~Diva Edition~

Notifications and Threads

8/13/2015

Jared Poelman

Real Quick: Builder Pattern

- When: you’re making an object with many fields
- Two naive solutions:
  1. Make a class with a telescoping constructors
  2. Make a class with a trillion setters and getters

1. TELESCOPING CONSTRUCTORS

```
public Food(String name)

public Food(String name, int calories)

public Food(String name, int calories, int
servingSize)

public Food(String name, int calories, int
servingSize, int fat)
```

2. SETTERS & GETTERS

setName()

getName()

setServingSize()

getServingSize()

setCalories()

getCalories()

setFat()

getFat()

……………………….and so on

BUILDER!

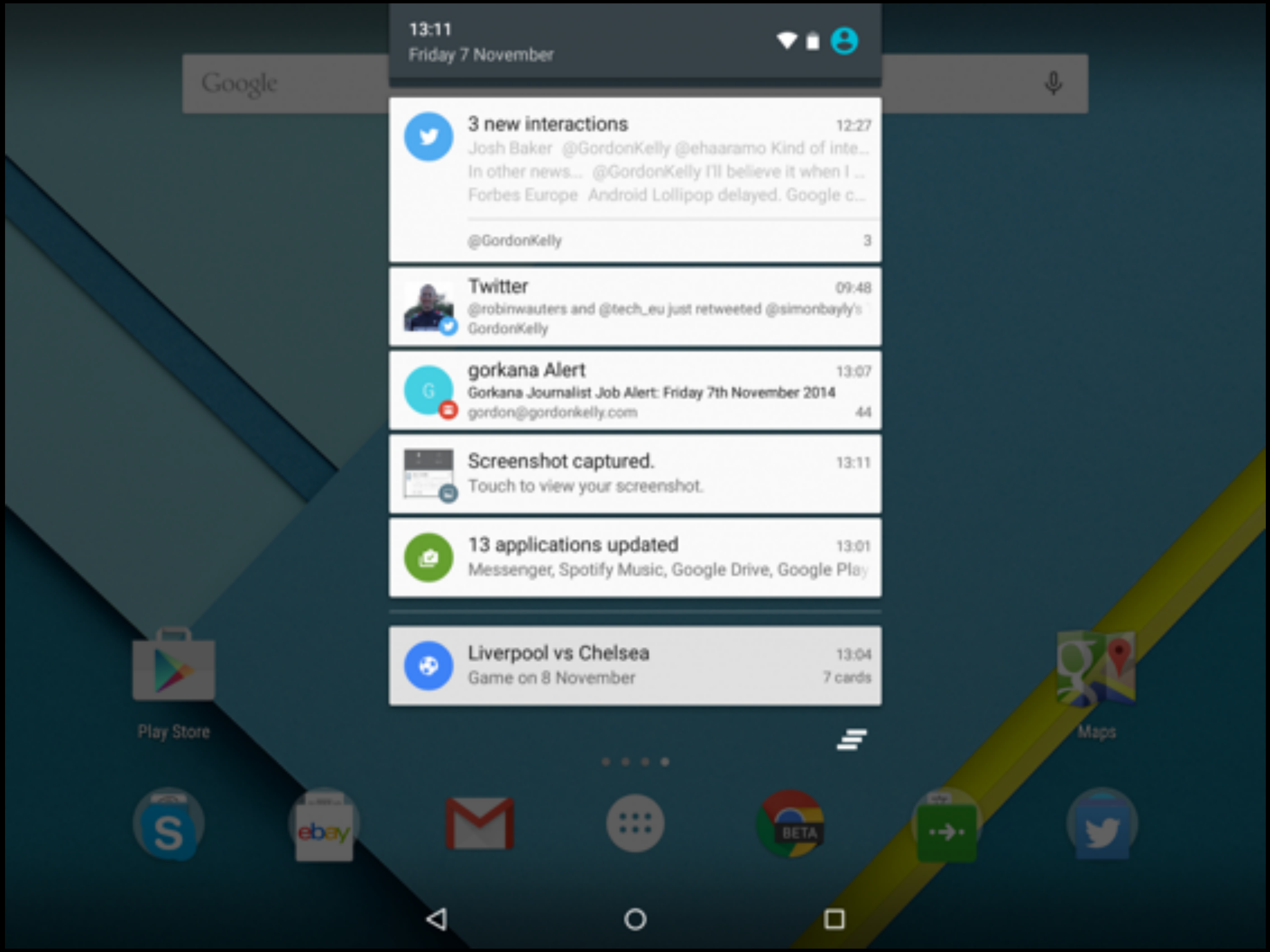
```
// constructor sets required fields (if any)

public Builder(String name) {}

// builder method

public Builder setCalories(int calories) {

    mCalories = calories;

    return this;

}

// build method

public Food build() {

    return new Food(this);  //constructor that takes a Builder

}
```

