

Kurs Java Android

Shared Preferences

Shared Preferences

SharedPreferences are not the same as the typical user preferences where one, for example selects the preferred default sound, text size, colours, etc. for an app.

SharedPreferences are key-value pairs of primitive data types that are saved in a file within an apps file structure.

Shared Preferences

You can save your preferences data in a single file or in multiple files, depending on your needs. The process is the same except for the `SharedPreferences` object that you get.

Shared Preferences

You can then access this file from anywhere within the app to either put data into the file or take data out of the file. You can't access the file from another app so it's pretty secure from that point of view.

Shared Preferences

Retrieving the data

Once you have the SharedPreferences object, you can use a number of get<type> methods to retrieve the values.

Saving your data

Saving the data is pretty straight forward too. Simply use the SharedPreferences.Editor class to get an Editor object by calling edit() on the relevant SharedPreferences object. Then use the put<type> methods to save the data, finalising the process by calling apply() or commit() on the Editor object.

Shared Preferences

apply()

This saves your data into memory immediately and saves the data to disk on a separate thread. So there is no chance of blocking the main thread (your app won't hang).

It is the preferred technique but has only been available since Gingerbread (API 9, Android 2.3).

Shared Preferences

commit()

Calling this will save the data to the file however, the process is carried out in the thread that called it, stopping everything else until the save is complete. It returns true on successful completion, false on failure.

Use commit() if you need confirmation of the success of saving your data or if you are developing for pre-Gingerbread devices. commit() has been available since API 1

Zadanie 1

Aplikacja która posiada dwa activity. W pierwszym activity widoczne są informacje (TextView) - język, kraj, oraz imię i nazwisko użytkownika. Dane można edytować przechodząc do drugiego activity za pomocą guzika. W drugim activity powinna być możliwość edycji danych (EditText'y) oraz języka (Spinner). Dwa guziki(Button) Save i Cancel służące do zapisu danych do SharedPreferences i odczytu.

Zadanie 1

- dodatkowy feature: jeśli po uruchomieniu w `sharedPreferences` nie ma dwóch podstawowych kluczy, to od razu (automatycznie) uruchom activity edycji danych w celu ich uzupełnienia. (jesli użytkownik nie wpisze imienia i nazwiska, nie pozwól mu powrócić)
- dodatkowy feature: zdefiniuj wartości Języki / Kraje korzystając z Enum'a.
- jeśli potrafisz: prześlij dane do drugiego activity intentem (w drugim activity nie ładuj danych z `sharedPreferences`).
Uruchom activity używając `startActivityForResult`, jeśli dane zostaną zapisane - wyświetl toast - dane zostały zmienione poprawnie, jeśli nie "dane nie zostały zmienione".

Zadanie 2

Aplikacja która posiada menu konfiguracyjne dotyczące kolorów guzików. Przechodząc do ustawień (drugie activity, podobnie jak w zadaniu 1) mamy możliwość edycji koloru i rozmiaru guzików, a także ich odstępów między sobą.

Zadanie 3

Aplikacja posiadająca menu logowania, oraz menu rejestracji. Po zarejestrowaniu (do rejestracji niezbędne są login i hasło - login musi być unikalny) dodajemy użytkownika do `sharedPreferences` które zawierają użytkowników. Po zalogowaniu użytkownik ma możliwość edycji danych personalnych (login, hasło, imie, nazwisko, kraj, język, płeć, data urodzenia).