

Android 検索インターフェース

[[Android](#)]

- ・ <http://developer.android.com/guide/topics/search/search-dialog.html>

2つの方法

Android では、アプリケーションの検索機能実装を手助けするために、以下のいずれかが用意されている。

1. アクティビティの最上部に表示される、検索ダイアログ (search dialog)
 2. レイアウトの中に配置できる 検索ウィジェット (search widget)
- ・ 上記のいずれも、ユーザーの検索クエリをアプリケーションの特定のアクティビティに届けることができる。
 - ・ ユーザーは、検索ダイアログまたはウィジェットが有効などのアクティビティからでも検索を開始できる。
 - ・ システムは適切なアクティビティを検索の結果に対して実行できる

その他の機能

- ・ ボイス検索
- ・ 最近のクエリに基づく検索サジェスト
- ・ アプリケーションデータの実際に一致する結果に基づく検索サジェスト

基本

- ・ 開始する前に、検索ダイアログまたはウィジェットのどちらを利用するのか決める必要がある
- ・ どちらも同じ検索機能を提供するが、若干方法が異なる。

検索ダイアログ

- ・ [Android](#) に制御される UI コンポーネント
- ・ ユーザーによりアクティベートされたとき、検索ダイアログはアクティビティの最前面に現れる。
- ・ [Android](#) システムがすべてのイベントを制御する
- ・ ユーザーがクエリを実行したとき、システムがアクティビティにクエリを届ける
- ・ ダイアログも検索サジェストを提供できる

検索ウィジェット

- ・ `SearchView` のインスタンス
- ・ レイアウトのどこにでも配置できる
- ・ 通常の `EditText` と同じように振る舞う
- ・ すべての入力を [Android](#) システムが制御するように設定できる
- ・ クエリを適切なアクティビティに届け、検索サジェストを提供するように設定できる
- ・ しかしながら、[Android](#) 3.0 (API レベル 11) 以降でのみ対応

手順の概要

- ・ ユーザーが検索をダイアログまたはウィジェットから実行するとき、システムは、インテントを生成し、ユーザークエリをそこに保持する。

- ・システムは、検索を扱うように宣言されたアクティビティを開始し、インテントを届ける
- ・アプリケーションをこのように検索と結びつけるには、以下の様な手順が必要

検索可能な設定

- ・XML ファイルに検索ダイアログまたはウィジェットの設定を行う
- ・設定の内容は、ボイスサーチ、検索サジェスト、検索ボックスへのヒントテキスト

検索可能なアクティビティ

- ・検索クエリを受け取り、データを検索し、結果を表示するアクティビティ

以下のいずれかを提供する、検索インターフェース

検索ダイアログ

- ・デフォルトでは、検索ダイアログは非表示。
- ・ユーザーがデバイスの検索ボタン（または、UI 上の他のボタン）を押したときに、スクリーンの最前面にあらわれる

検索ウィジェット

- ・アクティビティの任意の場所にウィジェットを置くことができる
- ・しかしながら、通常 Action Bar の Action View とする方が便利

手順

検索可能な設定

- ・最初のステップは、検索可能な設定と呼ばれる XML ファイル
- ・検索ダイアログもしくはウィジェットの設定
- ・検索サジェストやボイスサーチなどがどのように振る舞うのかの定義
- ・このファイルは伝統的に、searchable.xml とし、/res/xml/ に保存されるべき

システムはこのファイルを、SearchableInfo オブジェクトを初期化するのに利用する。アプリケーションからこのオブジェクトを生成できないので、設定を XML ファイルに記述する必要がある

- ・searchable 設定ファイルは、<searchable> 要素を root ノードとして含む必要がある

```
<?xml version="1.0" encoding="utf-8"?>
<searchable xmlns:android="http://schemas.android.com/apk/res/android"
    android:label="@string/app_label"
    android:hint="@string/search_hint" >
</searchable>
```

属性	内容
android:label	唯一の必須項目。アプリケーション名とすべき。Quick Search Box の検索サジェストを有効にするまでは、ユーザーには見えない。

android:hint	必須ではないが、含めるべき。検索ボックスにヒントを表示する。
--------------	--------------------------------