Using the Builder

```
Food food = new Food.Builder(“asparagus”)
    .setCalories(3)
    .setServingSize(1)
    .setFat(2)
    .build();
```

Notification Area vs Drawer

Both are system controlled



13:11
Friday 7 November

3 new interactions 12:27
Josh Baker @GordonKelly @ehaaramo Kind of inte...
In other news... @GordonKelly I'll believe it when I ...
Forbes Europe Android Lollipop delayed. Google c...
@GordonKelly 3

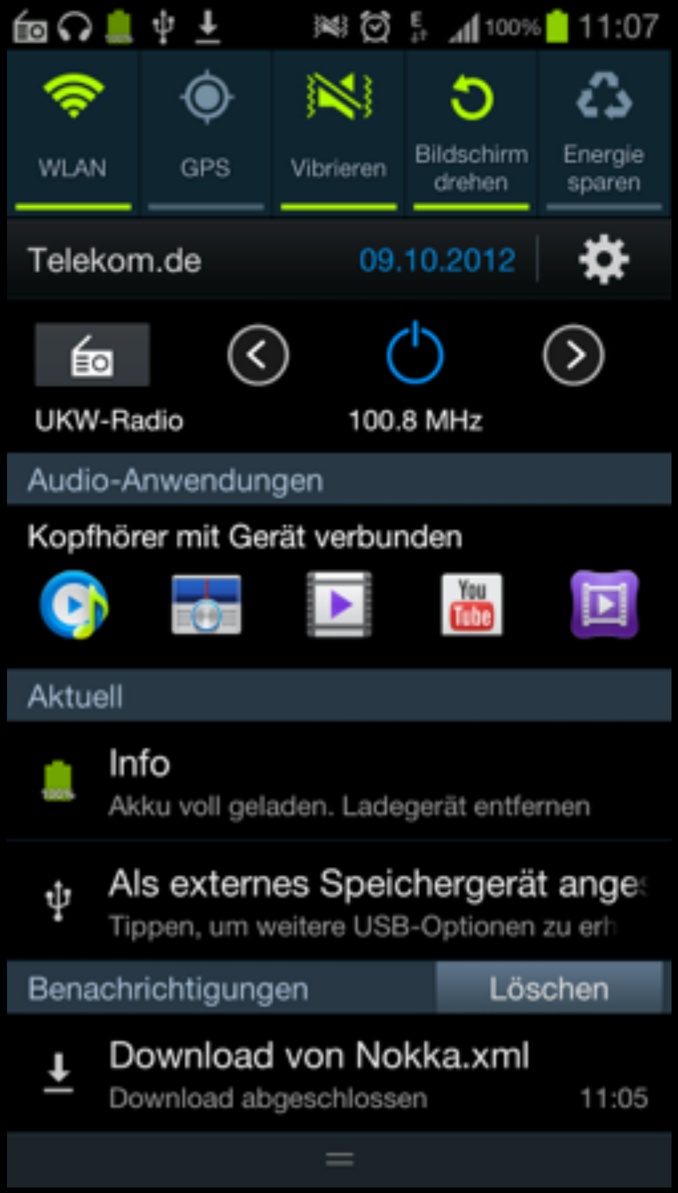
Twitter 09:48
@robinwauters and @tech_eu just retweeted @simonbayly's GordonKelly

gorkana Alert 13:07
Gorkana Journalist Job Alert: Friday 7th November 2014
gordon@gordonkelly.com 44

Screenshot captured. 13:11
Touch to view your screenshot.

