



# Android Development Guidelines and Good Practices

wizeline.com | [sergio.lagunas@wizeline.com](mailto:sergio.lagunas@wizeline.com)

**WIZELINE**<sup>®</sup>



# Shortcuts

- Every IDE and OS has its shortcuts and tricks to code faster





# Shortcuts

Description	Windows/Linux	Mac
Project quick fix (show intention actions and quick fixes)	Alt + Enter	Option + Enter
Reformat code	Control + Alt + L	Command + Option + L
Auto-indent lines	Control + Alt + I	Control + Option + I
Indent/unindent lines	Tab/Shift + Tab	Tab/Shift + Tab
Smart line join	Control + Shift + J	Control + Shift + J
Smart line split	Control + Enter	Command + Enter
Start new line	Shift + Enter	Shift + Enter
Next/previous highlighted error	F2 / Shift + F2	F2 / Shift + F2

<https://developer.android.com/studio/intro/keyboard-shortcuts.html>

<https://youtu.be/Y2GC6P5hPeA>



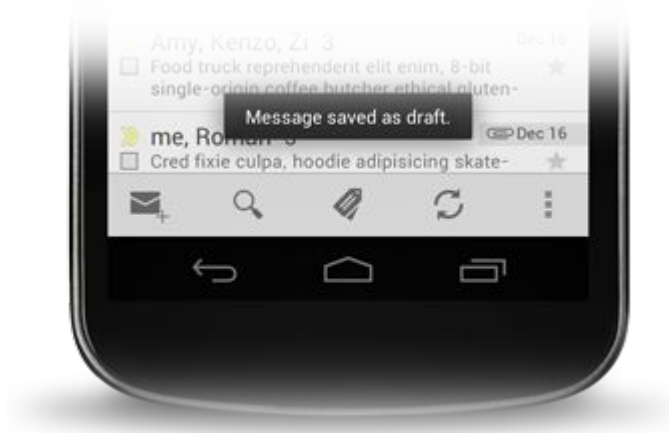
# Think about the user

- They doesn't have a cellphone as cool as yours
- Think about the processor, memory consumption, and data consumption of your app
- Try to look and avoid memory leaks on your app (Use LeakCanary) (Go to IDE and do an example of how to check if you have any)
- Avoid deep levels on your layouts (Learn how to use ConstraintLayout)



# Give your user feedback

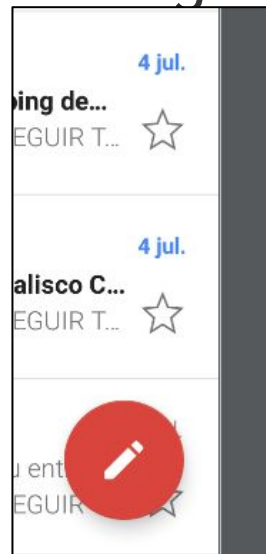
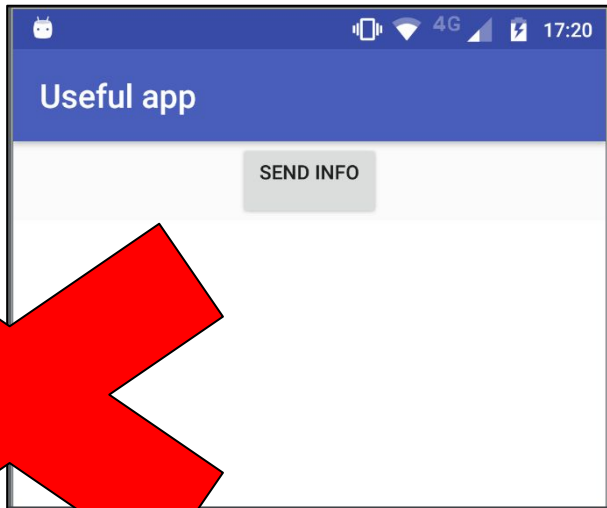
- Feedback: The user have to know what he just did





# Give your user feedforward

- Feedforward: The user have to know what he will do when he presses or interact with something



