

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Head First Android Development : عنوان كتاب

مترجم : حسن زمانى

استفاده از این کتاب با ذکر سه صلوات رایگان هست :

اولی برای شفای همه مریض ها

دومی برای خوب شدن همه بیماران جسمی

و سومی هم برای شفای همه بیماران روحی

انتشار این کتاب فقط به صورت الکترونیکی و مجازی با ذکر منبع بلا مانع است.

**از کسانی هم که می خواهند در کتاب ادرس سایتی یا تبلیغی بزنن
می خواهیم که صرف نظر کنن چون خواندن مطلب خوب حق خوانندگان
شماست لطفا رعایت کنید.
برای تبلیغات صفحه بعد را خالی می گذارم و هر چی می خواهید تبلیغ
کنین اولیش هم خودم هستم.**

WWW.HZAVR.IR

HEADFIRST@HZAVR.IR

A Brain-Friendly Guide

Head First Android Development

فصل
٣ و ٤



Build your
UI from scratch



Test important
concepts directly with
your brain



Build your own
visualizations of
patterns and statistics



Build
interesting
patterns



Build
what you want
through

O'REILLY™

Jonathan Simon

مقدمه :

سلام به همه شما علاقه مندان برنامه نویسی اندروید . کتاب بیشرو , ترجمه فصل های سوم و چهارم کتاب **head first android development** است و امیدوارم که لذت ببرید . شما میتوانید ترجمه فصل های قبل را از [وب سایت نیکو درو میکرو](#) دانلود کنید . در ترجمه این کتاب مطمئنا نقص ها و کاستی هایی است و شما میتوانید پیشنهادات خود را برای بهتر شدن این کتاب در سایت مطرح کنید . اگر هم نکته مبهمی در کتاب بود و یا سوالی داشتید , باز میتوانید انرا در قسمت نظرات سایت مطرح کنید .

حسن زمانی – تابستان ۱۳۹۴

www.HZavr.ir

headfirst@hzavr.ir

کار کردن با فیدها

عکسهای از فضا!

صبر کن بزار تا حرفم را مستقیم بگم . مردم
فیدهای RSS را روی وب سائتهاشون قرار میدن
و من میتونم از اونهای توی برنامه ام استفاده
کنم
این طوری هر روز مثل جشن تولد من توی
اینترنت هست!



فیدهای RSS همه جا هستند! از اخبار هواشناسی و بازار سهام تا اخبار وبلاگها ، همه و همه روی فیدهای RSS منتشر میشن و منتظر هستند تا شما از اونهای توی برنامه خودتون استفاده کنید . در حقیقت ناشرهای این فیدها این را از شما می خوان! در این فصل ، شما یاد میگیرید که چطور توی برنامه خودتون با این فیدها تعامل داشته باشید . در طول راه شما مطالب بیشتری هم در مورد لایه های و دسترسی ها و اشکال زدایی یاد میگیرید!

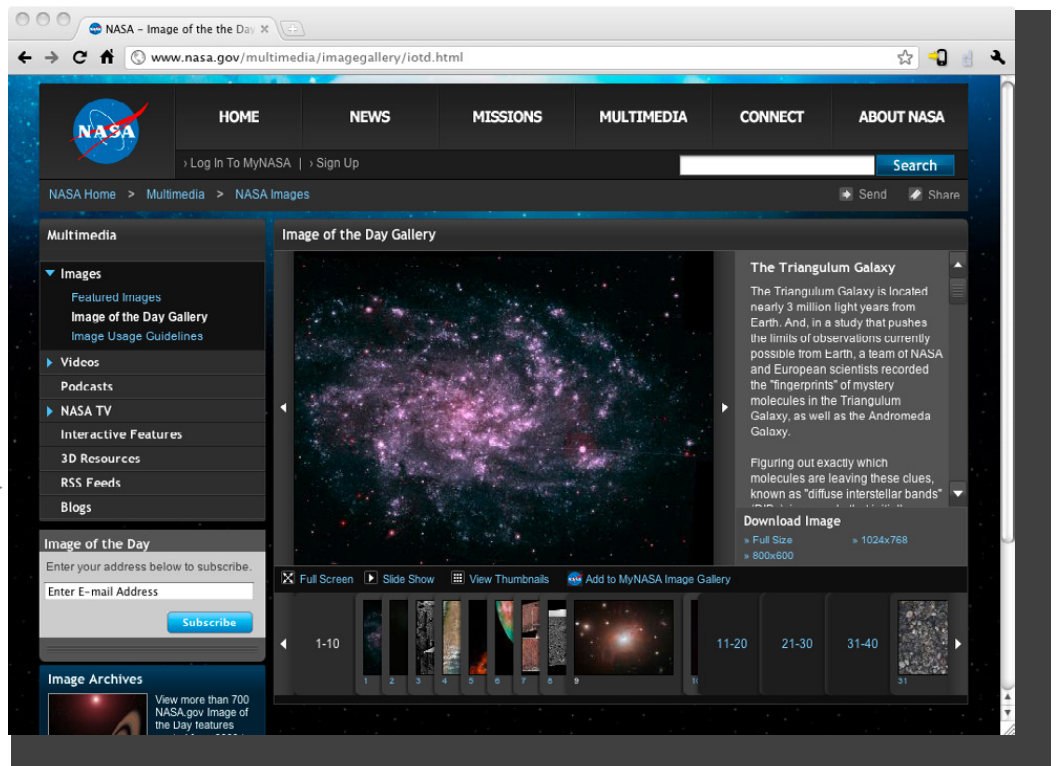
به ناسا خوش آمدید

من جدا به یک تلسکوپ نیاز دارم اما چیزی بیشتر
از این دوربین پیدا نکردم و به همین خاطره که من
نمی تونم فضا را ببینم . پس چاره ای جز این که
هر روز برم و سایت ناسا را برای عکسهای جدید
ببینم ، ندارم . توی این سایت هر روز عکسی از چیز
جدیدی میزارن !



Bobby

NASA's image of
the day site.



اما ایا این در مورد موبایلها هم درسته ؟

عکس روز این سایت روی کامپیوتر خوب به نظر می رسه اما روی موبایل زیاد جالب نیست . از نظر تنیکی کار میکنه اما باید کلی زوم کنی و صفحه را این ور و اون ور ببری تا این که بتونی عکس را ببینی . باید راه بهتری باشه



من توی سایت ناسا یک فید RSS را دیدم .
ایا تومیتونی برنامه ای بسازی که این فید را بخواند
و عکس را برای من نشون بده ؟ این راه
بهتره تا این که بری و کل وب سایت را توی موبایل
ببینی



بله ! ما میتونیم یک برنامه بنویسیم !

بزار تا از مهارتهای برنامه نویسیت توی اندروید استفاده کنیم و برنامه ای برای باب بسازیم تا اون بتونه عکسهای روز ناسا را روی موبایلش ببینه . مطمئنا اون از این برنامه خوشش میاد !

طرح کلی برنامه ات را بساز

قبل از این که بریم سراغ برنامه جدیدت بیا تا طرح کلی اون را بکشیم .

چون تو می خواهی برنامه ات را از روی فید عکس ناسا بسازی ، یک نگاه به این فید بنداز تا بفهمی با چه چیزی
قراره کار کنی .

این فید یک فید RSS(Really Simple Syndication) هست . تو میتونی با یک جستوجوی ساده در
اینترنت اطلاعات بیشتری راجع به اون کسب کنی اما در حال حاضر تو به اونها فقط به عنوان فایل های XML
نگاه کن .

Eclipse خودش ویرایشگری داره که به تو کمک میکنه این فید ها را نمایش بدی . به ادرس

<http://www.nasa.gov/rss/imageoftheday.rss>

و محتویات اون را روی کامپیوترت ذخیره کن . و تو به راحتی میتونی اون را در Eclipse باز کنی و ببینی !

فید عکس روز که روی کامپیوتر ذخیره شده و

در ویرایشگر باز شده .



اطلاعات هدر RSS

اطلاعات کلی راجع به فید

اطلاعات مربوط به عکس روز ناسا

منا دیتا های راجع به عکس

?? xml	version="1.0" encoding="UTF-8"
?? xml-stYLESHEET	href="/externalflash/NASA_Detail/NASA_Detail.xsl" type="text/xsl"
rss	
version	2.0
channel	
title	NASA Image of the Day
link	http://www.nasa.gov/multimedia/imagegallery/index.html
description	The latest NASA "Image of the Day" image.
language	en-us
docs	http://blogs.law.harvard.edu/tech/rss
managingEditor	yvette.smith-1@nasa.gov
webMaster	brian.dunbar@nasa.gov
image	
xmlns:java_code	xalan://gov.nasa.build.Utils1
url	http://www.nasa.gov/images/content/507664main_image_1830_516-387.jpg
title	Decorating the Sky
link	http://www.nasa.gov/multimedia/imagegallery/index.html
description	
item	
xmlns:java_code	xalan://gov.nasa.build.Utils1
title	Decorating the Sky
link	http://www.nasa.gov/multimedia/imagegallery/image_feature_1830.html
description	This mosaic image taken by NASA's Wide-field Infrared Survey Explorer, or WISE, fe
guid	
pubDate	Mon, 27 Dec 2010 00:00:00 EST
enclosure	



Sharpen your pencil

کلی چیز داخل این فید هست ! اگه همه اونها را بخوای به کاربرهات نشون بدی ،هم اون و هم تو از هدف اصلی خودت که همون نشون دادن عکس روز هست فاصله میگیری . گرچه نشون دادن عکس خالی بدون توضیحات هم میتونه خسته کننده باشه .

به کدهای این فید که به صورت XML هستند یک نگاهی بنداز و چند تا چیزی که فکر میکنی بدرد بخور هستند را انتخاب کن و مطمئن شو که برای ان کارت دلیلی هم داری . اولی را خودم نوشتم . جندتا چیز دیگه هم تو اضافه کن .

Property to include

Why include it?

ادرس عکس

من می خوام که عکس را نشون بدم .پس ادرس اون

هم شاملش میشه .

و میگم این عکس روزه !

.....

.....

.....

.....

.....

.....

.....

.....

.....



Sharpen your pencil Solution

There's a whole bunch of stuff in that feed! If you show it all, you're going to overload your users with information and miss the point of building a specialized mobile app for viewing the image of the day. At the same time, just showing the image would be pretty boring.

You were to look at the XML view of the feed, pick a few things you think you should show, and make say why you picked it.

Property to include Why include it?

image URL

I definitely want display the image, so I'll include the image URL. This is an image of the day app, after all!

فایل XML خودش حاوی عکس مورد نظر نیست بلکه فقط آدرس اون را داره بنابراین تو عکس را دانلود میکنی و اون را نشون میدی.

عنوان عکس

The image title will help users quickly tell what the image is about.

You'll need to make sure you get the correct title and description, because the example feed contains many of each. In the example feed, the image description is blank, but the item description is populated correctly.

item description

توضیح عکس یز باحالی باشه کاربر میخواد که چیز بیشتری راجع به اون بدونه . اینها این عکس راجع به چیه اطلاعات مهمی نیستند اما بهتره که بدونه . این عنوان به کاربر کمک میکنه که سریعاً بفهمه

زمان انتشار عکس

ناسا هر روز عکس جدید منتشر نمیکنه (برای مثال اخر هفته ها) بنابراین تاریخ انتشار کمک می کنه که بدونیم این عکسی که نشون داده شده کی منتشر شده.

ممکنه تو چیزهای دیگه ای انتخاب کرده باشی یا

چیزهای بیشتری اضافه کرده باشی (که البته هیچ

اشکالی نداره) اما ما در اینجا فقط از همین

گزینه ها استفاده میکنیم با وجود این که خودت

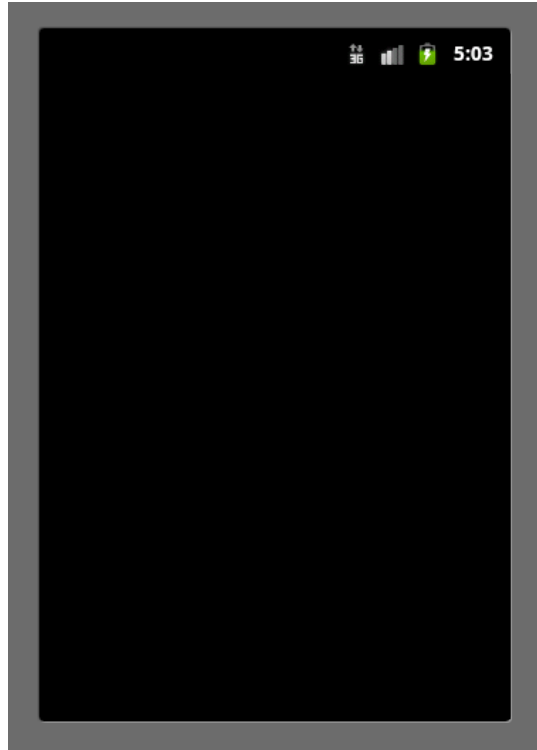
میتونی برنامه ات را به هر نحوی که خواستی

بسازی .



Screen Design Magnets

برای ساخت رابط کاربری از قطعاتی که این زیر هستند استفاده کن . برای هر چیزی که از فید RSS گرفتی یک View وجود داره .



