystkie są klientami Twittera, a użytkownik może swobodnie wybrać tę, która posłuży do udostępnienia zdjęcia w tym serwisie.

Rysunek 6.4. Użytkownik wybrał opcję udostępnienia pliku z aplikacji galerii systemu Android. System operacyjny Android prosi użytkownika o wskazanie aplikacji, która ma być użyta do zakończenia tej operacji
Źródło: Android

Podczas budowy aplikacji dla systemu Android nie musimy tracić czasu na integrację z serwisami społecznościowymi — wszystkim zajmie się platforma. Nie musimy wybierać aplikacji, które naszym zdaniem powinny być obsługiwane; nie musimy też implementować żadnych funkcji odpowiedzialnych za udostępnianie zasobów w konkretnych serwisach. Nasze zadania ograniczają się do implementacji intencji udostępniania zasobów zgodnie ze specyfikacją. Warto pamiętać, że na przykład lista aplikacji widoczna na rysunku 6.4 jest generowana automatycznie przez system operacyjny, zatem nawet tego elementu nie

musimy samodzielnie implementować. Wszystkim zajmuje się system operacyjny. Warto też podkreślić, że lista widoczna na rysunku 6.4 obejmuje tylko te aplikacje, które oferują możliwość udostępniania obrazów. Użytkownik nigdy nie otrzymuje do wyboru aplikacji, które nie będą „wiedziały", co zrobić z wybranym typem danych.

Przeanalizujmy teraz rysunek 6.5. Pokazano na nim efekt wywołania podobnej intencji udostępniania, jednak tym razem przedmiotem publikacji jest nie obraz, tylko adres URL skopiowany z przeglądarki. System interpretacji intencji Androida automatycznie określa, które aplikacje należy udostępnić na wyświetlanej liście. Do mechanizmu interpretacji intencji wrócę w dalszej części tego rozdziału.

Rysunek 6.5. Użytkownik wybrał opcję udostępnienia adresu URL. System operacyjny Android nie wyświetla już aplikacji obsługujących obrazy, tylko aplikacje, które potrafią przetworzyć adres URL
Źródło: Android

PRACA Z PRZEGLĄDARKAMI

Przeglądarki należą do najważniejszych elementów smartfonów i tabletów. Są bodaj najczęściej używanymi aplikacjami na wszystkich urządzeniach z systemem Android. Okazuje się, że system intencji umożliwia wiązanie naszych aplikacji także z przeglądarkami. Przeglądarki dla Androida (przynajmniej te prawidłowo zaimplementowane) używają intencji do otwierania każdego linka dotkniętego przez użytkownika. Takie intencje

zwykle są używane przez samą przeglądarkę, jednak zdarza się, że zasoby wskazywane przez adres URL mogą być lepiej prezentowane przez inną aplikację. Aplikacja może zasygnalizować systemowi operacyjnemu możliwość obsługi adresów URL pasujących do pewnych wzorców, na przykład z określoną nazwą domeny. W momencie dotknięcia przez użytkownika linka pasującego do tego wzorca system operacyjny wyświetli listę z możliwością wyboru właściwej aplikacji. Na rysunku 6.6 pokazano przykład listy wyświetlonej po dotknięciu przez użytkownika zwykłego linka HTML wskazującego sklep Google Play. System operacyjny Android rozpoznaje ten link jako specjalny przypadek, który może być dodatkowo obsłużony przez aplikację Google Play — w związku z tym system umożliwia użytkownikowi wybór aplikacji, która wyda mu się najwłaściwsza dla tego linka. Oprócz dwóch przeglądarek zainstalowanych przez samego użytkownika lista zawiera także aplikację Google Play jako jedną z możliwych opcji.

