Design Patterns

MSc in Computer Science

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit

Yamba

Xtend version Encapsulated as 3 Labs

- A - Enable Simple Tweet + timeline update on background thread
- B - Move background thread to an Android Service + restructure application to use Lambdas + Command pattern
- C - replace custom event mechanism with generic Broadcast Receivers

Lab

START

Objectives

- Develop an application in Android using the Xtend language
- Base the application on a twitter-like service hosted here:
  - http://yamba.marakana.com
- The structure of the application is derived from Learning Android

Lab

START

Objectives

- Introduce Services into the Xtend YambaX application
- Implement a background service to periodically update the twitter timeline

Lab

START

Objectives

- Incorporate Broadcast Event Senders/Receivers into the application
- Use BootReciever to recieve boot event to start application service on launch
- Use NetworkReveicer to stop/start service when network is starting/stopping
- Use Broadcast Events for status updates from background service to TimelineActivity

Yamba X

Status Update

hello sync!

hello sync!

Update

Preferences

Username
Please enter your username

Password
Please enter your password

API Root
URL of Root API for your service

Timeline

Marakana Student 05/06/2014 06:42:35
hello sync!

Marakana Student 05/06/2014 06:38:41
hello sync!

Marakana Student 05/06/2014 06:10:30
hi baby

Marakana Student 05/06/2014 05:58:57
aergaer

Marakana Student 05/06/2014 05:03:34

a Student 05/06/2014 05:03:40

a Student 05/06/2014 05:12:35

a Student 05/06/2014 05:16:33

a Student 05/06/2014 05:16:58

a Student 05/06/2014 05:19:00
ocks !

Status Update

Timeline

Preferences

Purge Data

Start Service

Revised to incorporte BaseActivity + 2 derived Activities

edeleastar authored 2 days ago

dc1a6072d6

Browse code

```
import android.os.Handler

class UpdaterService extends BackgroundService
{
  var YambaApplication app
  var TwitterAPI       twitter
  var Iterable<Status> newTweets
  // ...
  override def void doBackgroundTask()
  {
    try
    {
      val List<Twitter.Status> timeline  = twitter.getFriendsTimeline
      newTweets = if (app.timeline.size == 0) timeline else timeline.filter [it.id > app.timeline.get(0).id]

      Log.e("YAMBA", "number of new tweets= " + newTweets.size)
      val handler = new Handler(Looper.getMainLooper)

      handler.post ([| app.updateTimeline(newTweets)])
    }
    catch (TwitterException e)
    {
      Log.e("YAMBA", "Failed to connect to twitter service", e);
    }
  }
}
```

- UpdaterService running on Background Thread
- ‘Posts’ to application object on main thread updated tweets

Revised to incorporte BaseActivity + 2 derived Activities

edeleastar authored 2 days ago

dc1a6072d6

Browse code

```
class YambaApplication extends Application
{
  @Property TwitterAPI             twitter        = new TwitterAPI("student", "password",
                                                                  "http://yamba.marakana.com/api")
  @Property boolean                serviceRunning = false
  @Property List<Twitter.Status>   timeline       = new LinkedList<Status>
  @Property TimelineUpdateListener updateListener

  var prefsChanged = [ SharedPreferences prefs, String s |
                       twitter.changeAccount(prefs)  ]  as OnSharedPreferenceChangeListener

  override onCreate()
  {
    super.onCreate
    val prefs = PreferenceManager.getDefaultSharedPreferences(this)
    prefs.registerOnSharedPreferenceChangeListener = prefsChanged
  }

  override onTerminate()
  {
    super.onTerminate
  }

  def updateTimeline(Iterable<Twitter.Status> newTweets)
  {
    newTweets.forEach[timeline.add(0, it)]
    updateListener?.timelineUpdate
  }

  def clearTimeline()
  {
    timeline.clear
    updateListener?.timelineUpdate
  }
}
```

```
interface TimelineUpdateListener
{
  def void timelineUpdate()
}
```

- TimeLineUpdateListener defined to receive timeLineUpdates

- New tweets arrive here from UpdaterService

Revised to incorporte BaseActivity + 2 derived Activities

edeleastar authored 2 days ago

dc1a6072d6

Browse code

Note Lambda for TimelineUpdates

```
class TimelineActivity extends BaseActivity
{
  var TimelineAdapter timelineAdapter

  var timelineUpdate = [| timelineAdapter.notifyDataSetChanged ] as TimelineUpdateListener

  override onCreate(Bundle savedInstanceState)
  {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.timeline)
    timelineAdapter    = new TimelineAdapter(this, R.layout.row, app.timeline)
    app.updateListener = timelineUpdate
  }

  override onStart()
  {
    super.onStart
    val listTimeline = findViewById(R.id.listTimeline) as ListView
    listTimeline.setAdapter(timelineAdapter);
  }
}

class StatusAdapter
{
 ...
}

class TimelineAdapter extends ArrayAdapter<Twitter.Status>
{
  ...
}
```