検索サジェスト

- ・ <http://developer.android.com/guide/topics/search/search-dialog.html#SearchSuggestions>

ボイスサーチ

- ・ <http://developer.android.com/guide/topics/search/search-dialog.html#VoiceSearch>

リファレンス

- ・ <http://developer.android.com/guide/topics/search/searchable-config.html>

検索可能なアクティビティの作成

- ・ 検索可能なアクティビティとは、クエリ文字列を元に検索を実行し、検索結果を表示するアクティビティ
- ・ ユーザーが検索をダイアログまたはウィジェットから行ったとき、システムは、検索可能なアクティビティを開始し、検索クエリを ACTION_SEARCH アクションとともにインテントを届ける
- ・ 検索可能なアクティビティはインテントの、QUERY extra からクエリを参照し、検索結果を表示する
- ・ 検索ダイアログもしくはウィジェットを他のどんなアクティビティに含むこともできるのは、システムは、どのアクティビティが検索可能なアクティビティか知っている必要があるため、正確に検索クエリを届けることができる
- ・ まず、検索可能なアクティビティを Android manifest ファイルに宣言する必要がある

検索可能なアクティビティの宣言

- ・ もし、まだ無い場合、検索を行い、結果を返すアクティビティを作成する
 - ・ 検索機能はまだ実装する必要はない。
1. アクティビティが ACTION_SEARCH インテントを受け取るように、<intent-filter> に宣言する
 2. 検索設定を <meta-data> に記述する

例

```
<application ... >
  <activity android:name=".SearchableActivity" >
    <intent-filter>
      <action android:name="android.intent.action.SEARCH" />
    </intent-filter>
    <meta-data android:name="android.app.searchable"
      android:resource="@xml/searchable"/>
  </activity>
  ...
</application>
```

<meta-data> 要素は、android:name 属性を android.app.searchable の値とともに含まなければならない。また、android:resource 属性は、検索設定ファイルへの参照を記述する

検索の実行

- ・ 検索可能アクティビティの宣言ができれば、以下の手順を検索可能アクティビティに適用する。

1. クエリの受け取り
2. データの検索
3. 結果の表示

- ・ 伝統的に、検索結果は、ListView に表示するとよい。
- ・ よって、検索可能アクティビティは、ListActivity を継承させるとよい。
- ・ ListActivity は、ListView を含み、いくつかの便利なメソッドを提供する

クエリの受け取り

- ・ ユーザーが検索をダイアログまたはウィジェットから実行した場合、システムは検索可能アクティビティを開始し、ACTION_SEARCH インテントを送信する
- ・ このインテントは、クエリを QUERY 文字列 extra に含んでいる。
- ・ アクティビティの開始で、インテントをチェックする。

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.search);

    // Get the intent, verify the action and get the query
    Intent intent = getIntent();
    if (Intent.ACTION_SEARCH.equals(intent.getAction())) {
        String query = intent.getStringExtra(SearchManager.QUERY);
        doMySearch(query);
    }
}
```

データの検索

- ・ データのストアと検索はアプリケーション固有の処理
- ・ SQLite の利用や オンラインの利用が考えられる

結果の表示

- ・ 上記でも述べたが、ListView および ListActivity の利用が推奨

検索ダイアログの利用

- ・ 検索ダイアログは、スクリーンの最上部にアプリケーションアイコンとともにフォローディング検索ボックスを提供する
- ・ 検索ダイアログは検索サジェストを提供できる
- ・ システムは検索クエリを検索可能アクティビティに送信する
- ・ しかしながら、Android 3.0 デバイス向けのアプリケーションを開発しているなら、検索ウィジェットを利用することを検討した方がよい
- ・ 検索ダイアログは通常は非表示になっている
- ・ デバイスが検索ボタンを持っている場合、押されることでアクティブになる
- ・ アプリケーションは、onSearchRequested() が呼ばれることでもアクティブになる

- ・ 検索ダイアログを使用可能にするには、システムに検索可能アクティビティが検索ダイアログからクエリを受け取ることをシステムに示さなければいけない
- ・ 検索可能アクティビティを宣言するには、<meta-data> 要素を受け取り側のアクティビティの <activity> 要素に追加する。
- ・ <meta-data> 要素は、検索可能アクティビティクラスを特定する、android:value 属性を含む必要がある。
- ・ また、android:name 属性を android.app.default_searchable という値とともに記述する

例