Rysunek 6.6. Użytkownik kliknął w przeglądarce Androida link wskazujący sklep Google Play. System operacyjny wykrył, że istnieje inna aplikacja, która może obsłużyć ten adres URL, i zaproponował użytkownikowi wybór kilku aplikacji, które mogą przetworzyć to żądanie
Źródło: Android

O sile opisanego mechanizmu decyduje między innymi brak wymagań dotyczących specjalnych konstrukcji składniowych po stronie serwisu internetowego. Strona internetowa zawiera standardowy link, który przeniósłby użytkownika na witrynę sklepu z aplikacjami Androida (gdyby ten użytkownik korzystał ze standardowej przeglądarki internetowej lub nie dysponował na swoim urządzeniu zainstalowaną aplikacją Google Play).

Wskazówka. Jeśli nasza aplikacja oferuje alternatywny sposób przeglądania treści dostępnej w internecie, koniecznie powinniśmy zadbać o subskrypcję adresów URL pasujących do odpowiedniego wzorca (na przykład domeny). Nie ma powodu, by nie skorzystać z tej możliwości. Tak zbudowana aplikacja będzie traktowana jako lepsza alternatywa dla przeglądarki podczas interakcji z wybranym rodzajem treści. Jeśli aplikacja nie jest taką alternatywą, powinniśmy albo przemyśleć swoją strategię, albo poprawić samą aplikację.

Warto pamiętać, że intencje związane z adresami URL obejmują pełne adresy, w tym wszystkie parametry. Oznacza to, że nasza aplikacja może bezpośrednio otwierać właściwą treść. W powyższym przykładzie wybór aplikacji Google Play spowoduje skierowanie użytkownika bezpośrednio do strony odpowiedniej aplikacji w sklepie Google Play. Podobnie, link do serwisu YouTube otwarty w aplikacji YouTube skieruje użytkownika bezpośrednio na stronę umożliwiającą odtworzenie odpowiedniego zapisu wideo.

JAK DZIAŁAJĄ INTENCJE ANDROIDA?

Czas zajrzeć do wnętrza systemu i sprawdzić, jak naprawdę działają intencje platformy Android. Ten podrozdział polecam także czytelnikom, których nie interesują szczegóły techniczne, ponieważ dobre rozumienie intencji może bardzo ułatwić ocenę, co jest, a co nie jest możliwe do osiągnięcia za pomocą tego mechanizmu. Warto też opanować kilka terminów związanych z intencjami. Przykłady prezentowane w tym rozdziale są bardzo proste, jednak czytelnicy niebędący programistami mogą je pominąć.

RODZAJE INTENCJI

Istnieją dwa rodzaje intencji: intencje aktywności (ang. *activity intents*) i intencje rozgłaszania (ang. *broadcast intents*).

- **Intencje aktywności.** Intencje aktywności zawsze cechują się jedną aplikacją nadawcy i jedną aplikacją obsługującą. Aplikacja obsługująca może mieć postać aktywności lub usługi. Intencje aktywności podzielono na dwie dodatkowe kategorie: intencje jawne i intencje niejawne.
  - **Intencje jawne.** Jeśli aplikacja została przygotowana z myślą o obsłudze danej intencji przez konkretną aktywność lub klasę usługi, warto wywołać intencję jawną. Taka intencja będzie obsługiwana bezpośrednio przez odpowiednią aktywność lub usługę. W ten sposób aplikacje zwykle obsługują komunikację wewnętrzną. Mimo że intencje jawne są bardzo ważnymi konstrukcjami, nie są zbyt interesujące w kontekście interfejsów użytkownika.
  - **Intencje niejawne.** Intencje niejawne są używane w sytuacji, gdy aplikacja wywołująca nie dysponuje z góry informacją, która aplikacja obsłuży to żądanie. Aplikacja wywołująca tworzy intencję opisującą rodzaj akcji, która ma być wykonana, po czym dołącza do tej intencji niezbędne dane i wysyła ją do systemu operacyjnego.

    Mechanizm intencji niejawnych umożliwia tworzenie luźnych relacji pomiędzy aplikacjami wywołującymi a aplikacjami odpowiadającymi na te wywołania.