Viewها را روی صفحه بچین



عنوان عکس در یک TextView



Decorating the Sky

تاریخ انتشار در یک TextView



Mon, 27 Dec 2010 00:00:00 EST

توضیح عکس باز هم در یک TextView



This mosaic image taken by NASA's Wide-field Infrared Survey Explorer, or WISE, features nebulae that are part of the giant Orion Molecular Cloud--the Flame nebula, the Horsehead nebula and NGC 2023. Despite its name, there is no fire roaring in the Flame nebula. What makes this nebula shine is the bright blue star seen to the right of the cen

عکس مورد نظر ، نه دیگه این یکی TextView نیست بلکه بهش میگوین Image View (نگران نباش به زودی در مورد این هم یاد میگیری)





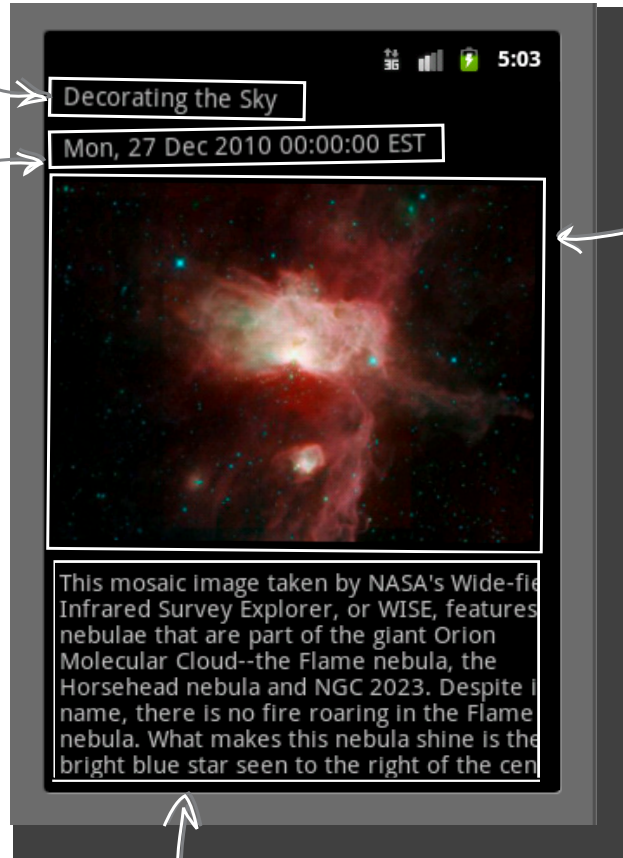
Screen Design Magnet Solution

برای ساخت رابط کاربری از قطعاتی که این زیر هستند استفاده کن . برای هر چیزی که از فید RSS گرفتی یک

View وجود داره .

عنوان در بالا قرار میگیره بنابراین تو میدونی به چی نگاه می کنی

تاریخ میتونه هر جایی باشه ولی بهتره که زیر عنوان باشه ، مگه نه؟



عکس هم وسط قرار میگیره و تا اندازه صفحه کش میاد.

بهتره توضیح باشه اما وجودش واجب نیست .

همچنین اون خیلی بزرگه ، پس بهتره این زیر

بزاریم تا توی دست و پا نباشه .

من امتحان ریاضی دارم . پس بهتره عجله کنم .
اما فعلا این طرح به نظرم خوبه . نمیتونم صبر کنم ،
دوست دارم ببینم چطور کار میکنه !



وقت کد نویسیه !

هر برنامه خوبی اولش با یک طرح خوب شروع می شه که ما الان اون را داریم . پس حالا وقتش رسیده تا شروع به برنامه نویسی بکنیم .

اینها کارهاییه که باید انجام بدی .

1 یک پروژه جدید ایجاد کن

تو داری یک برنامه جدید میسازی پس یک پروژه جدید ایجاد کن .

2 اطلاعات فید را به صورت محلی ذخیره کن

حذف کردن چیزهای متغیر و نا ثابت در طول برنامه نویسی فکر خوبیه بنابراین فید را روی کامپیوتر ذخیره کن ، با این کار تو میتونی روی برنامه نویسی تمرکز کنی به ارتباط برقرار کردن با فید .

3 حالا واسط کاربری و محتوای صفحه را با استفاده از فید بساز

تو یک طرح برای واسط کاربری داری حالا وقتشه اون طرح را اجرا کنی .

یک لایه بساز و برنامه ات را بساز و اجراش کن !

4 برنامه ات را به فید RSS که به فرم XML است وصل کن

وقتی که برنامه ات را ساختی . فقط کافیست که اون را به فیدهای XML وصل کنی تا برنامه ات زنده بشه .

این کارها خیلی آسون هستن ، قول میدم !

حالا که آماده کد نویسی شدی یک پروژه جدید بساز . با استفاده از مسیر زیر میتونی این کار را بکنی .

File-New – android Project

The project name can have spaces or not. But It's better leave out spaces, because a directory is created with the project name in your workspace, and command-line navigation is usually easier without spaces.

New Android Project
Creates a new Android Project resource.

Project name:

Contents

- Create new project in workspace
- Create project from existing source
- Use default location

Location:

- Create project from existing sample

Samples:

Build Target

Target Name	Vendor	Platform	A
<input type="checkbox"/> Android 1.5	Android Open Source Project	1.5	3
<input type="checkbox"/> Android 1.6	Android Open Source Project	1.6	4
<input type="checkbox"/> Android 2.1-update1	Android Open Source Project	2.1-update 7	
<input type="checkbox"/> Android 2.2	Android Open Source Project	2.2	8
<input checked="" type="checkbox"/> Android 2.3	Android Open Source Project	2.3	9

Standard Android platform 2.3

Properties

Application name:

Package name:

Create Activity:

Min SDK Version:

? < Back Next > Cancel Finish

Select the latest platform you have installed (2.3 at the time of this writing).

The application name has spaces. This is shown to your users, so format it to be human readable.

Web site plus application name is a pretty safe bet for a package name.

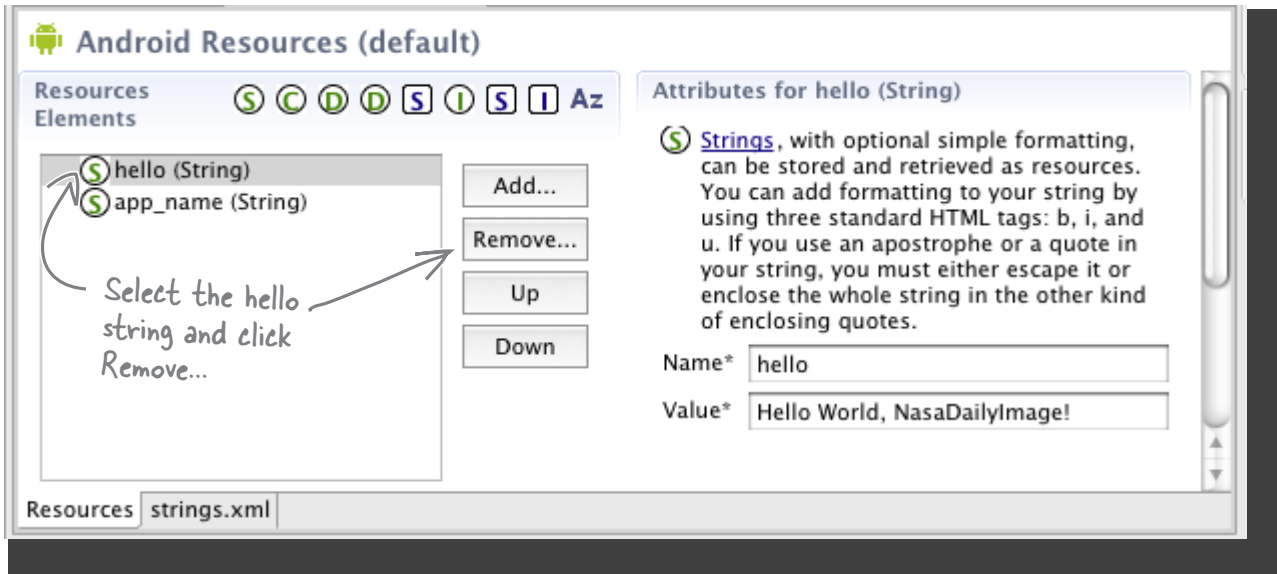
Make a default activity. Naming the activity to match the project name is a good rule for single-screen apps.

از شر چیزهای که به طور خودکار تولید شدن خلاص شو مثل 'hello'

تو به متن "hello world nasadailyimage" که به صورت خودکار ایجاد شده نیازی نداری ، پس قبل این که ادامه بدیم اون رشته و TextView را حذف کن .

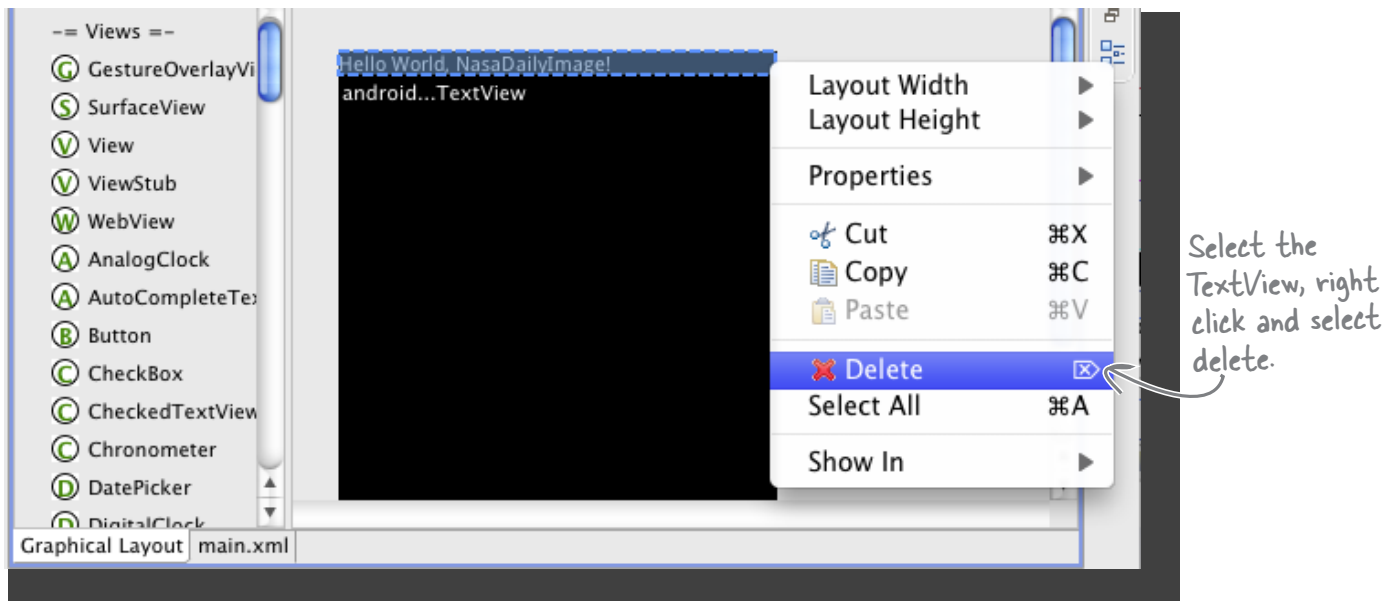
فایل string.xml را باز کن و hello string را حذف کن .

1



فایل main.xml را باز کن و TextView را حذف کن .

2



برنامه ات را ذخیره کن . حالا تو یک برنامه ترو تمیز داری .

3

کارت را با ذخیره کردن متن ها در منبع رشته ای آغاز کن . string.xml را باز کن و سه رشته ی جدید ، یکی برای عنوان عکس و دوتای دیگه هم به ترتیب برای تاریخ و توضیحات ایجاد کن . ساده ترین کار اینه که این متن ها را مستقیما از داخل فید کپی کنی .

فایل strings.xml و اطلاعاتی که به اون اضافه

کردیم

```
<resources>
  <string name="app_name">NASA Daily Image</string><string
  name="test-image_title">
    Decorating the Sky</string>
  <string name="test-image_date">
    Mon, 27 Dec 2010 00:00:00 EST </string>
  <string name="test-image_description">
    This mosaic image taken by NASA\'s Wide-field Infrared Survey
    Explorer, or WISE features
    three nebulae that are part of the giant Orion Molecular
    Cloud--the Flame nebula, the Horsehead nebula and NGC 2023.
    Despite its name, there is
```

مراقب باش اینجا ها باید کاراکتر \ را وارد کنی

مقادیری که از فایل RSS بدست اومده



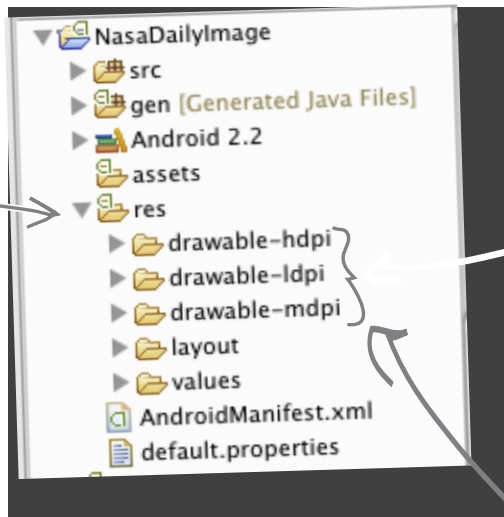
Watch it!

Watch out for escape characters

بعضی از کاراکترها در فایل XML باید نادیده گرفته بشن مثل \ " ' . برای اینکه جاوا بدونه که اینها کارکترهای کنترلی نیستن و تو این کار را فقط با اضافه کردن \ به قبل از این کاراکترها میتونی انجام بدی .

عکس را در پروژه ات ذخیره کن

عکسها در پروژه تو به عنوان منبع در محل rss ذخیره میشن. ایا میتونی فولدری را که اسمش drawable هست را در محل res پیدا کنی؟



ممممممم..... اینجا سه تا پوشه به اسم drawable هست

در محل res سه تا پوشه به اسم drawable هست . حالا چی؟

اوه درسته ، پوشه ها برای صفحات با سایزهای مختلف هستند .