```
<application ... >
  <!-- 検索可能アクティビティ ; 検索を実行する -->
  <activity android:name=".SearchableActivity" >
    <intent-filter>
      <action android:name="android.intent.action.SEARCH" />
    </intent-filter>
    <meta-data android:name="android.app.searchable"
      android:resource="@xml/searchable" />
  </activity>

  <!-- 検索ダイアログを検索可能アクティビティの中で検索を開始するために検索ダイアログを有効にする
  アクティビティ -->
  <activity android:name=".OtherActivity" ... >
    <!-- 検索可能アクティビティに検索を送信するための検索ダイアログを有効にする -->
    <meta-data android:name="android.app.default_searchable"
      android:value=".SearchableActivity" />
  </activity>
</application>
```

- ・ OtherActivity は、<meta-data> 検索可能アクティビティの <meta-data> を含んでいるので、アクティビティは検索ダイアログを有効化できる。
- ・ ユーザーがこのアクティビティにいる間、デバイスの検索ボタンおよび onSearchRequested() メソッドは、検索ダイアログをアクティブにする
- ・ ユーザーが検索を実行すると、システムは検索可能なアクティビティを開始し、ACTION_SEARCH インテントを送信する
- ・ アプリケーションのすべてのアクティビティで検索ダイアログを提供したい場合、上記の <meta-data> 要素を <application> の子供として挿入する。

検索ダイアログは、現時点で有効になった。検索を実装する準備は整った。

検索ダイアログの呼び出し

- ・ 現在のアクティビティで、検索可能アクティビティを利用するように宣言さえしてあれば、デバイスの検索ボタンで、検索ダイアログを開く
- ・ しかしながら、いくつかのデバイスでは、検索ボタンを持っていないので、検索ボタンがいつでも利用可能であると仮定してはいけない。
- ・ 検索ダイアログを利用する場合、UI に、onSearchRequested() を呼ぶことで検索ダイアログを有効化する、他の検索ボタンを提供すべき

アクティビティライフサイクルにおける検索ダイアログの影響

- ・ 検索ダイアログは、スクリーンの最上部に浮かべられた Dialog
- ・ アクティビティスタックには影響を及ぼさない
- ・ 検索ダイアログが表示されるとき、ライフサイクルメソッド (onPause() など) は呼び出されない。
- ・ アクティビティが入力フォーカスを失い、ダイアログにフォーカスが与えられる

- ・ 検索ダイアログがアクティブになったのを通知したい場合、[onSearchRequested\(\)](#) をオーバーライドする。
- ・ システムがこのメソッドを呼び出すとき、何らかの適切な作業を行うことができる。

例

```
@Override
public boolean onSearchRequested() {
    pauseSomeStuff();
    return super.onSearchRequested();
}
```

- ・ ユーザーが検索を戻すボタンでキャンセルした場合、検索ダイアログは閉じられ、アクティビティが再度フォーカスを得る
- ・ 検索ダイアログが閉じられた場合、[setOnDismissListener\(\)](#) や [setOnCancelListener\(\)](#) を利用して、通知を得るようにできる
- ・ これらは、検索ダイアログが閉じられた時に毎回呼び出される

音声検索の追加

- ・ 音声検索の昨日を検索ダイアログもしくはウィジェットに追加できる。
- ・ 追加するには、検索設定に、`android:voiceSearchMode` 属性を記述する

例

```
<?xml version="1.0" encoding="utf-8"?>
<searchable xmlns:android="http://schemas.android.com/apk/res/android"
    android:label="@string/search_label"
    android:hint="@string/search_hint"
    android:voiceSearchMode="showVoiceSearchButton|launchRecognizer" >
</searchable>
```

- ・ `showVoiceSearchButton` の値は音声検索を有効にするのに必須
- ・ `launchRecognizer` は音声検索ボタンが変換されたテキストを検索アクティビティに戻す音声認識機を起動すべきか設定する
- ・ 音声検索に追加の属性を提供することができる[設定](#)を参照