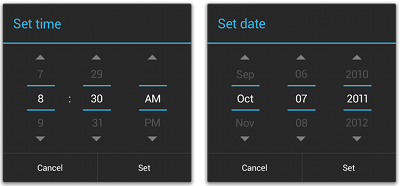
Pickers

Android provides controls for the user to pick a time or pick a date as ready-to-use dialogs. Each picker provides controls for selecting each part of the time (hour, minute, AM/PM) or date (month, day, year). Using these pickers helps ensure that your users can pick a time or date that is valid, formatted correctly, and adjusted to the user's locale.



We recommend that you use [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) to host each time or date picker. The [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) manages the dialog lifecycle for you and allows you to display the pickers in different layout configurations, such as in a basic dialog on handsets or as an embedded part of the layout on large screens.

Although [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\DialogFragment.html) was first added to the platform in Android 3.0 (API level 11), if your app supports versions of Android older than 3.0—even as low as Android 1.6—you can use the[DialogFragment](file:///G:\Studio\SDK\docs\reference\android\support\v4\app\DialogFragment.html) class that's available in the [support library](file:///G:\Studio\SDK\docs\tools\support-library\index.html) for backward compatibility.

**Note:** The code samples below show how to create dialogs for a time picker and date picker using the [support library](file:///G:\Studio\SDK\docs\tools\support-library\index.html) APIs for [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html). If your app's [minSdkVersion](file:///G:\\Studio\\SDK\\docs\\guide\\topics\\manifest\\uses-sdk-element.html" \l "min) is 11 or higher, you can instead use the platform version of [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\DialogFragment.html).

## Creating a Time Picker

To display a [TimePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\TimePickerDialog.html) using [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html), you need to define a fragment class that extends [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) and return a [TimePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\TimePickerDialog.html) from the fragment's [onCreateDialog()](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html" \l "onCreateDialog(android.os.Bundle))method.

**Note:** If your app supports versions of Android older than 3.0, be sure you've set up your Android project with the support library as described in [Setting Up a Project to Use a Library](file:///G:\Studio\SDK\docs\tools\support-library\setup.html).

### Extending DialogFragment for a time picker

To define a [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) for a [TimePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\TimePickerDialog.html), you must:

* Define the [onCreateDialog()](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html" \l "onCreateDialog(android.os.Bundle)) method to return an instance of [TimePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\TimePickerDialog.html)
* Implement the [TimePickerDialog.OnTimeSetListener](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\TimePickerDialog.OnTimeSetListener.html) interface to receive a callback when the user sets the time.

Here's an example:

public static class TimePickerFragment extends DialogFragment  
                            implements TimePickerDialog.OnTimeSetListener {  
  
    @Override  
    public Dialog onCreateDialog(Bundle savedInstanceState) {  
        // Use the current time as the default values for the picker  
        final Calendar c = Calendar.getInstance();  
        int hour = c.get(Calendar.HOUR\_OF\_DAY);  
        int minute = c.get(Calendar.MINUTE);  
  
        // Create a new instance of TimePickerDialog and return it  
        return new TimePickerDialog(getActivity(), this, hour, minute,  
                DateFormat.is24HourFormat(getActivity()));  
    }  
  
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
        // Do something with the time chosen by the user  
    }  
}

See the [TimePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\TimePickerDialog.html) class for information about the constructor arguments.

Now all you need is an event that adds an instance of this fragment to your activity.

### Showing the time picker

Once you've defined a [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) like the one shown above, you can display the time picker by creating an instance of the [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) and calling [show()](file:///G:\Studio\SDK\docs\reference\android\support\v4\app\DialogFragment.html#show(android.support.v4.app.FragmentManager, java.lang.String)).

For example, here's a button that, when clicked, calls a method to show the dialog:

<Button   
    android:layout\_width="wrap\_content"   
    android:layout\_height="wrap\_content"  
    android:text="@string/pick\_time"   
    android:onClick="showTimePickerDialog" />

When the user clicks this button, the system calls the following method:

public void showTimePickerDialog(View v) {  
    DialogFragment newFragment = new TimePickerFragment();  
    newFragment.show(getSupportFragmentManager(), "timePicker");  
}

This method calls [show()](file:///G:\Studio\SDK\docs\reference\android\support\v4\app\DialogFragment.html#show(android.support.v4.app.FragmentManager, java.lang.String)) on a new instance of the [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) defined above. The [show()](file:///G:\Studio\SDK\docs\reference\android\support\v4\app\DialogFragment.html#show(android.support.v4.app.FragmentManager, java.lang.String))method requires an instance of [FragmentManager](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\FragmentManager.html) and a unique tag name for the fragment.