Interfejs pomiędzy tymi aplikacjami jest precyzyjnie zdefiniowany, ale żadna z nich nie dysponuje wiedzą o aplikacji po drugiej stronie. Luźne związki pomiędzy komponentami znacznie ułatwiają konserwację aplikacji, ponieważ zmiany w pozostałych komponentach lub aplikacjach nie utrudniają pracy innych aplikacji i komponentów (o ile nie naruszają wcześniejszego interfejsu). Niezależność od pozostałych aplikacji oznacza też, że aplikacje, o których być może nawet nie słyszeliśmy podczas prac nad własną aplikacją, mogą w przyszłości udostępniać jej przydatne funkcje.

Intencje niejawne są bardzo ciekawe i istotne z perspektywy projektanta interfejsu użytkownika. Zrozumienie ich działania jest absolutnie konieczne do budowania dobrych aplikacji dla Androida.

- **Intencje rozgłaszania.** Intencje rozgłaszania są (jak nietrudno się domyślić) wysyłane przez jedną aplikację, ale mogą być odbierane i obsługiwane przez wielu adresatów. Intencje aktywności zawsze są wysyłane przez jedną aplikację i obsługiwane przez jedną aplikację docelową. W niektórych przypadkach taka komunikacja jeden-jeden nie wystarczy. Niektóre zdarzenia, na przykład te dotyczące niskiego poziomu naładowania baterii, mogą zainteresować więcej aplikacji. W takich przypadkach konieczne jest zastosowanie techniki rozgłaszania. Mechanizm rozgłaszania wykorzystuje te same rozwiązania co intencje jawne, jednak tak wysyłane intencje nie są obsługiwane przez aktywności ani usługi, tylko przez odbiorców rozgłaszania.

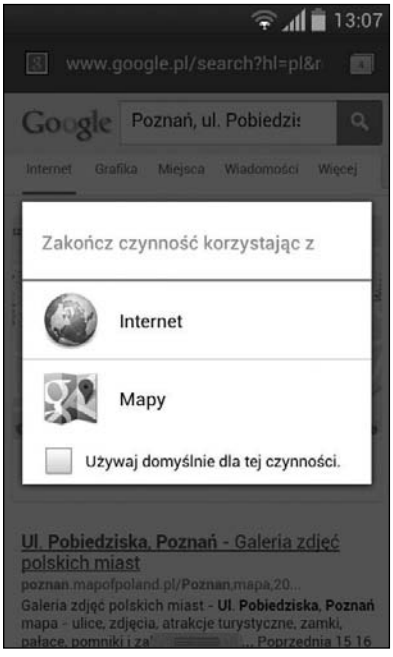
TECHNICZNY PRZYKŁAD WYSYŁANIA INTENCJI

W tym punkcie przeanalizujemy przykład kodu umożliwiającego użytkownikom aplikacji wykonywanie dodatkowych operacji na adresach pocztowych. Podobne rozwiązania są dość powszechne w takich domyślnych aplikacjach Androida jak Kalendarz czy Mapy Google. Dane adresowe dobrze ilustrują potencjał systemu intencji. Przypuśćmy, że nasza aplikacja dysponuje informacjami o adresie pocztowym. Warto w takim przypadku umożliwić użytkownikom przejście do widoku mapy lub nawet wybór opcji nawigowania do danego adresu. Warto pamiętać, że nie musimy wiedzieć, co użytkownicy zrobią z informacjami adresowymi. To do nich należy wybór aplikacji, której będą chcieli użyć.

Jedną z najważniejszych zalet stosowania intencji jest brak konieczności pisania jakiegokolwiek kodu map czy nawigacji na poziomie naszej aplikacji — możemy po prostu przekazać dane adresowe do przetworzenia przez inne aplikacje. Wysyłanie intencji jest bardzo proste. Warto przeanalizować poniższy przykład kodu. Kod w tej formie mógłby występować dosłownie wszędzie, jednak na potrzeby tego przykładu zostanie umieszczony w klasie aktywności. Intencja jest wywoływana w momencie dotknięcia przez użytkownika przycisku interfejsu.

```
sendIntent.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    Uri geoUri = Uri.parse("geo:0,0?q="+ addressField.getText().toString());
    Intent mapCall = new Intent(Intent.ACTION_VIEW, geoUri);
    startActivity(mapCall);
  }
});
```

Po wysłaniu intencji na ekranie zostanie wyświetlone doskonale znane okno dialogowe wyboru aplikacji (patrz rysunek 6.7). Jeśli na przykład użytkownik wybierze Mapy Google, aplikacja otworzy widok mapy, od razu wskazując prawidłowy adres (patrz rysunek 6.8).