13 applications updated 13:01
Messenger, Spotify Music, Google Drive, Google Pla

Liverpool vs Chelsea 13:04
Game on 8 November 7 cards



11:07

Telekom.de 09.10.2012

UKW-Radio 100.8 MHz

Audio-Anwendungen

Kopfhörer mit Gerät verbunden

Aktuell

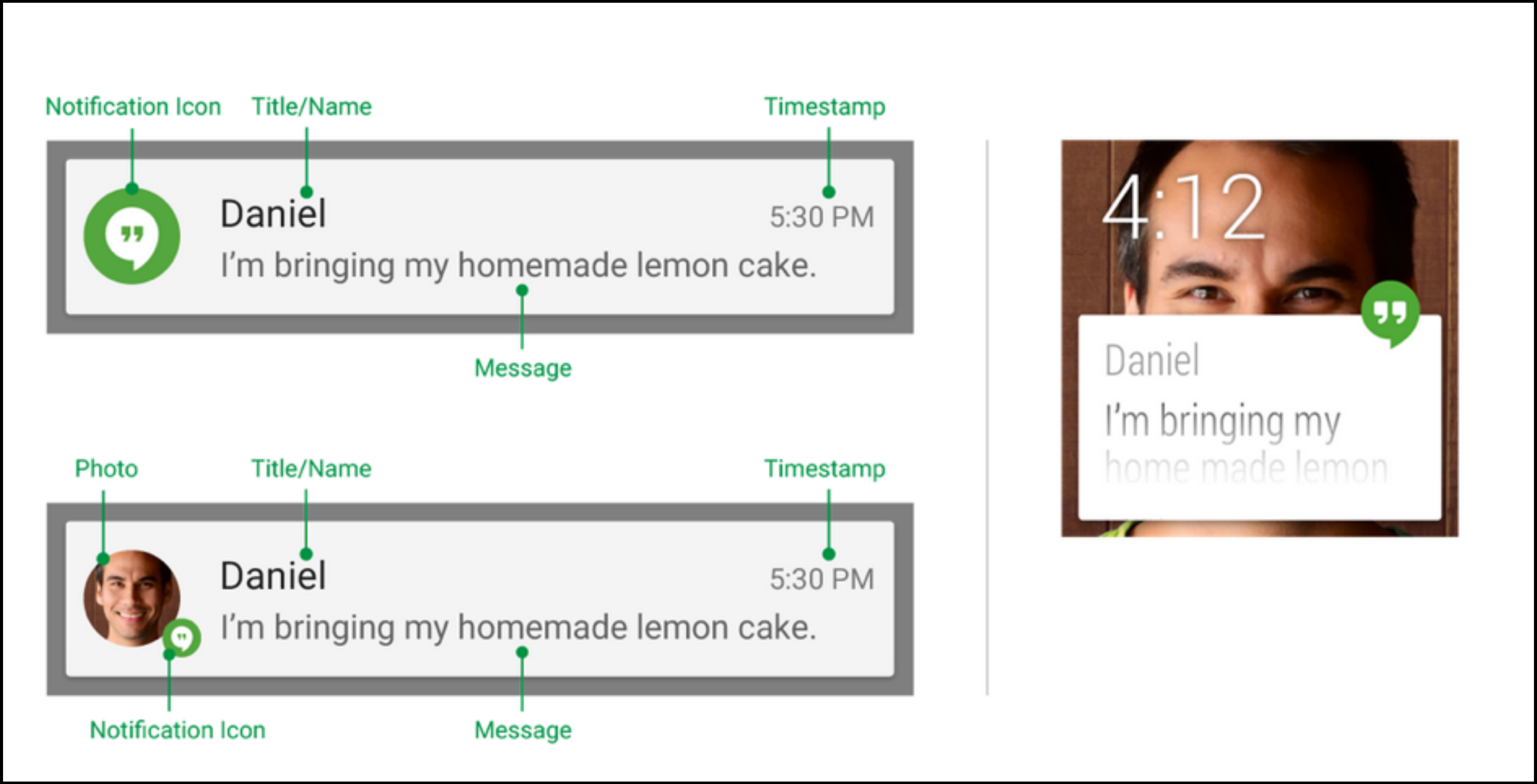
Info
Akku voll geladen. Ladegerät entfernen

