Developers

Search Overview

Topics

- Creating a Search Interface
- Adding Recent Query Suggestions
- Adding Custom Suggestions

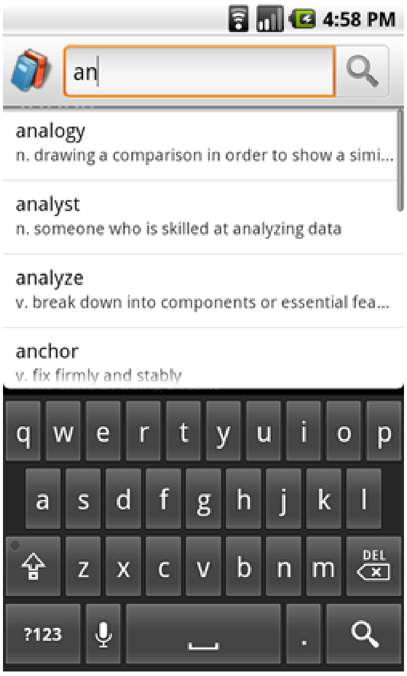
Reference

- Searchable Configuration

Related samples

- Searchable Dictionary

Search is a core user feature on Android. Users should be able to search any data that is available to them, whether the content is located on the device or the Internet. To help create a consistent search experience for users, Android provides a search framework that helps you implement search for your application.



analogy
n. drawing a comparison in order to show a simi...

analyst
n. someone who is skilled at analyzing data

analyze
v. break down into components or essential fea...

anchor
v. fix firmly and stably

Figure 1. Screenshot of a search dialog with custom search suggestions.

The search framework offers two modes of search input: a search dialog at the top of the screen or a search widget (`SearchView`) that you can embed in your activity layout. In either case, the Android system will assist your search implementation by delivering search queries to a specific activity that performs searchs. You can also enable either the search dialog or widget to provide search suggestions as the user types. Figure 1 shows an example of the search dialog with optional search suggestions.

Once you've set up either the search dialog or the search widget, you can:

- Enable voice search
- Provide search suggestions based on recent user queries

- Provide custom search suggestions that match actual results in your application data
- Offer your application's search suggestions in the system-wide Quick Search Box

> **Note**: The search framework does *not* provide APIs to search your data. To perform a search, you need to use APIs appropriate for your data. For example, if your data is stored in an SQLite database, you should use the `android.database.sqlite` APIs to perform searches.
>
> Also, there is no guarantee that a device provides a dedicated SEARCH button that invokes the search interface in your application. When using the search dialog or a custom interface, you must provide a search button in your UI that activates the search interface. For more information, see Invoking the search dialog.

The following documents show you how to use Android's framework to implement search:

Creating a Search Interface

How to set up your application to use the search dialog or search widget.

Adding Recent Query Suggestions

How to provide suggestions based on queries previously used.

Adding Custom Suggestions

How to provide suggestions based on custom data from your application and also offer them in the system-wide Quick Search Box.

Searchable Configuration

A reference document for the searchable configuration file (though the other documents also discuss the configuration file in terms of specific behaviors).

Protecting User Privacy

When you implement search in your application, take steps to protect the user's privacy. Many users consider their activities on the phone—including searches—to be private information. To protect each user's privacy, you should abide by the following principles:

- **Don't send personal information to servers, but if you must, do not log it.**

  Personal information is any information that can personally identify your users, such as their names, email addresses, billing information, or other data that can be reasonably linked to such information. If your application implements search with the assistance of a server, avoid sending personal information along with the search queries. For example, if you are searching for businesses near a zip code, you don't need to send the user ID as well; send only the zip code to the server. If you must send the personal information, you should not log it. If you must log it, protect that data very carefully and erase it as soon as possible.

- **Provide users with a way to clear their search history.**

  The search framework helps your application provide context-specific suggestions while the user types. Sometimes these suggestions are based on previous searches or other actions taken by the user in an earlier session. A user might not wish for previous searches to be revealed to other device users, for instance, if the user shares the device with a friend. If your application provides suggestions that can reveal previous search activities, you should implement the ability for the user to clear the search history. If you are using `SearchRecentSuggestions`, you can simply call the `clearHistory()` method. If you are implementing custom suggestions, you'll need to provide a similar "clear history" method in your content provider that the user can execute.