یکی از چیزهای شگفت انگیز در مورد اندروید اینه که روی دستگاه های مختلفی اجرا میشهو فکرش را بکن که چندتا دستگاه هست که برنامه ات میتونه روی اونها اجرا بشه ! پس بهای این انعطاف پذیری و گستردگی اینه که برنامه ات از تمام دستگاه های با سایزهای متفاوت پشتیبانی بکنه .

تو در مورد دستگاهها با سایز صفحات مختلف در آینده بیشتر یاد میگیری . اما فعلا عکس را به پوشه drawable-hdpi اضافه کن . شبیه ساز به طور پیش فرض از این دارکتوری استفاده میکنه



مرورگر را باز کن و به ادرس عکسی که در RSS هست برو و اون را در پروژه ات و در محل res/drawable-hdpi ذخیره کن و اسم اون را test_image.jpeg بزار.

خپ حالا که داده هات را به صورت محلی ذخیره کردی ، اجازه بده بریم سراغ ساختن لایه !





Exercise

در زیر قطعات لایه xml را میبینی . هر کدام از این قطعات را روی لایه ای که صفحه بعد هست بکش و لایه برنامه ات را کامل کن .

View visuals here, just for reference.



Image

View XML declaration magnets

```
<ImageView
    android:id="@+id/imageDisplay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/test_image"/>
```

Decorating the Sky

Title

```
<TextView
    android:id="@+id/imageTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/test_image_title"/>
```

This mosaic image taken by NASA's Wide-field Infrared Survey Explorer, or WISE, features three nebulae that are part of the giant Orion Molecular Cloud--the Flame nebula, the Horsehead nebula and NGC 2023. Despite its name, there is no fire roaring in the Flame nebula. What makes this nebula shine is the bright blue star seen to the right of the central

Description

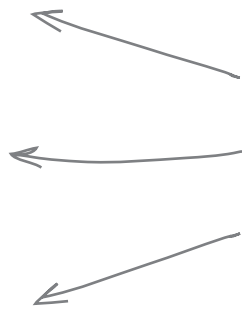
```
<TextView
    android:id="@+id/imageDescription"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/test_image_description"/>
```

Mon, 27 Dec 2010 00:00:00 EST

Date

```
<TextView
    android:id="@+id/imageDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/test_image_date"/>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
```



قطعات را کنار هم بزار تا لایه کامل بشه . تو از **LinearLayout** استفاده می کنی بنابراین
کافیه که اونها را از بالا به پایین بچینی .

```
</LinearLayout>
```



Exercise Solution

در زیر کدهای هر کدام از View ها را میبینی که اونها را چیدی و لایه برنامه ات را کامل کردی .

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
```

```
<TextView
    android:id="@+id/imageTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/test_image_title"/>
```

Title

```
<TextView
    android:id="@+id/imageDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:text="@string/test_image_date"/>
```

Date

```
<ImageView
    android:id="@+id/imageDisplay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/test_image"/>
```

Image

```
<TextView
    android:id="@+id/imageDescription"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/test_image_description"/>
```

Description

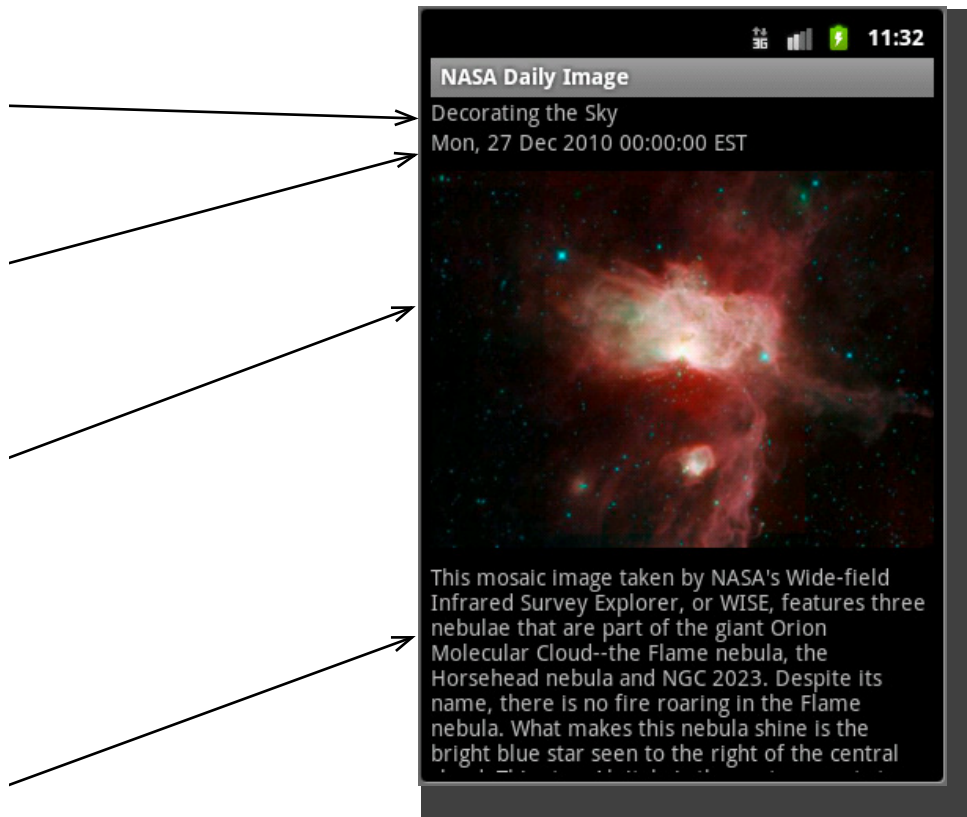
```
</LinearLayout>
```



TEST DRIVE

برنامه ات را اجرا کن ، برای این کار تو باید پروژه خودت را از پنجره explorer انتخاب کنی و run را بزنی .

اینجا پنجره ای تحت عنوان run as android application باز می شه که انتخاب کن .



عالیه ! صفحه به نظر خوب میاد .

این صفحه با اون چیزی که طراحی کرده بودی مطابقت میکنه . کارت عالی بود .

هی توضیحات نشون داده شدن اما قسمتی از
اون بریده شده . فکر کنم بهتره صفحه بیاد
پایین یا همون اسکرول بشه .

راست می گی ، اسکرول کردن فکر خوبیه !

تو هرگز نمیتونی بفهمی که ناسا چقدر توضیح میزاره ، اصلا ممکنه یک کتاب را به عنوان
توضیح بزاره ! کاری که ما میتونیم انجام بدیم برنامه خودمون را منعطف بسازیم
یک راه خوبی که هست اینه که تمام صفحه را اسکرول کنیم .



ای کاش میشد کل لایه را داخل چیزی مثل view دیگه قرار داد و اون خودش اسکروول را به طور خودکار انجام بده ، اما من میدونم که این رویایی بیش نیست



از ScrollView استفاده کن تا محتوای بیشتری را نشون بده

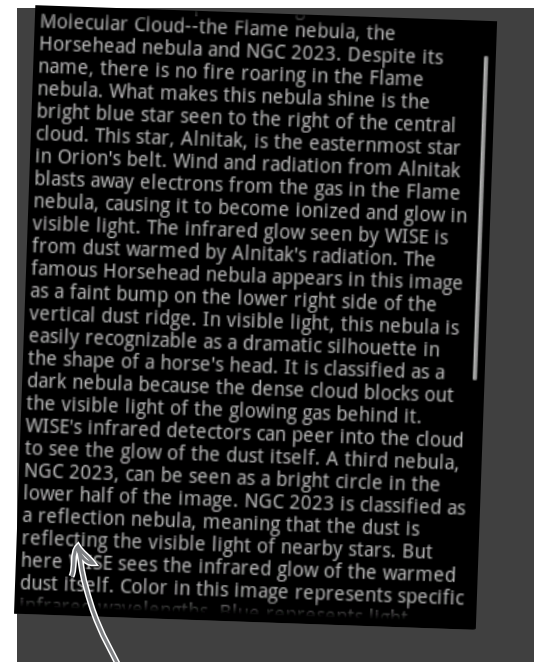
ScrollView هم مثل TextView یک View هست که با استفاده از اون میتونی محتوای بیشتری را نشون بدی. ScrollView یک ViewGroup هست (این نام، اسمی هست که اندروید به مدیریت کننده لایه نسبت داده). برای استفاده از ScrollView قطعات را به اون اضافه کن و اون به طور خودکار عمل اسکرول را انجام میده.

و اینجاست که TextView اسکرول میشه

TextView درون ScrollView پیچیده شده

```
<ScrollView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TextView
        android:text="@string/image_description"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
</ScrollView>
```

TextView اونقدر بزرگ هست که بتونه صفحه را پر کنه



LOTS of scrolling text!

چقدر باید اسکرول بشه ؟

تو میتونی یک یا چند View را در ScrollView قرار بدی یا بیچی. هر View را که در ScrollView قرار بدی اسکرول میشه. چون که هدف اینست که جنبه بصری برنامه را مقیاس پذیر بکنی پس کل لایه را اسکرول کن. با این کار مقیاس پذیری رابط کاربریت تضمین شده هست حتی اگه اطلاعاتی که از فید میان خیلی زیاد باشن.

چیزی که در استفاده از ScrollView باید به اون دقت کرد اینست فقط میتونه یک View داشته باشه.

در مثال این صفحه ما TextView را مستقیماً به ScrollView اضافه کردیم. اما اگه بخواهیم این کار را با کل View های صفحه انجام بدیم، راه حل اینست که کل View ها را به یک LinearLayout اضافه کنیم که یک View حساب بشه بعد اون را به ScrollView اضافه کنیم.

Sharpen your pencil

کد زیر را که اصلاح شده را اضافه کن تا کل صفحه اسکرول بشه. تو باید `ScrollView` را به عنوان پایه اصلی خودت قرار بدی. و چون این لایه یک `View` بیشتر نمیتونه داشته باشه باید کل `LinearLayout` را به اون اضافه کنی.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
```

نکته: این قسمت باید در لایه اصلی باشه پس اگه میخوای ان لایه را به `ScrollView` اضافه کنی باید این خط کد را به `ScrollView` اضافه کنی.

```
<TextView
    android:id="@+id/imageTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/test_image_title"/><TextView
    android:id="@+id/imageDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/test_image_date"/><ImageView
    android:id="@+id/imageDisplay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/test_image"/>
<TextView
    android:id="@+id/imageDescription"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/test_image_description"/>
```

Wrap this
entire
layout in a
ScrollView.

```
</LinearLayout>
```




Sharpen your pencil Solution

You were to use the `ScrollView` to make the entire screen scroll. You needed to make the `ScrollView` the main layout. And since the `ScrollView` can hold only one `View`, you should have added the entire `LinearLayout` as the one `ScrollView` child `View`.

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
```



یادت که نرفته ، باید این قسمت را به `ScrollView` منتقل کنی (به لایه اصلی یا ریشه)

Beginning
of the
ScrollView



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
```

The complete
non-scrolling
layout

```
    <TextView
        android:id="@+id/imageTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/test_image_title"/>
    <TextView
        android:id="@+id/imageDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/test_image_date"/>
    <ImageView
        android:id="@+id/imageDisplay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/test_image"/>
    <TextView
        android:id="@+id/imageDescription"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/test_image_description"/>
</LinearLayout>
```

The inner widgets remain
untouched inside the
`LinearLayout`.

```
</ScrollView>
```



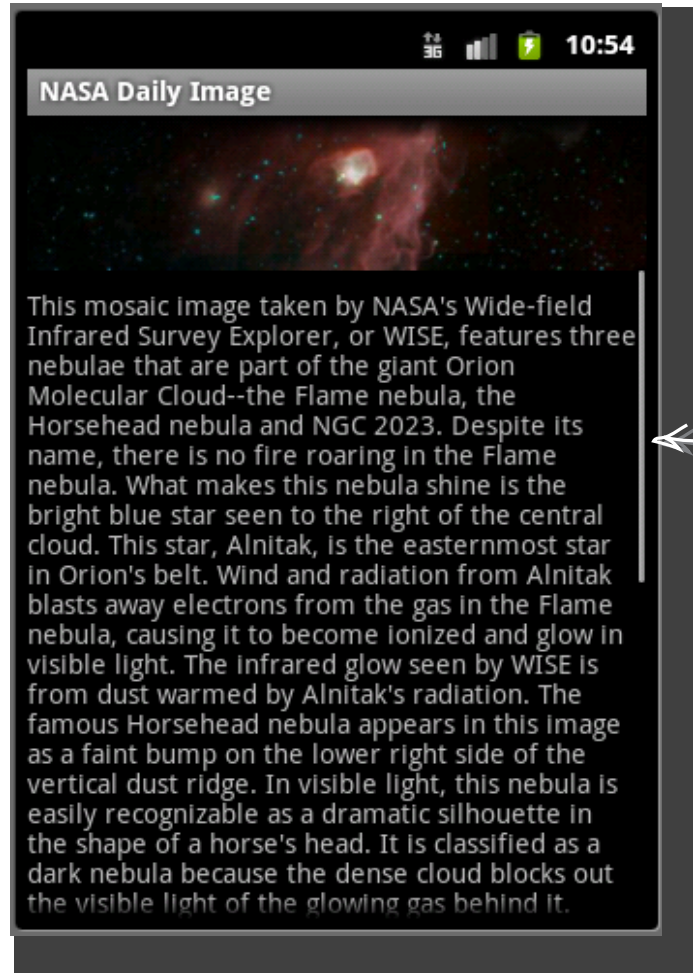
end of the
ScrollView



Test Drive

برنامه را اجرا کن تا اسکرولی که اضافه کردی را امتحان کنی . الان باید ببینی که کل صفحه اسکرول میشه .