Als externes Speichergerät ange
Tippen, um weitere USB-Optionen zu erh

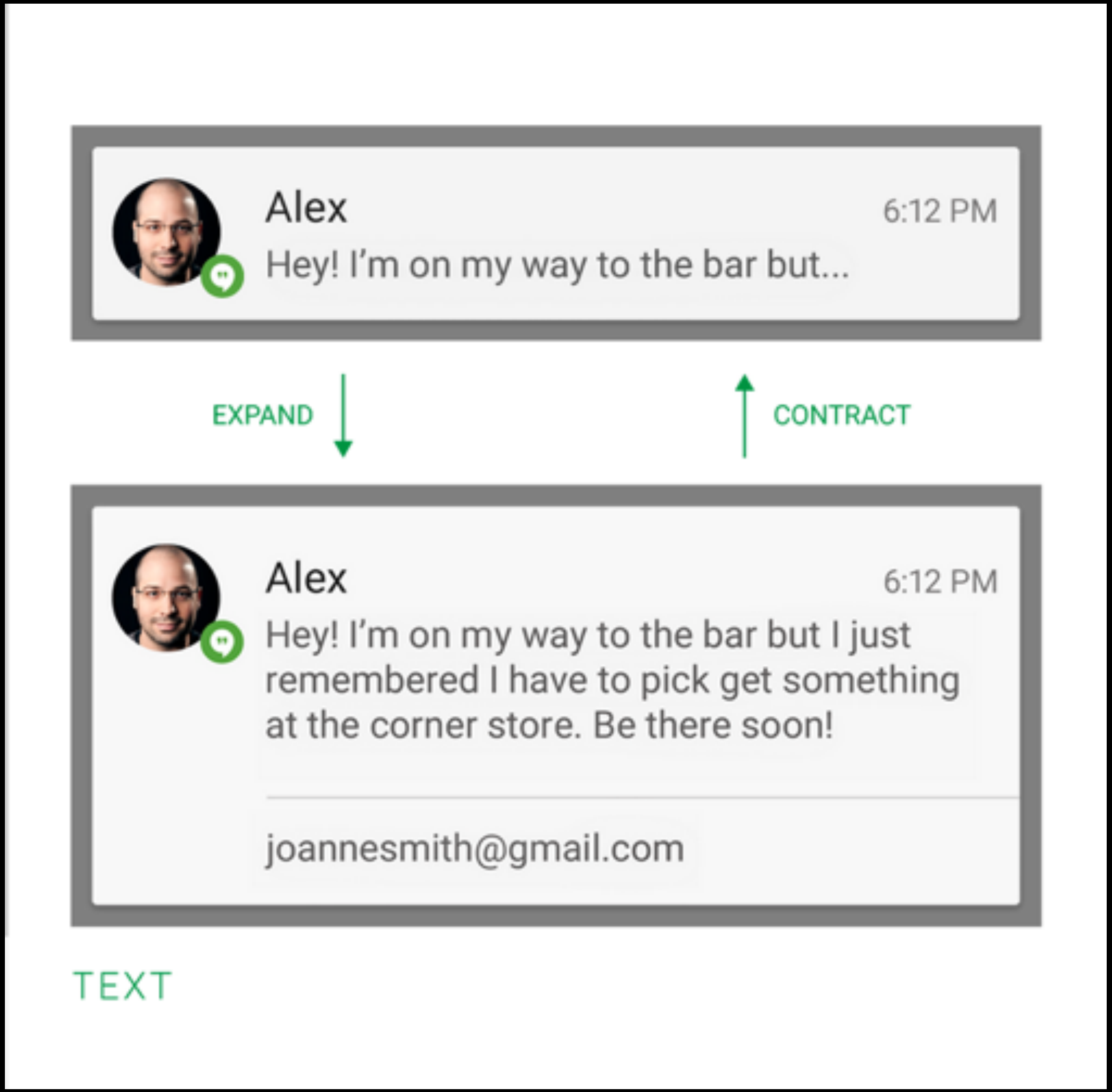
Benachrichtigungen Löschen

Download von Nokka.xml
Download abgeschlossen 11:05

Anatomy


Notification Icon
Title/Name
Timestamp
Daniel
5:30 PM
I'm bringing my homemade lemon cake.
Message
Photo
Title/Name
Timestamp
Daniel
5:30 PM
I'm bringing my homemade lemon cake.
Notification Icon
Message
4:12
Daniel
I'm bringing my
home made lemon

