Saving Key-Value Sets

If you have a relatively small collection of key-values that you'd like to save, you should use the `SharedPreferences` (/reference/android/content/SharedPreferences.html) APIs. A `SharedPreferences` (/reference/android/content/SharedPreferences.html) object points to a file containing key-value pairs and provides simple methods to read and write them. Each `SharedPreferences` (/reference/android/content/SharedPreferences.html) file is managed by the framework and can be private or shared.

This class shows you how to use the `SharedPreferences` (/reference/android/content/SharedPreferences.html) APIs to store and retrieve simple values.

THIS LESSON TEACHES YOU TO

1. Get a Handle to a SharedPreferences
2. Write to Shared Preferences
3. Read from Shared Preferences

YOU SHOULD ALSO READ

- Using Shared Preferences

> **Note:** The `SharedPreferences` (/reference/android/content/SharedPreferences.html) APIs are only for reading and writing key-value pairs and you should not confuse them with the `Preference` (/reference/android/preference/Preference.html) APIs, which help you build a user interface for your app settings (although they use `SharedPreferences` (/reference/android/content/SharedPreferences.html) as their implementation to save the app settings). For information about using the `Preference` (/reference/android/preference/Preference.html) APIs, see the Settings (/guide/topics/ui/settings.html) guide.

Get a Handle to a SharedPreferences

You can create a new shared preference file or access an existing one by calling one of two methods:

- `getSharedPreferences()` — Use this if you need multiple shared preference files identified by name, which you specify with the first parameter. You can call this from any `Context` in your app.
- `getPreferences()` — Use this from an `Activity` if you need to use only one shared preference file for the activity. Because this retrieves a default shared preference file that belongs to the activity, you don't need to supply a name.

For example, the following code is executed inside a `Fragment` (/reference/android/app/Fragment.html). It accesses the shared preferences file that's identified by the resource string `R.string.preference_file_key` and opens it using the private mode so the file is accessible by only your app.

```
Context context = getActivity();
SharedPreferences sharedPref = context.getSharedPreferences(
        getString(R.string.preference_file_key), Context.MODE_PRIVATE);
```

When naming your shared preference files, you should use a name that's uniquely identifiable to your app, such as `"com.example.myapp.PREFERENCE_FILE_KEY"`

Alternatively, if you need just one shared preference file for your activity, you can use the `getPreferences()` (/reference/android/app/Activity.html#getPreferences(int)) method:

```
SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE_PRIVATE);
```

> **Caution:** If you create a shared preferences file with `MODE_WORLD_READABLE` (/reference/android/content/Context.html#MODE_WORLD_READABLE) or `MODE_WORLD_WRITEABLE` (/reference/android/content/Context.html#MODE_WORLD_WRITEABLE), then any other apps that know the file identifier can access your data.

Write to Shared Preferences

To write to a shared preferences file, create a `SharedPreferences.Editor` (/reference/android/content/SharedPreferences.Editor.html) by calling `edit()` (/reference/android/content/SharedPreferences.html#edit()) on your `SharedPreferences` (/reference/android/content/SharedPreferences.html).

Pass the keys and values you want to write with methods such as `putInt()` (/reference/android/content/SharedPreferences.Editor.html#putInt(java.lang.String, int)) and `putString()` (/reference/android/content/SharedPreferences.Editor.html#putString(java.lang.String, java.lang.String)). Then call `commit()` (/reference/android/content/SharedPreferences.Editor.html#commit()) to save the changes. For example:

```
SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE_PRIVATE);
SharedPreferences.Editor editor = sharedPref.edit();
editor.putInt(getString(R.string.saved_high_score), newHighScore);
editor.commit();
```

Read from Shared Preferences

To retrieve values from a shared preferences file, call methods such as `getInt()` (/reference/android/content/SharedPreferences.html#getInt(java.lang.String, int)) and `getString()` (/reference/android/content/SharedPreferences.html#getString(java.lang.String, java.lang.String)), providing the key for the value you want, and optionally a default value to return if the key isn't present. For example:

```
SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE_PRIVATE);
int defaultValue = getResources().getInteger(R.string.saved_high_score_default);
long highScore = sharedPref.getInt(getString(R.string.saved_high_score), defaultValue)
```