Rysunek 6.7. Okno dialogowe wyboru aplikacji po wysłaniu intencji przez aplikację

Źródło: Android

Rysunek 6.8. Mapy Google otwierają wysłany adres

Źródło: Google Inc.

FILTRY INTENCJI: AKCJE, DANE I KATEGORIE

Skąd system operacyjny „wie", która aktywność, usługa lub który odbiorca rozgłaszania ma otrzymać daną intencję? Skąd pewność, że wysyłana przez naszą aplikację intencja będzie obsługiwana tylko przez aktywności wykonujące operacje, na których nam zależy? Przekazujemy sterowanie poza własną aplikację do jakiejś innej aplikacji. W tej kwestii musimy się zdać na system operacyjny, który zadba o kierowanie użytkowników do właściwych aplikacji.

Warto w tym kontekście zajrzeć do wnętrza systemu i zrozumieć, jak działa mechanizm Androida odpowiedzialny za interpretację intencji. W systemie Android zastosowano dwa główne komponenty. Dla każdej aktywności, usługi lub każdego odbiorcy rozgłaszania można zdefiniować zbiór powiązanych filtrów intencji (zbiór intencji możemy zdefiniować albo w pliku manifestu aplikacji, albo dynamicznie, w czasie wykonywania kodu). Intencja zawiera definicję akcji i pole danych, a często obejmuje także kategorie i pewne dane dodatkowe. Po otrzymaniu intencji system operacyjny porównuje akcję, dane i kategorie z filtrami intencji wszystkich aplikacji i wybiera tylko pasujące aplikacje.

Akcje i kategorie to po prostu nazwy. Nie ma w tym mechanizmie niczego skomplikowanego. Nieco trudniejsza jest obsługa właściwych danych i ew. danych dodatkowych. Dane są definiowane albo w formie identyfikatora URI, albo jako typ MIME. Identyfikator URI składa się z dwóch części oddzielonych dwukropkiem. Pierwsza część definiuje typ danych lub schemat. Druga część identyfikuje same dane. Na przykład identyfikator URI *tel:123456789* oznacza, że typem danych jest *tel*, natomiast dane mają postać *123456789*. Z perspektywy mechanizmu interpretacji intencji zasadnicze znaczenie ma typ danych.

Interfejsy API systemu Android definiują wiele standardowych akcji, kategorii i kluczy dodatkowych danych. Te standardowe definicje intencji są stosowane niemal we wszystkich elementach platformy Android. Niektóre są wywoływane przez sam system operacyjny, inne są używane przez aplikacje domyślne dostarczane wraz z tym systemem. Standardowe akcje obejmują wysyłanie (udostępnianie), wybieranie numeru, dzwonienie, wyświetlanie i wiele innych zadań.

W tabeli 6.1 opisano standardowe akcje aktywności, natomiast w tabeli 6.2 wymieniono standardowe akcje rozgłaszania. Kompletną listę akcji używanych w pakiecie Android SDK można znaleźć w dokumentacji intencji na stronie *http://developer.android.com/reference/android/content/Intent.html*.