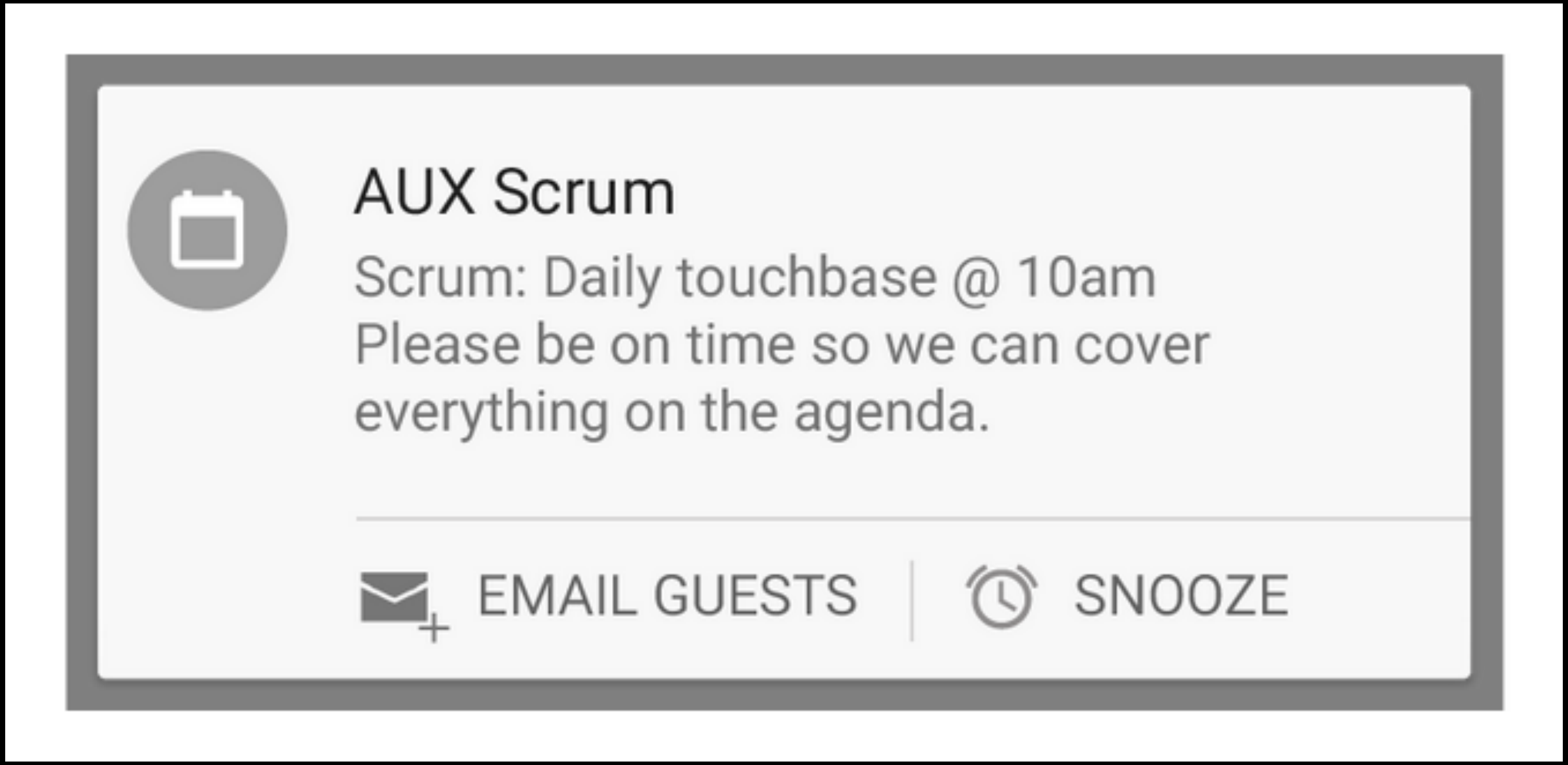
Expanded View



Hey! I'm on my way to the bar but I just remembered I have to pick get something at the corner store. Be there soon!

joannesmith@gmail.com

Actions



AUX Scrum

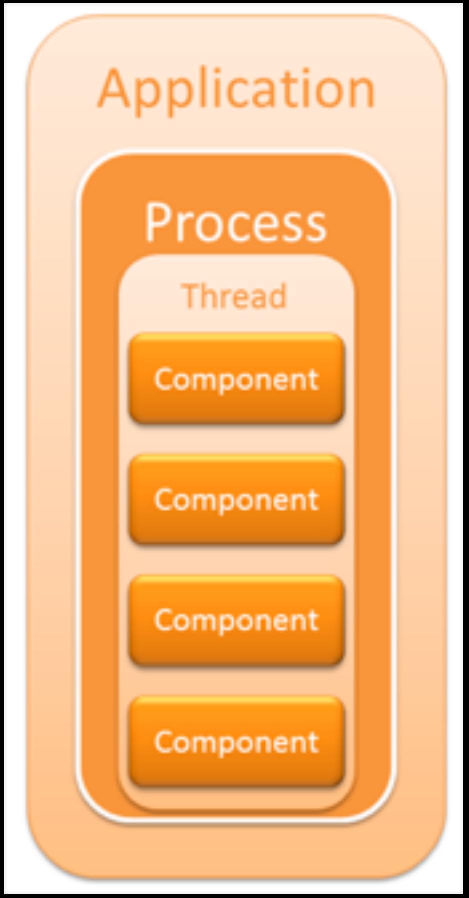
Scrum: Daily touchbase @ 10am
Please be on time so we can cover everything on the agenda.