نگاه کن که چطور کل صفحه اسکرول میشه .



این اسکرول بار است

همون طور که انتظارش را داشتیم همه چیز داره اسکرول میشه .

عمل اسکرول به خوبی داره انجام میشه . ببین که چطور کل صفحه بالا و پایین میره ، این به خاطر اینه که تو کل LinearLayout را به ScrollView اضافه کردی .

بیا تا اون را به بابتی نشون بدیم و ببینم نظر اون چیه !

وووو . خیلی قشنگ به نظر میرسه !
اما یک مسله هست و اون اینه که هنوز
از فیدهای RSS واقعی ناسا استفاده نکردیم .



اوپسسس(مترجم : همون((ای وای)) خودمون) ! تقریبا داشت یادم میرفت .

همه چیز در مورد لایه و طراحی درسته و صفحه اون طوری که می خواستی شد . اما حالا وقتشه که اون طور که
میخواهی کار کنه یعنی اطلاعات فید را به صورت انی و در لحظه نشون بده یا به عبارتی اونها را تجزیه کنه .

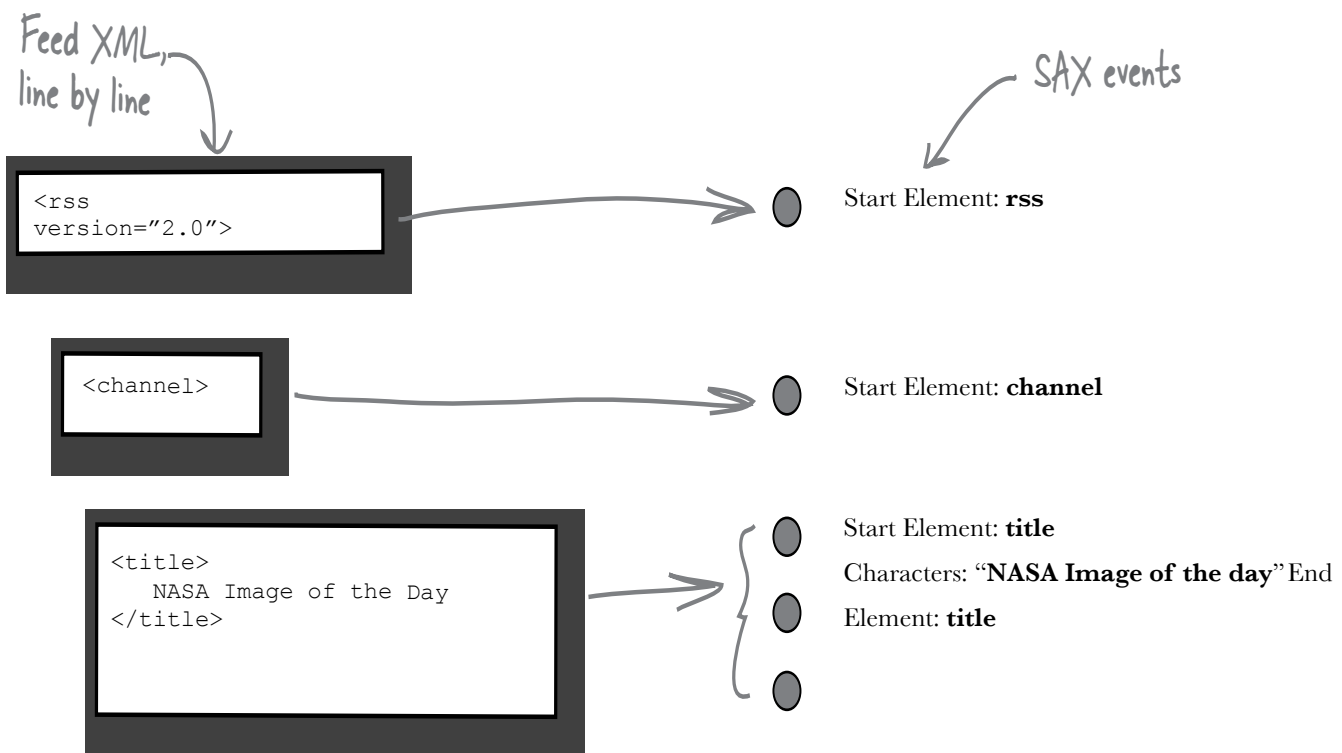
یک تجزیه کننده انتخاب کن

اون بیرون کلی تجزیه کننده فایل های XML هست و اندروید سه تا از اونها را پشتیبانی میکنه
DOM , SAX , XMLPULL
هر کدوم از اونها از روشهای مخصوصی برای تجزیه فایل های XML استفاده میکنن که مضایا و معایبی دارن .
بزار برای راحتی از **SAX** استفاده کنیم .

SAX Parsing Up Close



این تجزیه گر به صورت تصادفی کار نمیکنه و از اول شروع میکنه و پیامها و رویدادهای مخصوصی را صادر میکنه و خارج میشه . اینجا یک مثال ساده آوردیم که در اون این تجزیه گر در سه خط اول فید ناسا چند رویداد و پیام هایی صادر میکنه .



تجزیه گر فید ناسا فقط کافیه تا به پیامهای قطعات آغازین SAX گوش کنه و هر کدوم را متناسب با نیاز (عنوان ، ادرس عکس ، تاریخ و.....) بگیره و از اونها استفاده کنه . همه اش همین بود ! (مترجم : اره جون خودت ، شما را نمی دونم ولی من که نفهمیدم چی گفت و چی نوشتم ، زیاد هم نگران نباشین جون در ادامه از کدهای آماده تحت عنوان Ready Bake استفاده میکنیم ، (مترجم : پس فکر کنم خود نویسنده هم نفهمیده چی گفته) (این اخری را شوخی کردم))



Ready Bake Cone

تجزیه گرهای SAX خیلی به هم شبیه هستند . حالا تو به طور مفهومی یاد میگیری که این تجزیه گر چطور

کار میکنه ، این جا کدهای آماده تجزیه گر را میبینی که از قبل آماده شده و تو میتونی اون را داخل برنامه ات

قرار بدی . نگران نباش ، همه چیز را که نباید بفهمی و اگر تو داخل قطعه ای هستی که مورد نظرت هست ، کاراکترهای اون را دریافت کن

هر چقدر میخوای درون اون کنکاش و جستجو کنی .

```
private String url = "http://www.nasa.gov/rss/image_of_the_day.rss";
private boolean inUrl = false;
private boolean inTitle = false;
private boolean inDescription = false;
private boolean inItem = false;
private boolean inDate = false;
private Bitmap image = null;
private String title = null;
private StringBuffer description = new StringBuffer();
private String date = null;
```

به خاطر این که هر قطعه مثل عکس و ادرس و عنوان و به طور جدا فراخوانی میشن اینها؛
به خاطر این هستند که بفهمی داخل کدوم قطعه هستی

```
public void processFeed() {
```

```
    try {
```

```
        SAXParserFactory factory =
```

```
            SAXParserFactory.newInstance();
```

```
        SAXParser parser = factory.newSAXParser();
```

```
        XMLReader reader = parser.getXMLReader();
```

```
        reader.setContentHandler(this);
```

```
        InputStream inputStream = new URL(url).openStream();
```

```
        reader.parse(new InputSource(inputStream));
```

```
    } catch (Exception e) {
```

یک رشته ورودی از ادرس اینترنتی فید میسازه.

Configur
reader and parser.

```
        private Bitmap getBitmap(String url) {
```

```
            try {
```

```
                HttpURLConnection connection =
```

```
                    (HttpURLConnection)new URL(url).openConnection();
```

```
                connection.setDoInput(true);
```

```
                connection.connect();
```

```
                InputStream input = connection.getInputStream();
```

```
                Bitmap bitmap = BitmapFactory.decodeStream(input);
```

```
                input.close();
```

```
                return bitmap;
```

```
            } catch (IOException ioe) { return null; }
```

```
        }
```

شروع تجزیه !

```

public void startElement(String uri, String localName, String qName,
    Attributes attributes) throws SAXException {
    if (localName.equals("url")) { inUrl = true; }
    else { inUrl = false; }

    if (localName.startsWith("item")) { inItem = true; }
    else if (inItem) {
        if (localName.equals("title")) { inTitle = true; }
        else { inTitle = false; }

        if (localName.equals("description")) { inDescription = true; }
        else {
            inDescription = false; }

        if (localName.equals("pubDate")) { inDate = true; }
        else { inDate = false; }
    }
}

```

inDescription = false; }

```

public void characters(char ch[], int start, int length) {
    String chars = new String(ch).substring(start, start + length);
    if (inUrl && url == null) { image = getBitmap(chars); }
    if (inTitle && title == null) { title = chars; }
    if (inDescription) { description.append(chars); }
    if (inDate && date == null) { date = chars; }
}

```

.... and if you're in an element that you are interested in, cache the charac

این جا هم تعدادی متد یا تابع دستیابی هست که میتونی متغیرهای مورد نظرت را از

```

public String getImage() { return image; }
public String getTitle() { return title; }
public StringBuffer getDescription() { return description; }
public String getDate() { return date; }

```



Do this!



تجدید گرافیک سایت را از سایت Head First Android Development دریافت کن و به پروژه ات اضافه کن .

حالا که تو تجزیه گرفید را به پروژه ات اضافه کردی و قتشه تا از اون در اکتیویته خودت استفاده کنی . این کار را با نمونه سازی از handler در متد onCreate شروع کن .

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    IotdHandler handler = new IotdHandler();  
  
    handler.processFeed();
```

Handler را نمونه سازی کن

..... و شروع به تجزیه کن .

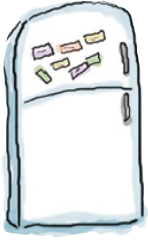
برنامه هنوز کار نمیکنه . تو فید را تجزیه میکنی

اما از مقادیر اون در Viewها استفاده نمی کنی .



درسته ، تو مقادیر را دریافت میکنی اما از اونها را نشون نمیدی .

پس بزار تا متد یا تابعی به اسم `resetDisplay()` بسازیم که همه اطلاعات را روی صفحه قرار میده . بعد تو میتونی این متد را داخل `onCreate ()` و بعد از `processFeed()` فراخوانی کنی .



Code Magnets

با استفاده از قطعات داده شده متد `resetDisplay()` را کامل کن ، اول `View` های صفحه را با استفاده از `findViewById()` به فیلدهای برنامه جاوا متصل کن و بعد مقادیر دریافتی از تجزیه گر را دراونها قرار بده .

```
private void resetDisplay(String title, String date, String  
imageUrl, String description) {
```

Get a reference to each on
screen View. Then set the
values on those Views to the
cached values from the parser.



Here are your
magnets.



```
(TextView) findViewById(R.id.imageDate);
```

```
(TextView) findViewById(R.id.imageTitle);
```

```
titleView.setText(title);
```

```
TextView descriptionView =
```

```
(ImageView) findViewById(R.id.imageDisplay);
```

```
(TextView) findViewById(R.id.imageDescription);
```

```
dateView.setText(date);
```

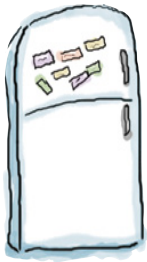
```
ImageView imageView =
```

```
imageView.setImageBitmap(image);
```

```
descriptionView.setText(description);
```

```
TextView dateView =
```

```
TextView titleView =
```

Code Magnets Solution

You were to complete the `resetDisplay()` method below by retrieving references to the on screen Views (using `findViewById`) and setting the values on those Views with the values passed in. With this method complete, you can use it to pass in the values from the feed.

```
private void resetDisplay(String title, String date, String  
imageUrl, String description) {
```

```
TextView titleView =
```

```
titleView.setText(title);
```

به title وصل شو و متنی را که از handler یا همون تجزیه گر گرفتی به اون نسبت بده .

```
TextView dateView =
```

```
(TextView) findViewById(R.id.imageDate);
```

```
dateView.setText(date);
```

Same deal with date View: get the View reference and set the text to the value from the parser.

Get a reference to the ImageView.

```
ImageView imageView =
```

```
imageView.setImageBitmap(image);
```

Use the image from the feed parser and set it on the ImageView.

```
TextView descriptionView =
```

```
(TextView) findViewById(R.id.imageDescription);
```

```
descriptionView.setText(description);
```

Finish up by getting the description View reference and setting the text with the cached description value.