Tabela 6.1. Standardowe akcje aktywności Androida

Nazwa akcji	Opis akcji
`ACTION_ANSWER`	Żąda obsłużenia przychodzącego połączenia telefonicznego.
`ACTION_ATTACH_DATA`	Określa, że pewna część danych powinna zostać dołączona w jakimś innym miejscu.
`ACTION_CALL`	Żąda połączenia telefonicznego z osobą wskazaną w danych.
`ACTION_CHOOSER`	Żąda wyświetlenia okna wyboru aktywności, w którym użytkownik będzie mógł wskazać dalsze działania.
`ACTION_DELETE`	Żąda usunięcia przekazanych danych z kontenera.
`ACTION_DIAL`	Żąda wybrania numeru telefonicznego wskazanego w danych.
`ACTION_EDIT`	Żąda bezpośredniego dostępu do edycji przekazanych danych.
`ACTION_FACTORY_TEST`	Główny punkt wejściowy na potrzeby testów fabrycznych.
`ACTION_GET_CONTENT`	Żąda wyświetlenia okna, w którym użytkownik będzie mógł wybrać konkretny rodzaj danych, oraz zwrócenia wybranego typu.
`ACTION_INSERT`	Żąda umieszczenia pustego elementu w danym kontenerze.
`ACTION_MAIN`	Żąda uruchomienia jako głównego punktu wejściowego (nie oczekuje żadnych danych).
`ACTION_PICK`	Żąda wybrania elementu z przekazanych danych i zwrócenia wybranego elementu.
`ACTION_PICK_ACTIVITY`	Żąda wybrania aktywności dla danej intencji i zwrócenia odpowiedniej klasy.

Nazwa akcji	Opis akcji
ACTION_RUN	Żąda uruchomienia przekazanych danych (cokolwiek to znaczy).
ACTION_SEARCH	Żąda wykonania operacji wyszukiwania.
ACTION_SEND	Żąda dostarczenia pewnych danych do kogoś innego.
ACTION_SENDTO	Żąda wysłania wiadomości do adresata wskazanego w danych.
ACTION_SYNC	Żąda synchronizacji danych.
ACTION_VIEW	Żąda wyświetlenia danych na ekranie urządzenia.

Tabela 6.2. Standardowe akcje rozgłaszania Androida

Nazwa akcji	Opis akcji
ACTION_BATTERY_CHANGED	Ta akcja jest regularnie rozgłaszana w celu przekazania stanu ładowania, poziomu naładowania i innych informacji na temat baterii urządzenia.
ACTION_BOOT_COMPLETED	Ta akcja jest rozgłaszana tylko raz, bezpośrednio po uruchomieniu systemu.
ACTION_PACKAGE_ADDED	Akcja informuje o zainstalowaniu na urządzeniu nowego pakietu aplikacji.
ACTION_PACKAGE_DATA_CLEARED	Akcja informuje o usunięciu przez użytkownika danych jakiegoś pakietu.
ACTION_PACKAGE_REMOVED	Akcja informuje o usunięciu z urządzenia jakiegoś istniejącego pakietu aplikacji.
ACTION_PACKAGE_RESTARTED	Akcja informuje o ponownym uruchomieniu (przez użytkownika) jakiegoś pakietu i o zabiciu wszystkich procesów tego pakietu.
ACTION_POWER_CONNECTED	Akcja informuje o podłączeniu urządzenia do zewnętrznego źródła zasilania.
ACTION_POWER_DISCONNECTED	Akcja informuje o odłączeniu urządzenia od zewnętrznego źródła zasilania.
ACTION_SHUTDOWN	Akcja informuje o wyłączaniu urządzenia.
ACTION_TIME_CHANGED	Akcja informuje o zmianie strefy czasowej.
ACTION_TIMEZONE_CHANGED	Akcja informuje o ustawieniu godziny.
ACTION_TIME_TICK	Akcja informuje o zmianie bieżącej godziny.
ACTION_UID_REMOVED	Akcja informuje o usunięciu z systemu identyfikatora użytkownika.

Oprócz typu danych i akcji system operacyjny uwzględnia kategorię i filtr intencji. W większości przypadków jedyną pasującą kategorią jest kategoria **domyślna**. Za każdym razem, gdy nasz kod wysyła intencję, system operacyjny automatycznie dodaje do niej kategorię domyślną. Właśnie dlatego zawsze należy dodawać tę kategorię do definiowanych filtrów intencji.