Command Pattern

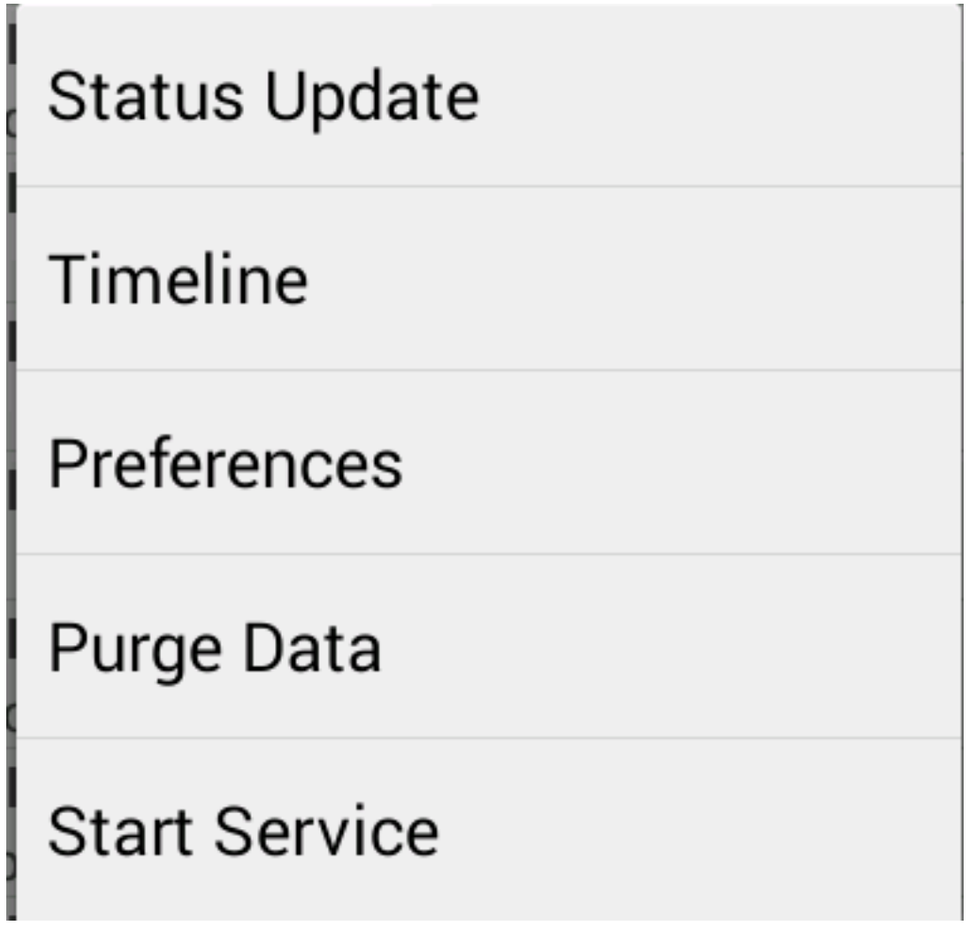
- Introduce Simple Command Interface for processing Menu Commands

- Each Command will be defined as a lambda

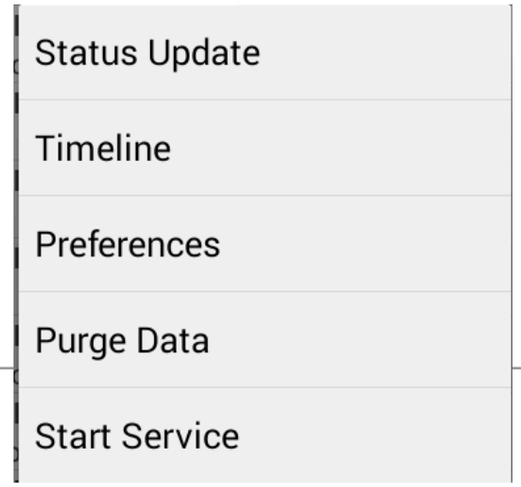
- ..And Dispatched based on id

```
interface Command
{
  def void doCommand()
}
```

```
override onOptionsItemSelected(MenuItem item)
{
  val command = commands.get(item.getItemId) as Command
  command.doCommand
  true
}
```



Command Lambdas




```
val prefs         = [ | startActivity(new Intent(this, typeof(PrefsActivity))
                                      .addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT)) ] as Command

val toggleService = [ | intent = new Intent(this, typeof(UpdaterService))
                        if (app.isServiceRunning)
                          stopService(intent)
                        else
                          startService(intent) ] as Command

val purge         = [ | app.clearTimeline
                        Toast.makeText(this, R.string.msgAllDataPurged, Toast.LENGTH_LONG).show() ] as Command

val timeline      = [ | startActivity(new Intent(this, typeof(TimelineActivity))
                                      .addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP)
                                      .addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT)) ] as Command

val status        = [ | startActivity(new Intent(this, typeof(StatusActivity))
                                      .addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT))] as Command
```