```
}
```

حالا میتونی کار وصل کردن handler را در متد onCreate() تموم کنی . متد resetDisplay() را بعد از Handler.processFeed() قرار بده . با این کار مقادیری که تجزیه گر از فید بدست آورده را در View ها قرار میدیم .

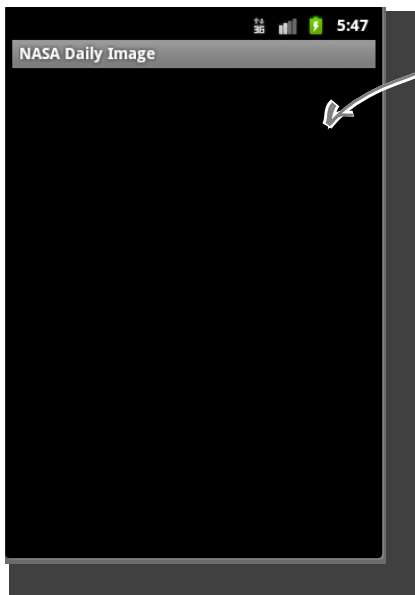
```
resetDisplay(iotdHandler.getTitle() , iotdHandler.getDate() ,  
            iotdHandler.getImage() , iotdHandler.getDescription());
```

The resetDisplay method is a helper method you're about to write to populate the fields on screen with the parsed data.



— Test Drive

همه چیز به تجزیه گر وصل شد . اکتیویتی و تجزیه گر به هم وصل شدن و نتایج حاصل از تجزیه روی صفحه نشان داده میشه . حالا برنامه را اجرا کن .



مممم..... ، صفحه خالی

اوه ! هیچی روی صفحه نیست !



واضح هست که نقصی توی کار بوده ، اما چه نقصی ؟ کجا باید دنبال این نقص بگردیم ؟

با استفاده از Logcat خطاها را پیدا کن

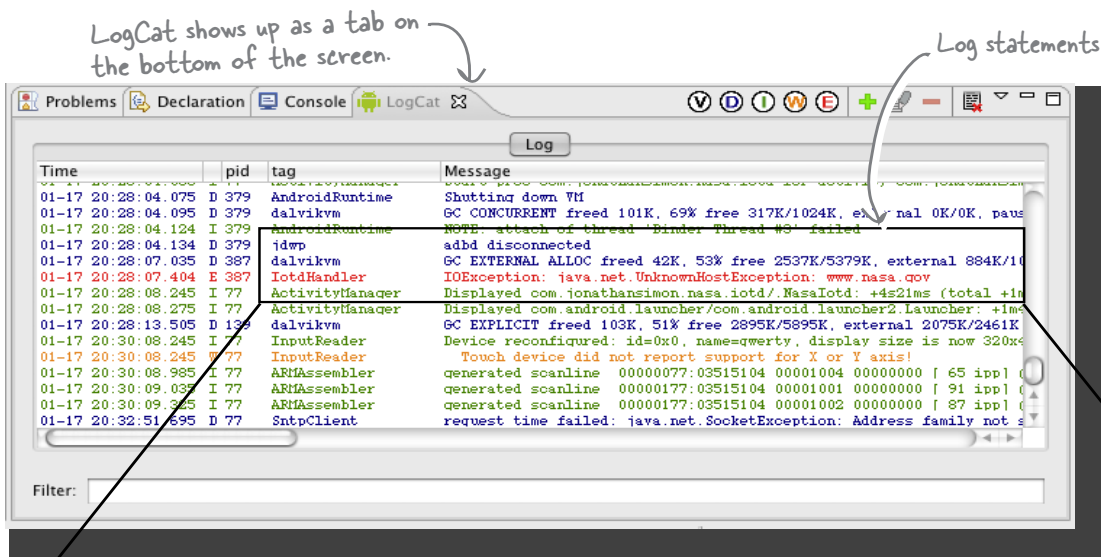
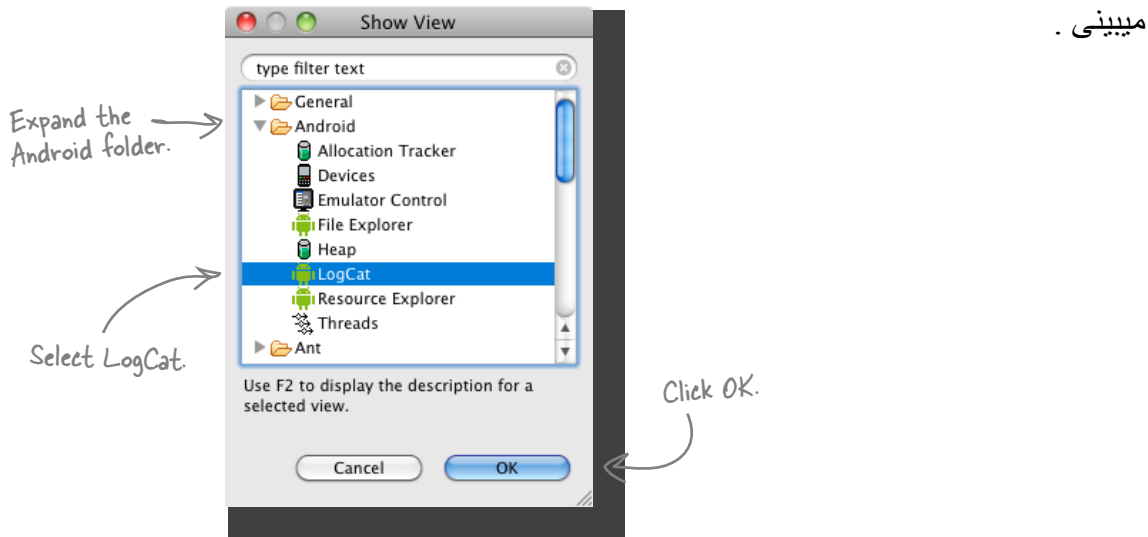
باشه قبول ، خطایی رخ داده ! مسله مهم اینه که کجا بریم تا بفهمیم چه اتفاقی برای برنامه ات افتاده ، بنابراین تو میتونی چیزهایی که اتفاق افتاده را تعمیر کنی . اندروید چیزی داخل خودش داره که وقایع اتفاق افتاده را گزارش میده و این داخل ADT گنجاده شده و اسمش هم LogCat هست .

برو به مسیر زیر و LogCat را باز کن

Window-show-View-Other

که این کار پنجره ای را باز میکنه از اون اندروید را انتخاب کن و بعدش هم LogCat و OK را فشار بده .

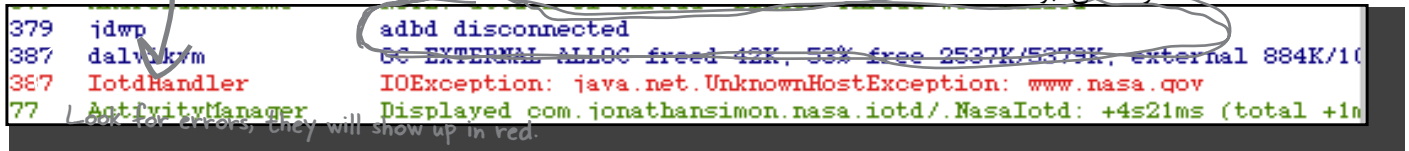
بعد از این کار LogCat را میبینی .



این جا یک IOException دریافت کردی که

میگه هاست پیدا نشده و این عجیبه چون تو با استفاده از مرورگر بت به nasa.gov بارها رفتی و مشکلی نبوده .

دنبال خطاها بگرد ، اونها به رنگ قرمز هستند



از permissions استفاده کن تا دسترسی پیدا کنی

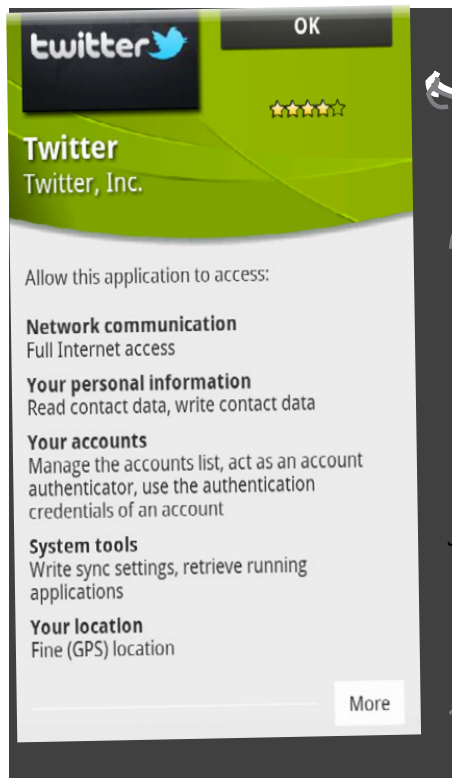
خطای `UnknownHostException` به خاطر این اتفاق افتاده چون تو نیاز به اجازه و `permission` دسترسی به اینترنت را داری .

با این همه چیز باحالی که دستگاهها ی اندرویدی دارن باید به یاد داشته باشیم که اونها دستگاه های موبایل یا قابل حمل هستند . و به همین خاطر هست که اندروید طوری ساخته شده که مطمئن باشه برنامه شما دقیقا به همون قسمتهایی دسترسی داره که واقعا به اونها نیاز داره و تنها راه این کار اینه که برنامه شما اجازه چنین دسترسی هایی را بگیره .

چطور اجازه دسترسی بگیریم ؟

درون اندروید ثابتیهایی قرار داده شده که تو با استفاده از این ثابتها و فایلی به اسم `AndroidManifest.xml` میتونی این مجوزها را بگیری . وقتی کاربران برنامه شما را میخوان نصب کنن با صفحه مواجه میشن که به اونها لیستی از مجوزهای دسترسی را که برنامه شما نیاز داره را نشون میده . اگه اونها موافق باشن برنامه شما نصب میشه .

برای مثال ، یک نگاهی به صفحه نصب برنامه `Twitter` بندازید .



صفحه نصب برنامه `Twitter` در فروشگاه اندروید

همه دسترسی هایی که برنامه میخواد

```
android.permission.INTERNET

android.permission.READ_CONTACTS
android.permission.WRITE_CONTACTS

android.permission.MANAGE_ACCOUNTS
android.permission.AUTHENTICATE_ACCOUNTS
android.permission.USE_CREDENTIALS