Kategorie intencji są istotne tylko wtedy, gdy chcemy zastąpić aktywności ekranu domowego lub stacji dokującej. Wyjątkiem od tej zasady jest kategoria programu startowego. Wszystkie aktywności z filtrem intencji obejmującym tę kategorię będą wyświetlane w oknie programu startowego aplikacji.

Intencje mogą oczywiście wysyłać więcej danych (nie tylko identyfikatory URI). Każda intencja może obejmować dodatkowe pola danych, które nie podlegają formalnej specyfikacji i nie są uwzględniane przez mechanizm interpretacji intencji. Dodatkowe pola są powiązane z różnymi akcjami. Aktywności obsługujące określone typy akcji oczekują dodatkowych danych z konkretnymi kluczami. Typowymi przykładami takich dodatkowych kluczy są wiadomości poczty elektronicznej, tytuły, teksty, tematy i dane strumieniowe (na przykład podczas udostępniania obrazów). Kompletną listę takich standardowych dodatków można znaleźć w dokumentacji platformy Android na stronie *http://developer.android.com/reference/android/content/Intent.html*.

TECHNICZNY PRZYKŁAD OTRZYMYWANIA INTENCJI

Z technicznego punktu widzenia odbieranie intencji nie jest bardziej złożone niż ich wysyłanie. W tym punkcie wykorzystamy ten sam przykład, tyle że zaczerpnięty ze strony odbierającej intencje. Wyobraźmy sobie, że nasza aplikacja może zaoferować użytkownikom pewną ciekawą usługę w momencie, w którym chcą wyświetlić adres pocztowy. Usługa może na przykład wyświetlić instrukcje nawigacyjne dla rowerzystów lub pasażerów transportu publicznego bądź tekstowy opis miejsca, w którym znajduje się dany adres.

W procesie odbierania intencji potrzebujemy dwóch komponentów. Po pierwsze, musimy dodać aktywność do pliku manifestu. W elemencie `activity` musimy zdefiniować filtr intencji, aby system Android mógł określić, które rodzaje intencji mogą być obsługiwane przez daną aktywność. W poniższym przykładzie kodu widać możliwy sposób definiowania filtra intencji z myślą o obsłudze adresów URI ze schematem *geo*. Schemat *geo* identyfikatora URI jest formalną specyfikacją opisywania geolokalizacji.

```
<activity
  android:name=".intents.ReceiveIntentExampleActivity"
  android:label="Smashing Android UI" >
  <intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="geo" />
  </intent-filter>
</activity>
```

Zachęcam do zeskanowania tego kodu QR za pomocą telefonu z systemem Android — w ten sposób można otworzyć aplikację i sprawdzić działanie tego przykładu. Warunkiem działania tego rozwiązania jest oczywiście wcześniejsza instalacja aplikacji dołączonej do tej książki. Informacje na ten temat można znaleźć we „Wprowadzeniu”.

W kodzie aktywności możemy odczytać geolokalizację dołączoną do intencji. W poniższym przykładzie identyfikator URI geolokalizacji jest odczytywany z danych intencji i wyświetlany w niezmienionej formie. W prawdziwej aplikacji należałoby przetworzyć identyfikator URI, aby na jego podstawie zdecydować o dalszych działaniach.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setTitle("Receive Intent Example");
  setContentView(R.layout.receive_intent_example);
  TextView addressText = (TextView) findViewById(R.id.example_address_field);
  addressText.setText("" + getIntent().getData());
}
```

TWORZENIE WŁASNYCH AKCJI

Nie musimy się ograniczać do standardowych, predefiniowanych akcji. Nic nie stoi na przeszkodzie, abyśmy tworzyli własne. Być może nasza aplikacja oferuje usługę, która będzie przydatna dla pozostałych programistów; a może sami budujemy wiele aplikacji, ale chcemy uniknąć ścisłych związków pomiędzy nimi.

Nasze niestandardowe akcje to tak naprawdę nazwy dla zdefiniowanych akcji. Zaleca się poprzedzanie akcji nazwą pakietu, aby uniknąć ich mylenia z akcjami definiowanymi przez innych producentów oprogramowania i programistów.

Poniższy kod definiuje filtr intencji dla zdefiniowanej przeze mnie niestandardowej akcji. Gdybym teraz opublikował nazwę tej akcji (`com.androiduipatterns.smashingandroidui.examples.EXAMPLE_ACTION`) na stronie internetowej mojej aplikacji, pozostali programiści mogliby jej użyć do zintegrowania swoich aplikacji z tą konkretną aktywnością.