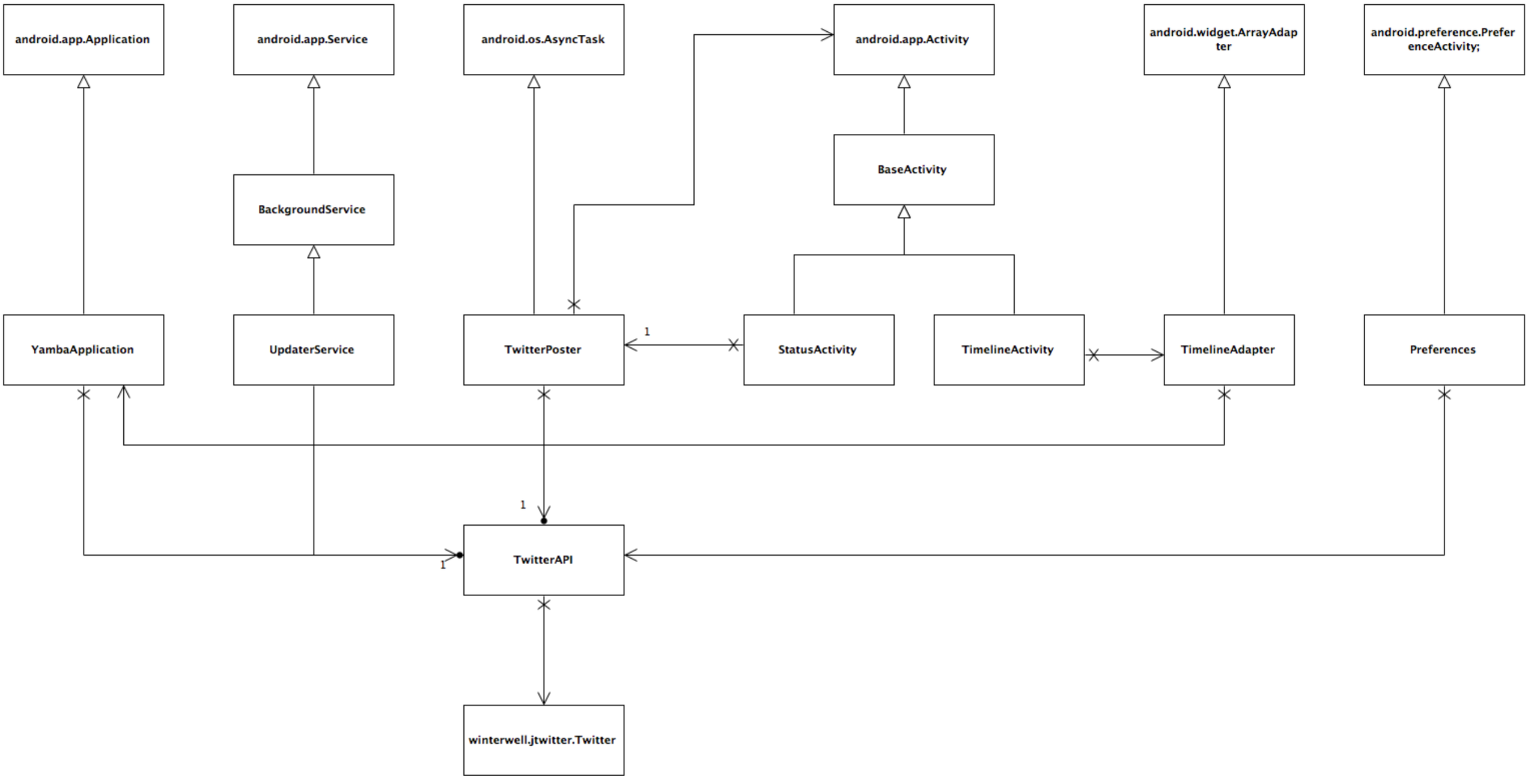
- Defined in BaseActivity class.
- All Activities in app derived from this class

Command Map

```
var commands      = #{ R.id.itemPrefs          -> prefs,
                       R.id.itemToggleService -> toggleService,
                       R.id.itemPurge         -> purge,
                       R.id.itemTimeline      -> timeline,
                       R.id.itemStatus        -> status  }
```

- Use Xtend literal syntax to create Command Map keyed on resource ID
- Menu Handler just look up map, and dispatches command
- Also implemented in BaseActivity

```
override onOptionsItemSelected(MenuItem item)
{
  val command = commands.get(item.getItemId) as Command
  command.doCommand
  true
}
```


android.app.Application
android.app.Service
android.os.AsyncTask
android.app.Activity
android.widget.ArrayAdapter
android.preference.PreferenceActivity;
BackgroundService
BaseActivity
YambaApplication
UpdaterService
TwitterPoster
1
StatusActivity
TimelineActivity
TimelineAdapter
Preferences
1
1
TwitterAPI
winterwell.jtwitter.Twitter