android.permission.WRITE_SYNC_SETTINGS
android.permission.GET_TASKS
android.permission.ACCESS_FINE_LOCATION
```

این مجوزهایی که میبینید از کدهای روبرو تولید شدن

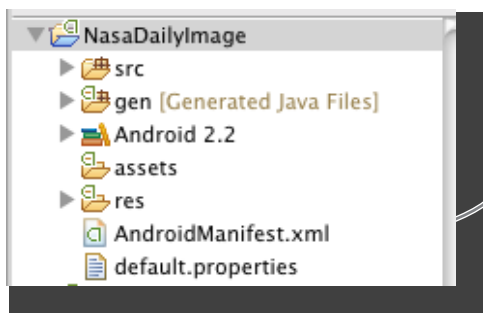
on the market
page ar

خپ در مورد `Twitter` کافیه . حالا بریم تا این مجوزها را به برنامه ات اضافه کنیم !

برنامه Twitter نیاز به مجوزهای زیادی داشت اما برنامه تو فقط مجوز دسترسی به اینترنت را میخواهد. برای این کار دستورات زیر را دنبال کن.

1 فایل AndroidManifest.xml را باز کن.

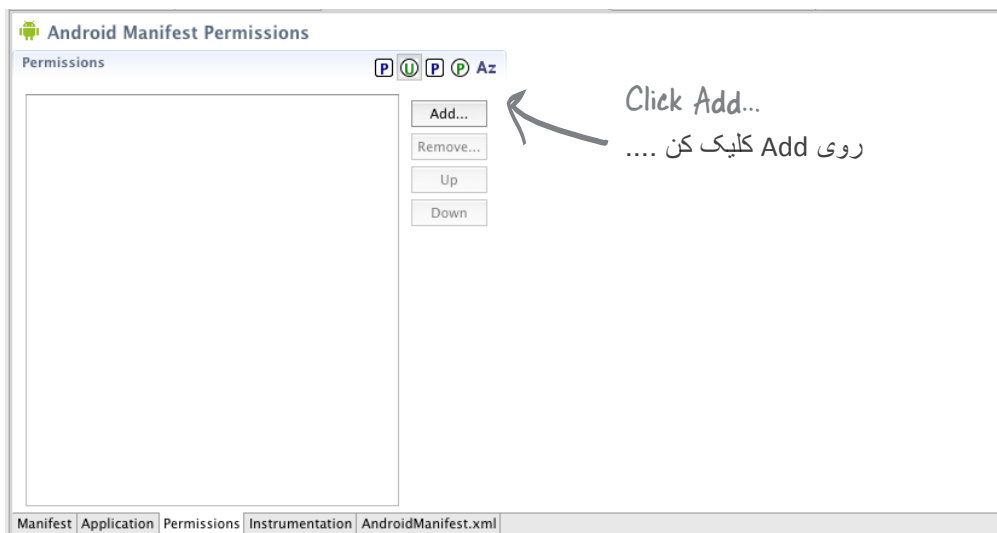
فایل androidmanifest.xml همون اول کار تولید شده و میتونی اون را در ریشه برنامه ات پیدا کنی. روی اون دو بار کلیک کن تا باز بشه.



Find the AndroidManifest XML file in the root of your project.

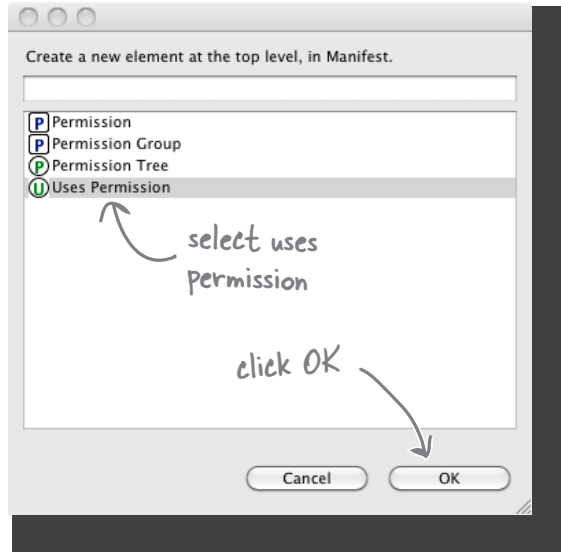
2 یک مجوز تازه به اون اضافه کن

درست مثل همه ویرایشگرهای XML که با اونها کار کردی یک ویرایشگر برای این فایل وجود داره. روی تب Permissions کلیک کن و روی دکمه Add برای اضافه کردن مجوز جدید کلیک کن.



تب Permissions را انتخاب کن

وقتی صفحه زیر باز شد گزینه Use Permission را انتخاب کن و OK را فشار بده . این به اندروید میگوید که تو میخواهی از permission در برنامه ات استفاده کنی .

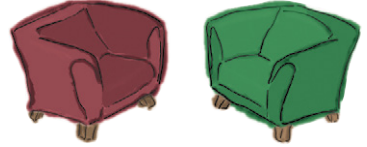


مجوز دسترسی ات را انتخاب کن

مجوزهای زیادی هستند که تو میتونی به برنامه ات اضافه کنی . چون تو برای دریافت فیدها به اینترنت نیاز داری android.permission.INTERNET را انتخاب کن .



برای اینکه این تغییرات اعمال بشن ، این فایل را ذخیره کن .



Android Operating System:

نه این مسخره نیست . من باید در مورد کاری که کنترل شده نیست خیلی محتاط باشم.

گوش بده ، کاربرهای من (که البته کاربر هم هستند) از ما میخوان تا طوری با هم کار کنیم که امنیت کل دستگاه حفظ بشه . ما نمیتونیم به هیچ ویروس یا هر چیزی که محل امنیت هست ، اجازه بدیم مقاصد خودشون را عملی کنن . اگه این طور بشه هر دوی ما باختیم !

شرمنده ، ولی باید این کار را بکنی . با این کار من میتونم به کاربرهامون بگم که تو چه کاری میخوای انجام بدی و اونها هستند که تصمیم میگیرن این کار را بکنی یا نه .

از کجا معلوم کاربر به تو گفت هم گوش دادی؟ در حالی اگه من سر راهت نبودم، تو به من هم گوش نمیدادی .

یعنی تو گوش میدادی ؟

من نظرم عوض نمیشه !

جواب منفیه ! همینه که هست میخوای بخواه نمی خوای نخواه .

Android App:

چی ؟ راست میگی؟ یعنی من باید هر کاری که میکنم از تو اجازه بگیرم ؟ تو به من اعتماد نداری؟ این خیلی مسخره است؟

کنترل نشده؟! ببین من که بچه نیستم !

OK ، حرفت درست . اما من باید همه چیز را به تو بگم ؟ همه چیز؟

چرا خودم این کار را نکنم ؟

خیلی دیگه تند رفتی .

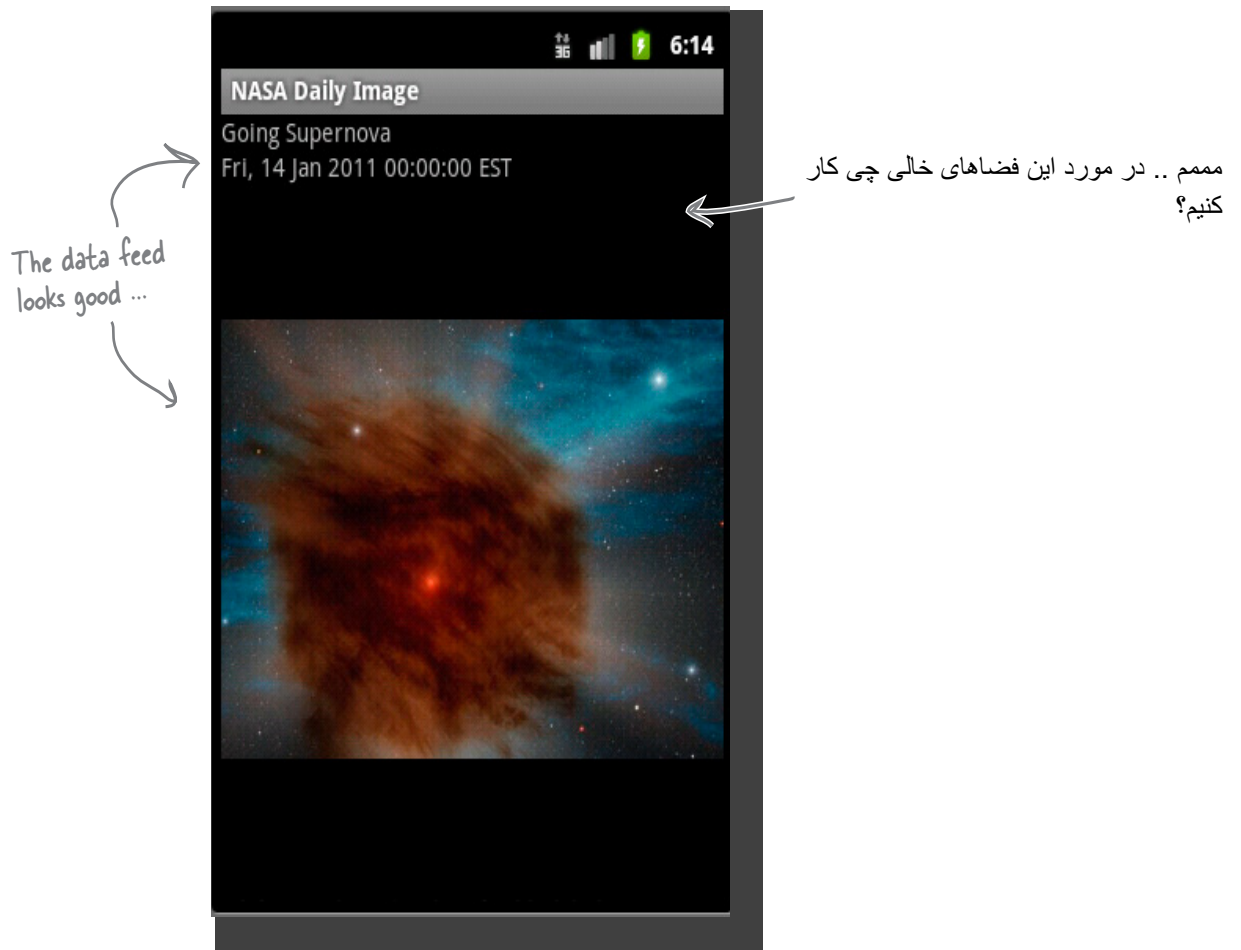
حق با توست ، شاید گوش نمیدادم . اما.....

اااااااا...!باشم مثل این که چاره دیگه ای ندارم، شایدم دارم ؟



TEST DRIVE

حالا مجوزها صادر شد و برنامه باید درست کار کنه و فید را تجزیه کنه و همه چیز را نشون بده . برنامه را اجرا کن !



بهتر شد ، اما هنوز مانده !

فید کار میکنه (عالیه) و داده های تازه روی صفحه به نمایش درآومدن . خوبه اما، یک نقص کوچولو توی قالب هست .

چطور بفهمیم این نقص چیه ؟

اگه چیزی برای اشکال زدایی هست پس میشه از اون برای پیدا کردن اشکال در لایه ها هم استفاده کرد



در حقیقت یک چنین ابزاری هست

اسم این ابزار Hierarchy Viewer هست. این ابزار خوب و مفید را که در android SDK در اختیار شما قرار میده ، به شما این امکان را میده که درون لایه ها و Viewها تون را ببینین

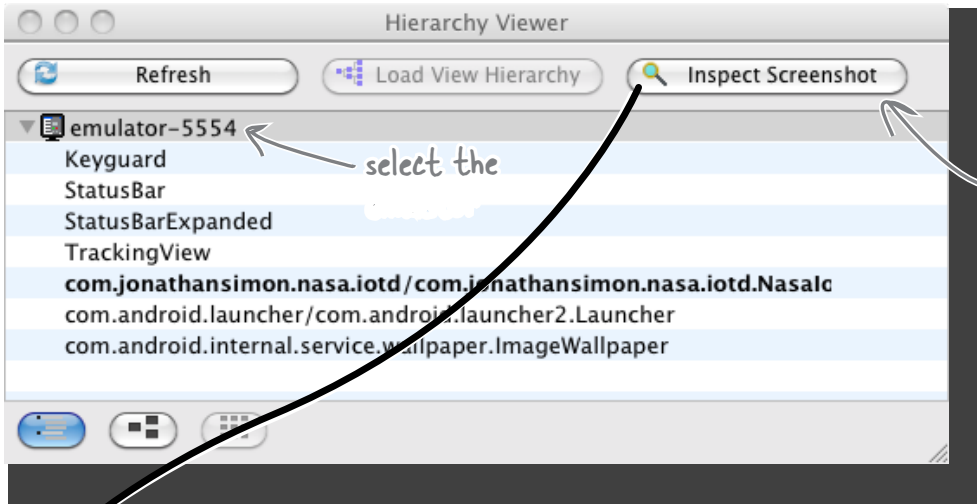


Hierarchy Viewer را از مسیر زیر اجرا کن
<code>sdk</code>/tools اجرا کنید.

مشکلات لایه را با استفاده از hierarchy Viewer پیدا کن .

وقتی hierarchy Viewer را اجرا کردی اولین چیزی که میبینی صفحه زیر هست . در اینجا دوتا قسمت هست ، اولی به تو کمک می کنه که Viewها را در نمودار درختی ببینی و اطلاعات این Viewها را ببینی (قسمت دوم هم میتونه مفید باشه چرا که اون هم اطلاعات را با جزئیات بالایی نشون میده)

وقتی hierarchy Viewer را اجرا کردی این صفحه را میبینی .



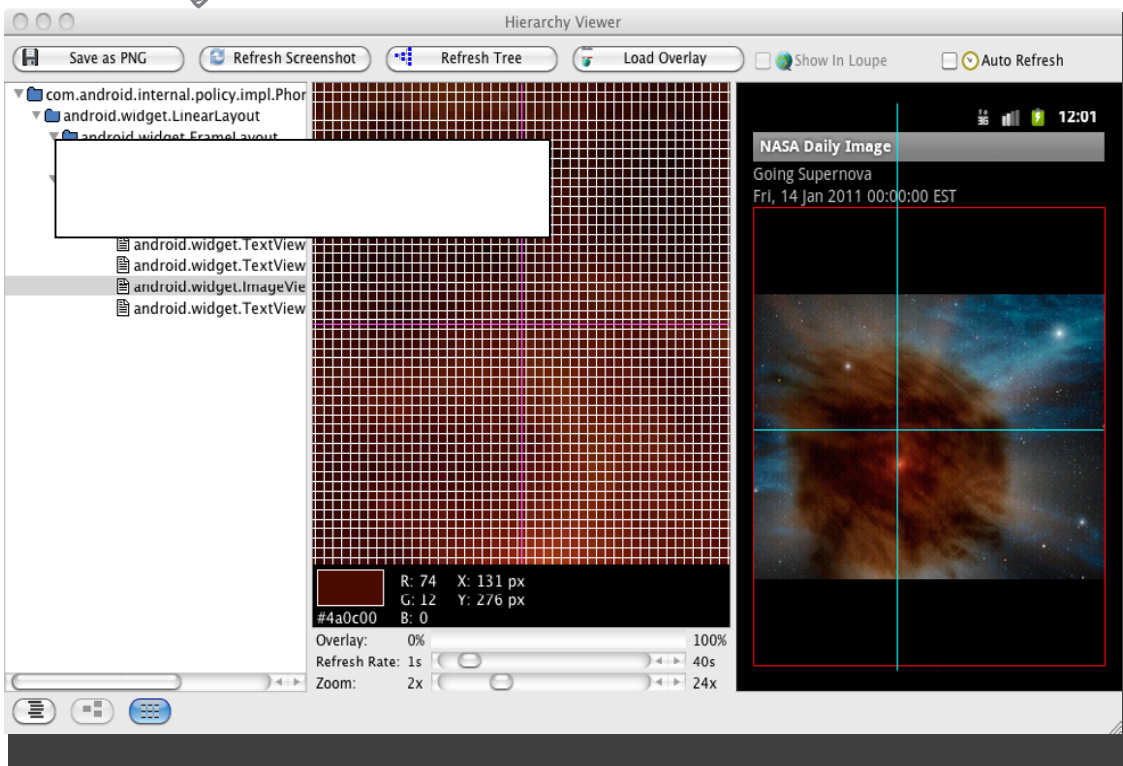
بعد از این که شبیه ساز را انتخاب کردی اینجا را کلیک کن تا پنجره اصلی اجرا بشه

select the

قسمت بزرگنمایی شده (این قسمت جایی که علامت بعلاوه سمت راست نشون میده روم کرده)

این هم برنامه تو هست . (مربع قرمز View ای را احاطه کرده که در پنل سمت چپ انتخاب شده)

سلسله مراتب Viewها

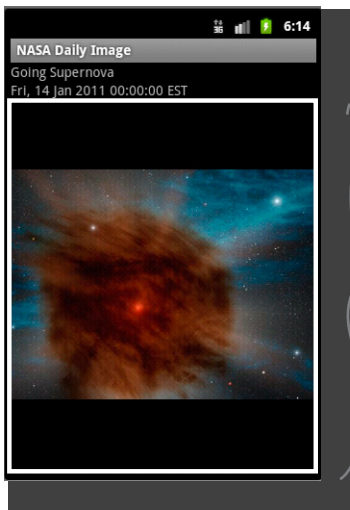


اینها فضاهای اضافی هستند... درست کنار ImageView (این حرف درسته چون گزینه ImageView از پانل سمت چپ انتخاب شده و مربع قرمز هم دور اون را گرفته)

مشخصه adjustViewBounds را تنظیم کن

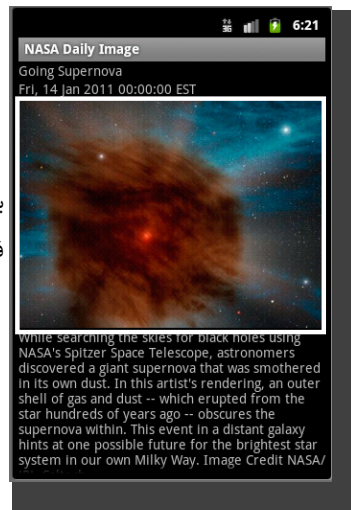
تو با استفاده از hierarchy Viewer میبینی که imageView خیلی بزرگه اما چرا؟ علتش اینه که نسبت عکس در زمان نمایش حفظ نمی شه. در واقع نسبت عکس کمیتی هست که نسبت پهنا به ارتفاع عکس را زمانی که اندازه ان کوچک یا بزرگ میشه حفظ میکنه و در برنامه این عکس توسط کدهای داخلی لایه تغییر اندازه داده شده.

adjustViewBounds = **false**



بدون حفظ نسبت عکس، این عکس کش میاد و فضای زیادی را اشغال میکنه.

adjustViewBounds = **true**



عکس تا لبه های صفحه کش اونده و ارتفاع اون متناسب با پهناى اون تنظیم شده

اگر تو مشخصه adjustViewBounds را true قرار بدی، فضاهای اضافی از بین میرن.

main.xml

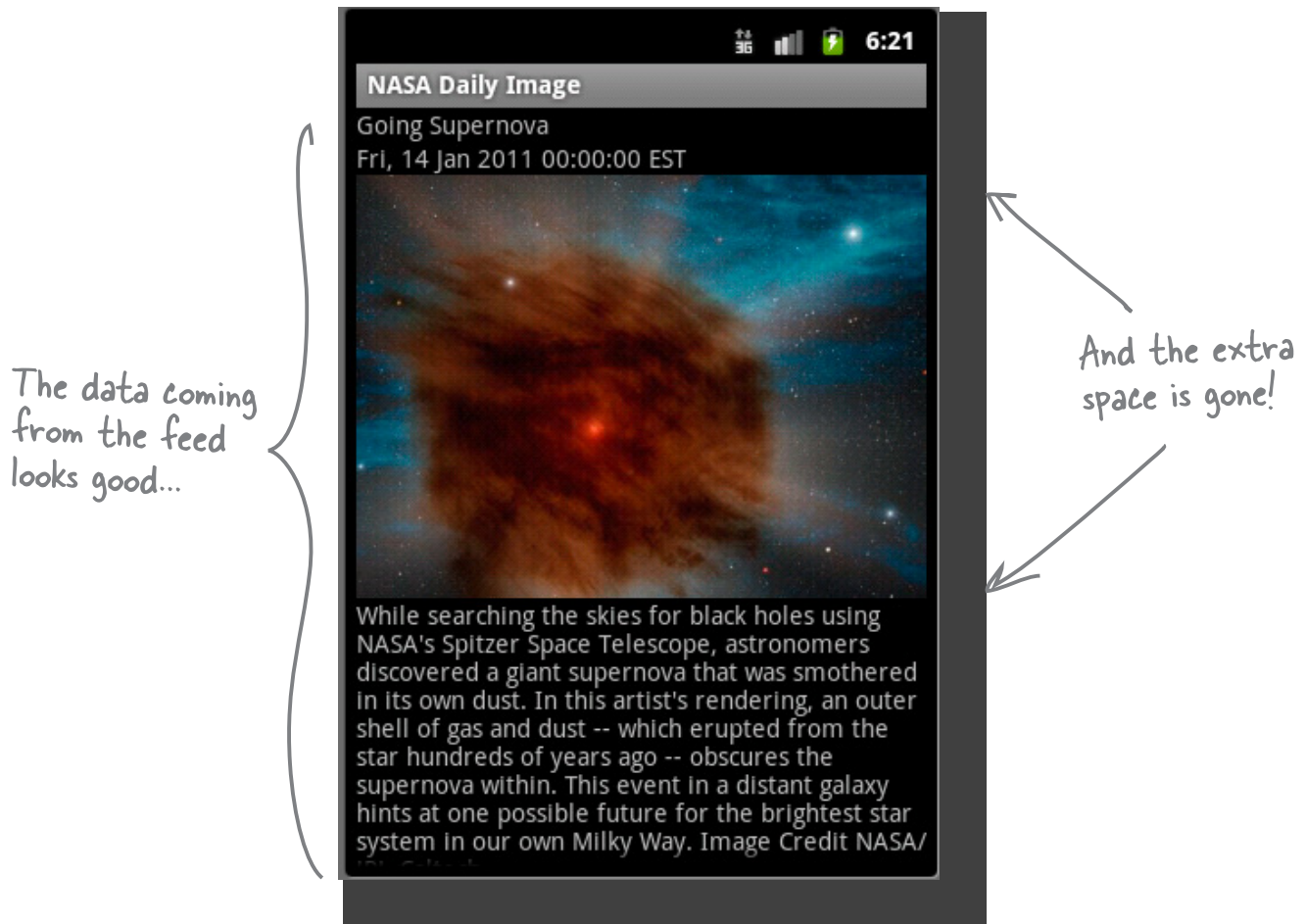
```
<ImageView
    android:id="@+id/imageDisplay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:adjustViewBounds="true" />
```

Set this property in your layout XML.



TEST DRIVE

حالا که `adjustViewBounds` را به درستی تنظیم کردی ، برنامه را دوباره اجرا کن . این دفعه باید ببینی که عکس به درستی تغییر اندازه داده شده .



همه چیز به درستی در کنار هم قرار گرفت !

لایه دقیقا به همان صورتی که طراحی کرده بودی کار کرد ، تجزیه گرهیم به درستی اطلاعات را تجزیه کرد و مشکل `ImageView` هم حل شد .



کارت عالی بود!

تو از تمام مهارتت در برنامه نویسی اندروید استفاده کردی تا یک برنامه جدید بسازی! و در طول راه هم چیزهای جدیدی یاد گرفتی مثل اضافه کردن اسکروول بار، عکس و اما مهمتر از اینها تو کاربردت را خوشحال کردی!

حالا تو یک تجزیه گر فیدهای

RSS با حال توی چعبه ابزارت داری

و میتونی هر نوع برنامه ای که

با فیدها کار میکنه را بسازی .



از



نقاط مهم

وقتی خواهی با فیدهای RSS کار کنی یک نمونه از اون را دانلود کن

و تصمیم بگیر که کدام جز از اون را میخوای توی برنامه ات نمایش بدی .

با تجزیه گر SAX شروع کن اما ببین اگه DOM and XMLPULL

بهتر کار میکنن ، از اونها استفاده کن .

این یک تمرین خوبییه که برنامه ات را به قطعات کوچکتر تقسیم کنی .

برای مثال در مورد RSSها که نیاز داری به اینترنت متصل بشی ،

اول برنامه ات را با داده های تمرینی بساز و بعد میتونی اون را به اینترنت متصل کنی

عکسها را به دایرکتوری res/drawablehdmی اضافه کن .

اینها توسط اندروید گرفته میشن و تو میتونی در برنامه ات از

اونها استفاده کنی .

از ImageView برای نمایش عکس استفاده کن .

زمانی که محتوای برنامه ات از صفحه بزرگتر هست

از ScrollView استفاده کن . (فقط یادت باشه که ScrollView

میتونه یک View به عنوان ورودی داشته باشه)

وقتی چیزی اشتباه کار میکنه از LogCat برای پیدا کردن خطاها استفاده کن .

مطمئن شو که برنامه ات مجوزهای لازم را کسب کرده .

وقتی برنامه ات چیزها را درست نشون نمیده

از hierarchyViewer برای پیدا کردن خطاها

استفاده کن .

مشکل یابهای درونی

LogCat به منظور پیدا کردن خطاها در برنامه استفاده کن

برای اشغال زدایی از Viewها و لایه ها از hierarchyViewer استفاده کن

معمولاً بزرگی به تو بکنه .

وقتی که لایه ها مطابق میل تو نیستن این ابزار میتونه

جمع بندی در مورد Viewها

از TextView برای نمایش متن ها استفاده کن . از متن های

کوچیک مثل عنوان تا متنهای بزرگ مثل توضیح عکس ، برای

هر دو میتونی از این استفاده کنی .

از imageView برای نمایش عکس استفاده کن .

تو میتونی عکسهای خودت را به منابع اضافه کنی

و از اونها در imageView استفاده کنی .


4 پردازشهای طولانی مدت

وقتی کارها زیاد طول میکشند



اوه ، باشه انجامش میدم اما قبل از اون باید صبحانه ام را بخورم ، قهوه ام را میل کنم (مترجم :نوش جوننت) و روزنامه ام را تمام کنمو

خیلی خوب میشد اگه همه چیز سریعاً اتفاق می افتاد. اما از شانس بد ما چیزهایی هستند که زمان بر هستند . این مسله خودش را در دستگاههای موبایل بیشتر نشان میدهد مثلاً وقتی که شبکه تاخیر داره یا پردازش کندی بر روی موبایل در حال اجراست . تو میتونی برنامه ات را با استفاده از بهینه سازی سریع بر کنی اما با این کار هم بلاخره چیزهایی وجود دارن که طول میکشن . اما تو میتونی یاد بگیری که چطور این پردازشهای طولانی را مدیریت کنی . در این فصل تو یاد میگیری که چطور حالت فعال یا خنثی را به گاربرها نشون بدی . همچنین یاد میگیری که چطور عملیاتیایی که سنگین هستند را از پردازشهای واسط کاربری جدا کنی تا با این کار تضمین کنی که برنامه ات همیشه جواب گو هست .



من از برنامه ناسا استفاده میکنم و ازش راضی
هستم . اما ناسا این فید را چند بار در روز بروز
رسانی میکنه . به نظرت میتونی دکمه Refresh به
اون اضافه کنی ؟ چون من مجبورم که برنامه را
ذوباره اجرا کنم تا بروزرسانی ها را ببینم

به نظر میرسه که درخواست منطقی هست

اما چه نیاز به دکمه refresh داری ؟ تو باید برنامه ات را با توجه

به بازخوردهای کاربرد ارتقا بدی اما این را هم باید بدونی که چرا

چنین درخواستی از تو شده . از تو خواسته شده تا یک دکمه

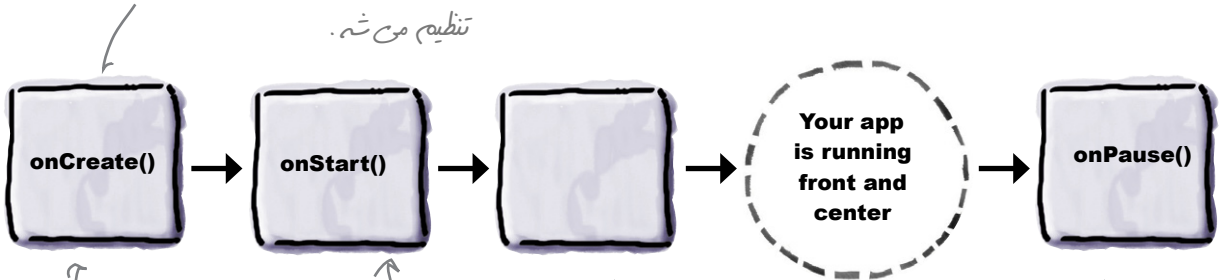
refresh اضافه کنی . بزار یک نگاهی به چرخه زندگی یک اکتیوینی

بندازیم تا ببینیم که فید چه زمانی بارگذاری میشه و چرا همین برای بای کافی نیست

چرخه زندگی یک اکتیویتی

اکتیویتی یک سری توابع خاص دارد که به اونها توابع چرخه زندگی میگن و در مراحل اجرا شدن اکتیویتی فراخوانی میشن . تابع onCreate() که شما در اون لایه XML را فراخوانی می کردید یکی از این توابع است و باید بگم که توابع بیشتری هم هستند . تعدادی از این توابع در زیر نمایش داده شده , بنابراین می تونی ببینی که فید ما در کجاها refresh می شه یا نمی شه .

اینجا جایه که فید در اون پردازش میشه و چیزی که می خواهید روی صفحه نمایش ظاهر بشه تنظیم می شه .



اینجا جایه که اکتیویتی شما خلق می شه و در اصل اتصال کایه و پیکر بندی اون صورت میگیره .

این تابع زمان فراخوانی میشه که اکتیویتی شما روی صفحه نمایش داده میشه .

این اتفاق میتونه وقتی یوقته که مثلاً تلفن زنگ بزنه یا یک پیغام از سایر برنامه ها صادر بشه یا

این که خود کاربر به برنامه رفته اک بره



این فقط قسمتی از چرخه زندگی اکتیویتی هست و شکل کامل اون را میتونید در مستندات اینترنت پیدا کنید

چه موقع فید refresh میشه ؟

onCreate()

فید فقط زمانی refresh می شه که اکتیویتی شروع میشه و متد فراخوانی میشه . فید زمانی که برنامه شروع شد , دیگه refresh نمیشه . در حال حاضر تنها راه این که برنامه ما فید را تازه سازی کنه اینه که از برنامه خارج بشی و دوباره اون را اجرا کنی .

برای این که فید را تازه سازی کنی کارهای مختلفی میتونی انجام بدی , برای مثال میتونی توابع چرخه زندگی اکتیویتی مثل onResume() را باز نویسی کنی , اما این فقط وقتی کار میکنه که برنامه به حالت مکث بره و برگرد (مترجم : مثل زمانی که داخل برنامه هستی بدون این که اون را بندید به برنامه دیگه ای برید و دوباره برگردید , که همونطور که فکر می کنید کار عاقلانه ای نیست) یا شما می تونید یک مکانیزم بسازید که فید به طور خودکار تازه سازی بشه اما این کار هم سی پی یو و هم باتری زیادی مصرف میکنه . پس به نظر میرسه که اگه یک دکمه بسازید که این کار را انجام بده فکر بهتری باشه .

بروز رسانی رابط کاربری

با دیدن رابط های کاربری متوجه میشوید که نوار مربوط به دکمه یک پانل خاکستری در پایین صفحه است که یک یا چند دکمه را در خودش جا داده . به نظر میرسد این پانل محل مناسبی برای قرار دادن دکمه refresh باشد .

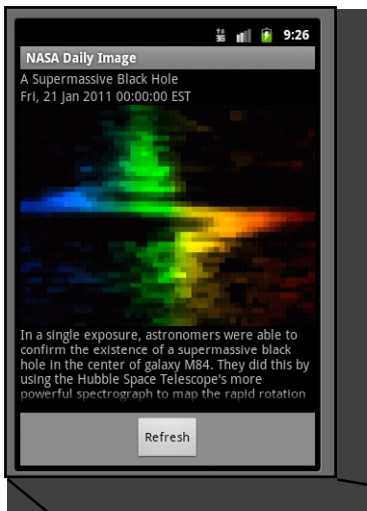
ما کار را به این صورت انجام میدیم که button bar (نوار دکمه ها) را به عنوان یک لایه مستقل درست می کنیم و بعد اون را به لایه حال حاضر برنامه اضافه می کنیم . تکه تکه کردن قسمتهای مختلف صفحه شما به قسمتهای کوچکتر فکر خوبی برای مدیریت بهتر لایه ها هست . از اونجا که LinearLayout زیر شاخه ViewGroup هست که اون هم زیرشاخه View هست پس تو میتونی کل LinearLayout ای که برای buttonbar می سازی را به عنوان فرزند ViewGroup اصلی معرفی کنی .

(مترجم: شاید مطالب بالا کمی مبهم باشه اما در ادامه بیشتر توضیح داده میشه اما شاید سوال پیش بیاد که چرا این دکمه را مثل سایر قسمتها به ScrollView متصل نمی کنیم و اون را در LinearLayout جدا قرار میدیم و بعد هم ScrollView و هم Buttonbar را داخل یک LinearLayout دیگه میزاریم؟ اگه شما ساختار لایه اولی را در ذهنتون تجسم کنید متوجه میشوید که اگه ما این کار را بکنیم Buttonbar جزو ScrollView میشه و اگه بخواییم صفحه را بازسازی کنیم باید اونقدر بیایم پایین تا دکمه refresh پیدا بشه اما با کار دوم دکمه همیشه در دسترس ماست)

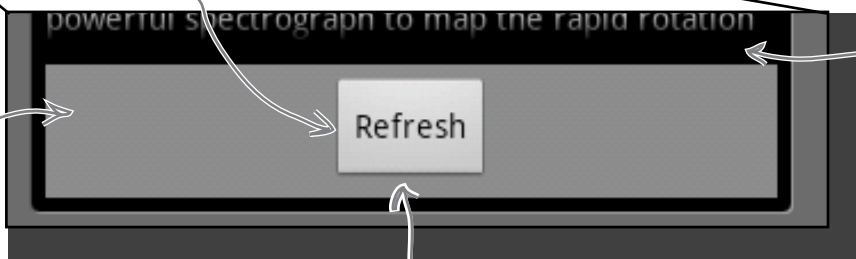


نمایش Button bar در یک صفحه

نمونه ای از Button bar در یک صفحه اضافه کننده مخاطب



دکمه در مرکز قرار داره



between the text and the button bar

پس زمینه خاکستری

آیکون فضای خالی کمی بین دکمه و ButtonBar وجود داره .

شروع کار با LinearLayout

LinearLayout یک لایه خوب برای ساخت صفحات هست . تو تا الان چندتا صفحه با اون ساختی و الان میریم تا یک Button bar را هم با اون بسازیم . تو در مورد LinearLayout در ادامه بیشتر یاد می گیری و همچنین در مورد لایه های دیگه یا بهتر بگم layoutmanagerهای دیگه هم چیزهای زیادی یاد میگیری , پس نگران نباش .

کلید استفاده از LinearLayout برای button bar اینه که دکمه بازسازی یا همون refresh را با استفاده از android:gravity در مرکز قرار بدی و بعد به راحتی لایه را تنظیم کنی .

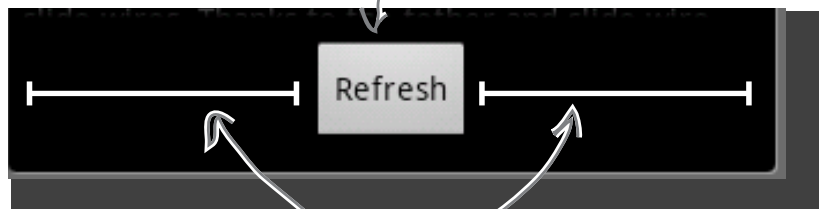
کدهای لایه buttonbar که یک

LinearLayout با یک button در وسطش

هست.