```
<activity
  android:name=".intents.ReceiveCustomIntentExampleActivity"
  android:label="Smashing Android UI" >
  <intent-filter>
    <action android:name="com.androiduipatterns
        .smashingandroidui.examples.EXAMPLE_ACTION" />
    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

INTENCJE SĄ WSZĘDZIE

W systemie Android niemal wszystko jest wywoływane przy użyciu intencji. Za każdym razem, gdy uruchamiamy aplikację z poziomu programu startowego, w rzeczywistości używamy intencji wywołanej przez ten program startowy lub ekran domowy. Co więcej, nawet ekran domowy uruchamiamy, wywołując intencję. Jeśli więc chcemy zastąpić domyślny ekran domowy innym ekranem, musimy doprowadzić do sytuacji, w której inna aplikacja będzie obsługiwała intencję ekranu domowego. Ekran domowy Androida i zasobnik aplikacji to także aplikacja, tyle że z odpowiednim filtrem intencji.

Nawet domyślna aplikacja telefonu jest uruchamiana za pośrednictwem intencji. Także wybór telefonu i nawiązanie połączenia odbywają się przy użyciu intencji. Obsługę tych intencji możemy zaimplementować też we własnych aplikacjach. Jeśli postanowimy napisać aplikację w miejsce domyślnego programu do dzwonienia, będziemy mogli to zrobić.

W systemie Android nie istnieje **pojęcie** przeglądarki Androida. Mówiąc precyzyjnie, istnieje aplikacja nazwana Android Browser, jednak jej istnienie nie wyklucza możliwości stosowania wielu innych przeglądarek na tej platformie. Oznacza to, że nie możemy z góry zakładać, że użytkownik naszej aplikacji używa tej czy innej przeglądarki. Każda aplikacja obsługująca adresy URL może zyskać status domyślnej przeglądarki użytkownika — wystarczy, że implementuje aktywność z odpowiednim filtrem intencji.

Ta elastyczność platformy Android stwarza mnóstwo możliwości także dla naszych aplikacji. Warto przy tym pamiętać o wyzwaniach — nie możemy zakładać, że użytkownik korzysta na przykład z domyślnej książki adresowej czy przeglądarki bądź domyślnego ekranu domowego lub programu do dzwonienia.

INTENCJE KONTRA INTERFEJSY API APLIKACJI

Intencje nie są jedynym sposobem udostępniania funkcji przez nasze aplikacje. Wiele serwisów społecznościowych i innych usług udostępnia interfejsy API, które umożliwiają integrowanie ich funkcji bezpośrednio z budowanymi aplikacjami. Mimo że w pewnych przypadkach takie rozwiązanie jest uzasadnione, warto dokładnie przeanalizować wady i zalety tego modelu. W wielu sytuacjach będziemy musieli sami zaimplementować mnóstwo funkcji, które w normalnych okolicznościach moglibyśmy czerpać z innych aplikacji za pomocą intencji. Co więcej, każda rozbudowa aplikacji i serwisu społecznościowego o nową funkcję będzie dezaktualizowała naszą aplikację do czasu uzupełnienia implementacji o obsługę tej funkcji. W przypadku zastosowania intencji nowe funkcje otrzymamy automatycznie przy okazji aktualizacji aplikacji klienckiej.