android.os.Build.VERSION

- Notification features are added constantly
- Accessing current version can be done programmatically to make sure you don't use a method that doesn't exist!
- priority example:

```
if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.JELLY_BEAN) {

notificationBuilder.setPriority(Notification.PRIORITY_MAX);

}
```

Threads

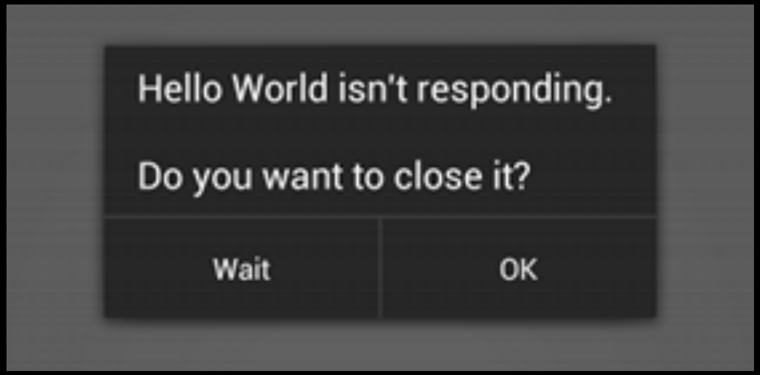


- By default, all applications are single-threaded
- You can create additional threads for any process but this isn't automatic
- The single spawned thread is called the UI thread
- This review will be schizophrenic!

UI Thread

- **Do:**
  - Lightweight setup in lifecycle methods (onCreate, etc)
  - Lightweight, non-blocking calculations
- **Don’t:**
  - Calculations for moves (games)
  - Database, network access (most I/O generally)
  - extreme in-memory calculations (bitmap calcs)

UI Thread & ANRs


Hello World isn't responding.

Do you want to close it?

- App not responding:

_ **generally triggered when app can't respond to user input**

_ **cause 1: unresponsive for 5 seconds**

_ **cause 2: broadcast receiver spends >10 seconds executing**

_ **about 100ms-200ms is the responsiveness threshold!**

Atomicity

- Guarantee of isolation from concurrent processes
- Succeed or Fail (change state or do nothing)
- Enforced by mutual exclusion
    - Example: update name + update age -> do both or do nothing~

Atomicity example: SharedPreferences

1. **commit()**: Synchronously commit changes **atomically**. Returns success/failure.

2. **apply()**: Asynchronously commit changes. Not atomic, returns void - no failure notification. writes:

    1. Synchronously to memory

    2. Asynchronously to “disk”, system handles in-flight writes

Discussion

- Which is preferable?
- What about testing?

Synchronization

- Synchronized statements and methods
    - We'll discuss methods
- Add the Synchronized keyword:

```
public synchronized void increment()
```

Properties of Synchronized Methods

- Two invocations on same object can’t interleave
- Other objects block until first thread is done
- When it exits, establishes a “happens before” relationship with subsequent invocations
- Constructors can’t be synchronized
  - why?

Without synchronization

- hello() // prints ***hello***, then prints ***how are you?***
- goodbye() // prints ***have a nice day***, then prints ***goodbye***.
- Without synchronization, you might get:

1) hello

2) have a nice day

3) how are you?

4) goodbye

DEADLOCK

- hello() // prints ***hello***, then prints ***how are you?***
- goodbye() // prints ***have a nice day***, then prints ***goodbye***.
- Without synchronization, you might get:

1) hello

2) have a nice day

3) how are you?

4) goodbye

DEADLOCK

https://docs.oracle.com/javase/tutorial/essential/concurrency/deadlock.html

Exercises

No order, complete 2.

- Create a notification with an API level gated feature
- Create a Builder class
- Challenge: Create a working example of deadlock
- Challenge: Create a notification that displays an image from a URL