```
<LinearLayout
  android:orientation="horizontal"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:gravity="center"
  >
  <Button android:text="@string/refresh"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
```

این قطعه کد اضافه هست چون این خصوصیت به طور پیش فرض افقی هست اما بهتره که نوشته بشه .



دکمه در وسط قرار داره

تو شروع خوبی داشتی ! حالا بریم که این LinearLayout را بهتر تنظیم کنیم

از خصوصیات استفاده کن تا لایه را بهتر تنظیم کنی

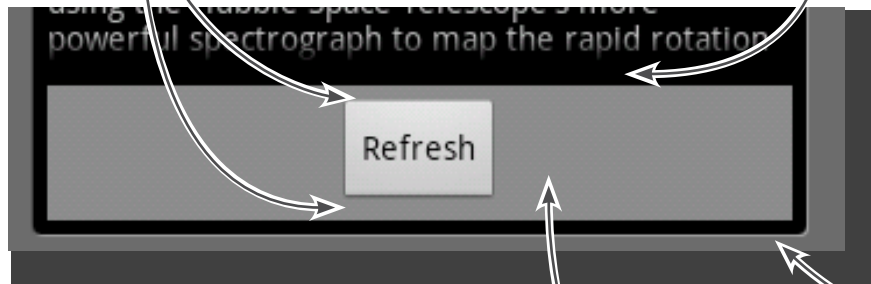
حالا که دکمه به خوبی در مرکز قرار گرفت ، بهتره روی رنگ و فضای بین دکمه و نوار دکمه تمرکز کنیم ، از این خصوصیات به صورتی باید استفاده کنی که نوار دکمه به صورتی که دوست داری به نظر بیاد.

Padding

این خصوصیت فاصله بین View ها در داخل لایه را کنترل می کنه . به جای واحد پیکسل خالی از واحد DIP (Density Independent Pixels) برای فاصله دادن استفاده کن تا لایه ات منعطف تر بشه .

Margin

این خصوصیت فاصله بین این View و View های خارج را کنترل می کنه . به جای واحد پیکسل خالی از واحد DIP (Density Independent Pixels) برای فاصله دادن استفاده کن تا لایه ات منعطف تر بشه .