Inną poważną wadą ścisłej integracji jest konieczność implementacji w kodzie aplikacji mechanizmów uwierzytelniania użytkownika (jeśli wymaga tego stosowany interfejs API). W takim przypadku musimy uzyskać od użytkownika dane uwierzytelniające lub otworzyć wbudowany widok odpowiedzialny za uwierzytelnianie. Niezależnie od wybranego modelu użytkownik będzie musiał zaufać naszej aplikacji — albo wpisać swoje dane uwierzytelniające w oknie naszej aplikacji, albo autoryzować wykonywanie operacji przez naszą aplikację w jego imieniu. Kto z nas byłby skłonny udostępnić swoje dane dostępu do

Facebooka przypadkowej aplikacji? Czy dalibyśmy nieznanej aplikacji prawo publikowania wpisów na naszej tablicy na Facebooku?

PODSUMOWANIE

Mam nadzieję, że udało mi się dowieść potencjału i elastyczności mechanizmu intencji. Podczas projektowania aplikacji warto poświęcić chwilę na analizę możliwości sprawienia, że nasz produkt będzie integralną częścią urządzenia użytkownika. Czy nasza aplikacja dysponuje funkcjami, które mogłaby udostępniać pozostałym aplikacjom w formie intencji? Czy może wykorzystać istniejące funkcje udostępniane przez inne aplikacje, tak abyśmy nie musieli sami pisać odpowiedniego kodu?

Ten rozdział nie miał na celu prezentacji wszystkich technicznych aspektów intencji. Czytelnikom zainteresowanym szczegółowymi informacjami na ten temat polecam książkę Reto Meiera zatytułowaną *Professional Android 4 Application Development* (Wiley, 2012) oraz dokumentację dla programistów opublikowaną na stronach firmy Google (patrz adres *http://developer.android.com/reference/android/content/Intent.html*).

Informacje na temat intencji udostępnianych przez inne aplikacje można znaleźć na stronie internetowej Open Intents pod adresem *www.openintents.org* (szczególnie cenny jest utrzymywany i udostępniany w tym serwisie rejestr intencji).

SKOROWIDZ

1

A

B

C

D

E

F

G

H

I

L

Ł

M

N

Q

R

S

T

U

V

W

X

Z

NIENAGANNY INTERFEJS UŻYTKOWNIKA TO WARUNEK KONIECZNY DO ODNIESIENIA SUKCESU!

Smashing Magazine to jeden z najpopularniejszych serwisów poświęconych profesjonalnemu tworzeniu stron WWW i interfejsów użytkownika. Korzystają z niego przede wszystkim profesjonalni projektanci i deweloperzy. Ta książka to kompilacja wiedzy najwybitniejszych autorów i ekspertów z zespołu Smashing Magazine. Sięgnij po nią już teraz i uświadom sobie, że nawet najlepsza aplikacja nie sprzeda się, jeżeli nie będzie atrakcyjna wizualnie i łatwa w zastosowaniu. Zanim napiszesz pierwszą linię kodu, przygotuj prototyp i przetestuj go na potencjalnych użytkownikach.

Dzięki lekturze tej książki poznasz struktury aplikacji, mechanizm intencji oraz najlepsze techniki nawigowania wśród dostępnych opcji. Kolejne rozdziały poprowadzą Cię przez szczegóły projektowania komponentów interfejsu, stosowania ikon oraz używania efektów. Część trzecia kompendium jest poświęcona zarządzaniu zasobami Androida, skalowaniu oraz układom interfejsu. Na końcu będziesz miał szansę zapoznać się z najlepszymi wzorcami tworzenia aplikacji na tę platformę. Książka ta jest nieocenionym źródłem informacji na temat tworzenia przyjaznych i wydajnych aplikacji. Musisz ją przeczytać!

Dzięki tej książce:

- poznasz najlepsze wzorce projektowe nawigacji i układu
- wykorzystasz w najlepszy sposób zasoby Androida
- zaprojektujesz funkcjonalny interfejs użytkownika
- odniesiesz sukces na platformie Android!

helion.pl
księgarnia internetowa

Nr katalogowy: 14167

Księgarnia internetowa:
http://helion.pl

Zamówienia telefoniczne:
0 801 339900
0 601 339900

WILEY
Helion

Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach
http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

ISBN 978-83-246-6859-5

Cena 69,00 zł

9 788324 668595

Informatyka w najlepszym wydaniu