**Caution:** If your app supports versions of Android lower than 3.0, be sure that you call[getSupportFragmentManager()](file:///G:\Studio\SDK\docs\reference\android\support\v4\app\FragmentActivity.html#getSupportFragmentManager()) to acquire an instance of [FragmentManager](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\FragmentManager.html). Also make sure that your activity that displays the time picker extends [FragmentActivity](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\FragmentActivity.html) instead of the standard[Activity](file:///G:\Studio\SDK\docs\reference\android\app\Activity.html) class.

## Creating a Date Picker

Creating a [DatePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\DatePickerDialog.html) is just like creating a [TimePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\TimePickerDialog.html). The only difference is the dialog you create for the fragment.

To display a [DatePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\DatePickerDialog.html) using [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html), you need to define a fragment class that extends [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) and return a [DatePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\DatePickerDialog.html) from the fragment's [onCreateDialog()](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html" \l "onCreateDialog(android.os.Bundle))method.

**Note:** If your app supports versions of Android older than 3.0, be sure you've set up your Android project with the support library as described in [Setting Up a Project to Use a Library](file:///G:\Studio\SDK\docs\tools\support-library\setup.html).

### Extending DialogFragment for a date picker

To define a [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) for a [DatePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\DatePickerDialog.html), you must:

* Define the [onCreateDialog()](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html" \l "onCreateDialog(android.os.Bundle)) method to return an instance of [DatePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\DatePickerDialog.html)
* Implement the [DatePickerDialog.OnDateSetListener](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\DatePickerDialog.OnDateSetListener.html) interface to receive a callback when the user sets the date.

Here's an example:

public static class DatePickerFragment extends DialogFragment  
                            implements DatePickerDialog.OnDateSetListener {  
  
    @Override  
    public Dialog onCreateDialog(Bundle savedInstanceState) {  
        // Use the current date as the default date in the picker  
        final Calendar c = Calendar.getInstance();  
        int year = c.get(Calendar.YEAR);  
        int month = c.get(Calendar.MONTH);  
        int day = c.get(Calendar.DAY\_OF\_MONTH);  
  
        // Create a new instance of DatePickerDialog and return it  
        return new DatePickerDialog(getActivity(), this, year, month, day);  
    }  
  
    public void onDateSet(DatePicker view, int year, int month, int day) {  
        // Do something with the date chosen by the user  
    }  
}

See the [DatePickerDialog](file:///G:\\Studio\\SDK\\docs\\reference\\android\\app\\DatePickerDialog.html) class for information about the constructor arguments.

Now all you need is an event that adds an instance of this fragment to your activity.

### Showing the date picker

Once you've defined a [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) like the one shown above, you can display the date picker by creating an instance of the [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) and calling [show()](file:///G:\Studio\SDK\docs\reference\android\support\v4\app\DialogFragment.html#show(android.support.v4.app.FragmentManager, java.lang.String)).

For example, here's a button that, when clicked, calls a method to show the dialog:

<Button   
    android:layout\_width="wrap\_content"   
    android:layout\_height="wrap\_content"  
    android:text="@string/pick\_date"   
    android:onClick="showDatePickerDialog" />

When the user clicks this button, the system calls the following method:

public void showDatePickerDialog(View v) {  
    DialogFragment newFragment = new DatePickerFragment();  
    newFragment.show(getSupportFragmentManager(), "datePicker");  
}

This method calls [show()](file:///G:\Studio\SDK\docs\reference\android\support\v4\app\DialogFragment.html#show(android.support.v4.app.FragmentManager, java.lang.String)) on a new instance of the [DialogFragment](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\DialogFragment.html) defined above. The [show()](file:///G:\Studio\SDK\docs\reference\android\support\v4\app\DialogFragment.html#show(android.support.v4.app.FragmentManager, java.lang.String))method requires an instance of [FragmentManager](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\FragmentManager.html) and a unique tag name for the fragment.

**Caution:** If your app supports versions of Android lower than 3.0, be sure that you call[getSupportFragmentManager()](file:///G:\Studio\SDK\docs\reference\android\support\v4\app\FragmentActivity.html#getSupportFragmentManager()) to acquire an instance of [FragmentManager](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\FragmentManager.html). Also make sure that your activity that displays the time picker extends [FragmentActivity](file:///G:\\Studio\\SDK\\docs\\reference\\android\\support\\v4\\app\\FragmentActivity.html) instead of the standard[Activity](file:///G:\Studio\SDK\docs\reference\android\app\Activity.html) class.

[Get news & tips](javascript:;)