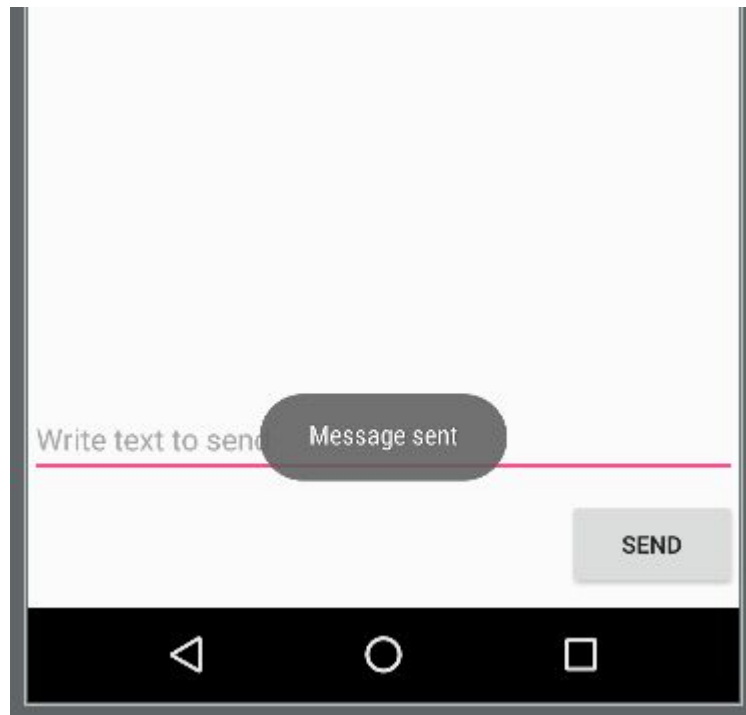
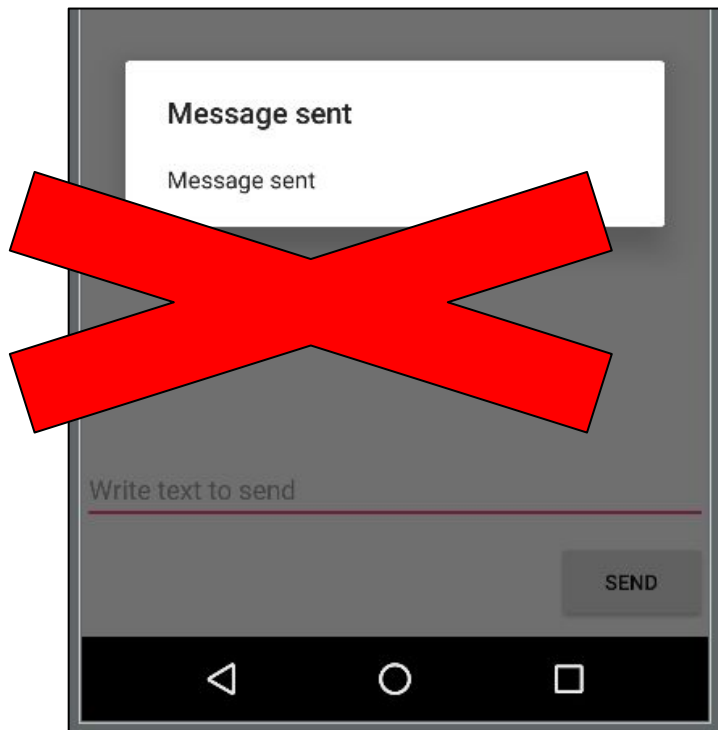


# Toast, Snackbar, and Dialog

- Toast: Message shown at the bottom of the screen
- Snackbar: Like the toast, but it can be swiped off the screen, and can contain icons and actions
- Dialog: Inform users about a specific task and may contain critical information; use it only when necessary, because it blocks user's flow



# Toast, Snackbar, and Dialog







# Architecture





# Architecture

## Act early

- Decisions you make stick with you
- Early design decisions deeply influence your app
- Your basic architecture makes it easier or harder to think about the problems you need to solve





# Activities and Fragment



# Activities and Fragment

Q: Which one do I use?



# Activities and Fragment

~~Q: Which one do I use?~~

A: Fragments are encapsulated parts of an Activity

ListFragment      DetailsFragment



MainActivity



# Fragments and views

Q: Which one do I use?



# Fragments and views

~~Q. Which one do I use?~~

A:

- <Fragment> tag does not make a fragment a view
- Activities/Fragments are lifecycle constructs
- Use both to keep responsibilities clear



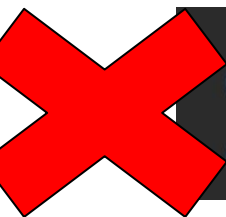
# Gradle

- Learn what is gradle, and how to use it (Omar will talk more about this)





- Avoid Maven dynamic dependency resolution (not use '+' when importing a library)



```
compile 'com.jakewharton.timber:timber:4.+'  
compile "com.jakewharton:butterknife:+"
```

- If you are going to use a library, be sure that it will be for a long time (history, and risk)



# Proguard

It helps you to:

- Remove unused code and resources
- Remove unused classes
- Obfuscate code

# THANK YOU



**WIZELINE**<sup>®</sup>

[sales@wizeline.com](mailto:sales@wizeline.com)





# Android Studio tricks

- Multi selection (go to IDE)
- Multi paste (go to IDE)
- Tab vs Enter (go to IDE)
- Auto completion, and smarter auto completions (go to IDE)
- Selection (Option + up)
- Logs (i, t, m, r, d, e)
- Search template + inspection
- Wrap and unwrap your code with `Cmd+alt+T` , and



# Android Studio tricks

- Wrap and unwrap your code with `Cmd+alt+T` , and `Cmd+Shift+delete`
- Bookmarks
- Postfix (for, fori, nn, etc) (`Cmd + J`)
- Auto Import(Editor-General-Auto Import)
- Show line numbers(Editor-General-Appearance)
- (almost) Every screen has a find words