Rajdip Sanyal (05/09/2024)
sanyalrajdip864@gmail.com

Android Framework with Java

Shared Preferences

In android development, SharedPreferences is a mechanism for storing simple key-value pairs of data. It is often used to store small amounts of data, such as user preferences, settings, or simple application state.

Key Features of SharedPreferences:

1. Shared Preference data is stored in a key-value format, making it easy to retrieve and store values using a unique key.

2. Data stored in SharedPreferences persists across user sessions. Once saved, it remains available even if the app is closed and reopened.

3. Shared Preference Supports basic data types like String, int, boolean, float, and long. It does not support complex data types like objects or arrays directly.

4. Data is stored in an XML file in the app's private storage area. Each Shared Preferences instance corresponds to a file.

```
SharedPreferences sp =getSharedPreferences( name: "mytodo",MODE_PRIVATE);
SharedPreferences.Editor myEdit =sp.edit();
```

```
    txtview2 = findViewById(R.id.txt2);
    addTodobtn = findViewById(R.id.addtodo);
addTodobtn.setOnClickListener(e -> {
        String todo = txtview2.getText().toString();
        myEdit.putString( s: "name", todo);
        Log.i( tag: "meow",  msg: "Inserted data");
        myEdit.commit();
        Toast.makeText( context: this,getData(),Toast.LENGTH_SHORT).show();
        doWork();
});
```

Background Processing in Android

By default, application code runs within the main thread. Every statement is therefore executed in sequence. If you perform an extended-lasting operation, the appliance blocks until the corresponding operation has finished.

Android modifies the interface and handles input events from one single thread, called the most thread. Android collects all events during this thread during a queue and processes this queue with an instance of the Looper class. The Android Operating System uses the Thread class to perform asynchronous tasks like the ones mentioned above the header image. The java.util.concurrent is also provided by it to perform certain tasks, something which we call background. For instance, by using the ThreadPools and Executor classes.

What Exactly is Background Work?

An app is taken into account to be running within the background as long as each of the subsequent conditions is satisfied

1. The app is not in the running state and no foreground services are performed, while the device is being used.
2. None of the app's activities (pages) or fragments are currently visible to the one using it.

A simple glyph to understand this process could be taken from the below Image: