

SESSION 07

```
export PATH=$PATH:/Users/sajib/Library/Android/sdk/  
platform-tools/
```

#DEBUGGING

- pause your program as needed
- view the call stack
- resume program
- click gutter area to set a breakpoint
- variables section shows a lot of information about all variables
- debugger search
 - debug search filter save
- remove all unwanted bloat debugging messages
- right click on a line -> fold lines like this
- debug instead of run
- debug breakpoint
 - drag breakpoints
- conditional breakpoint
 - conditional breakpoint more menu
 - “Disable breakpoint until”
- suspend breakpoint thread -> All vs one
- breakpoint condition “Evaluate and log”
- breakpoint groups
 - breakpoint > more > breakpoint is grouped
 - enable / disable breakpoint group
- step next vs step into
- drop frame (for Android 10+) : does not always work
- follow the lifecycle of a certain variable
 - right click on variable and add label

- go to watches and search for the label (you may find it)
- click on line gutter and you'll go to that line directly in the debugger
- evaluate expression button > evaluate expression dialog
 - write any evaluation
 - create variables
 - create statements
 - hit evaluate
 - live inspection shell of your app
- Analyze > analyze stack trace

#SERVICES

- used to perform long running tasks in background
 - does it run in the background as a separate thread?
- does not have an UI
- extends Service
 - onStart(Intent intent, int startId) : deprecated
 - onBind(Intent intent)
 - onDestroy()
 - onStartCommand(Intent intent, int flags, int startId) : gets executed when you start a service
- start service with an Intent
- declare service in manifest file:
 - <service
 android:name="path.to.package.service" />
 - life-cycle of the service has to be managed by the service itself
- use stopSelf() within the service to stop the service
- startService(intentService)
- stopService(stopService)

- the onDestroy is called