```
android:background="#ff8D8D8D"
```

background

این خصوصیت را میتونی به یک عکس ، یک رنگ و تعدادی دیگه از عناصر گرافیکی اندروید نسبت داد . از یک رنگ ثابت برای پس زمینه پانل دکمه استفاده کن . این رنگ به صورت هشت رقم در مبنای هگزادسیمال بیان میشه ، دوتا برای کانال آلفا ، دوتا برای قرمز ، دوتا برای سبز و دوتا هم برای آبی .

FF 8D 8D 8D

Alpha Red Green

عرض و ارتفاع لایه

اندازه دکه را wrap_content قرار بده ، این طوری دکه به اندازه ایش کش میاد که فقط متن داخل دکه را پوشش بده .

```
android:layout_height="wrap_content"
```

حالا اگه از fill_parent برای عرض

LInearLayout استفاده کنی اون وقت این لایه اونقدر کش میاد تا صفحه را پر کنه .

```
android:layout_height="fill_parent"
```

عرض و ارتفاع لایه ها را همیشه به حالت اولیه خودتون که به ترتیب wrap_content و fill_parent هستند را رها کرد ، همون طور که مقدار اولیه واحد پیکسل DIP هست . استفاده از wrap_content باعث میشه تا عرض یا طول لایه فقط به اندازه ای باشه که محتوا را در بر بگیره در حالی که fill_parent باعث میشه که View ما اونقدر گشیده بشه تا تمام فضایی که بهش داده شده را پر کنه .

Scholar's Corner

Density Independent Pixels (DIP) اندروید روی صفحات با ابعاد

متفاوت اجرا میشه و روی همه به خوبی کار میکنه . اگه از واحد پیکسل

خالی در اندازه لایه هاتون استفاده کنید ممکنه ربامه پیا روی یک دستگاہ

خوب به نظر بیاد اما روی دستگاہ دیگه ممکنه افتضاح باشه . اندروید از

یک واحد انپراعی که به اون Density Independent Pixel

میگن استفاده میکنه که از مشخصات خود دستگاہ مشتق میشه . اس به اس

معنی هست که وقتی تو اندازه لایه هات را با واحد DIP معرفی میکنی لایه

های تو روی همه دستگاہا به خوبی نشون داده می شن .



قطعات مربوط به لایه Button Bar



لایه Button Bar را با استفاده از قطعاتی که در زیر داده شده بساز .

به این فکر کن که عرض و ارتفاع دکمه و لایه ای که اون را در بر میگیره چقدر باید باشه . و نگران نباش در آخر چندتا قطعه باقی خواهد ماند که استفاده نمیکنی .

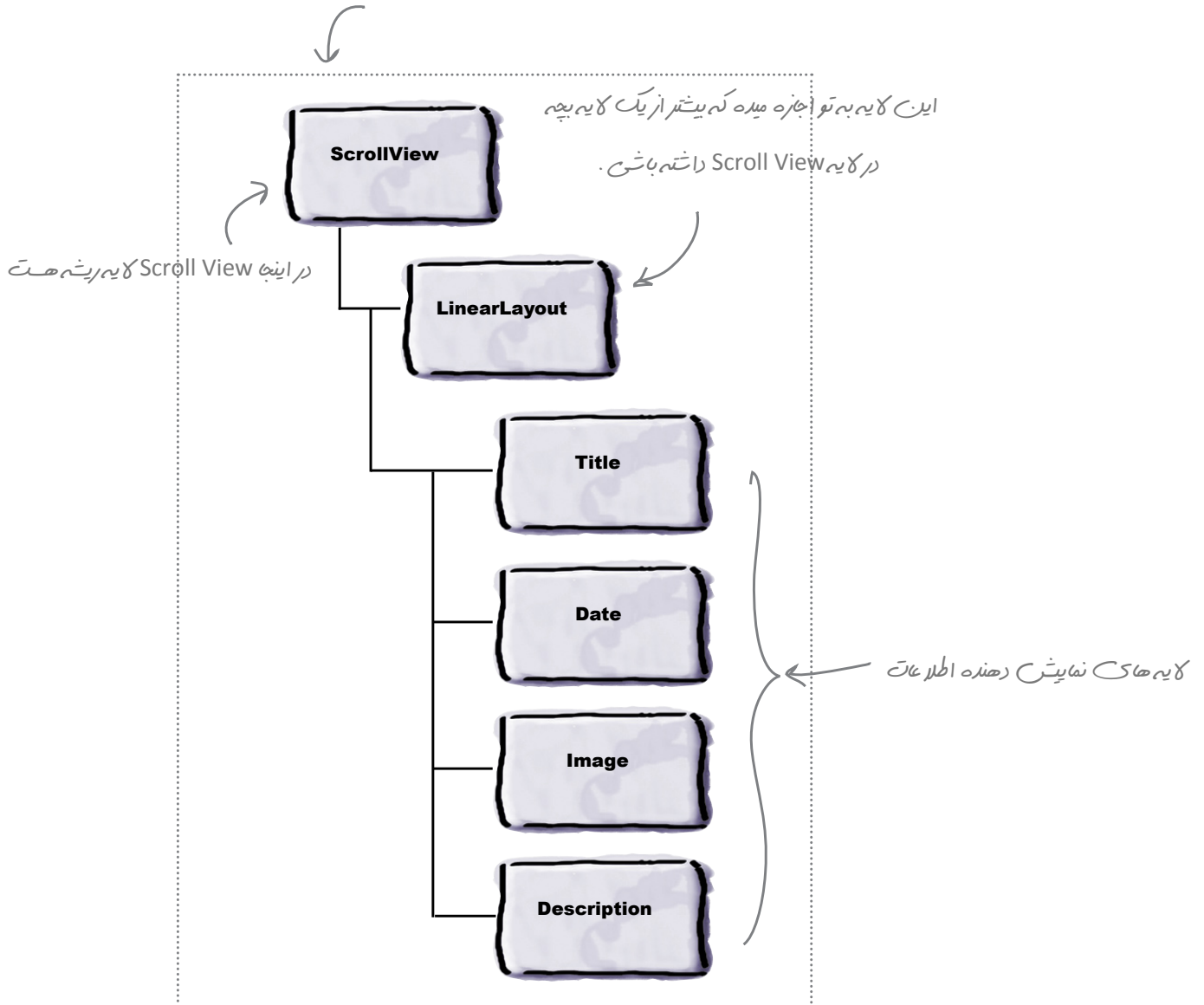




Exercise

حالا که تو لایه مربوط به Button Bar را داری تو باید اون را به صفحه ات اضافه کنی . در زیر یک شمای گرافیکی از سلسله مراتب لایه هات میبینی . لایه مربوط به Button Bar را در اون بخش تا لایه ات کامل بشه . همچنین یادت باشه که همون طور که Scroll View می تونه تنها یک زیر لایه یا به اصطلاح بچه داشته باشه , تو تنها میتونی یک لایه ریشه داشته باشی .

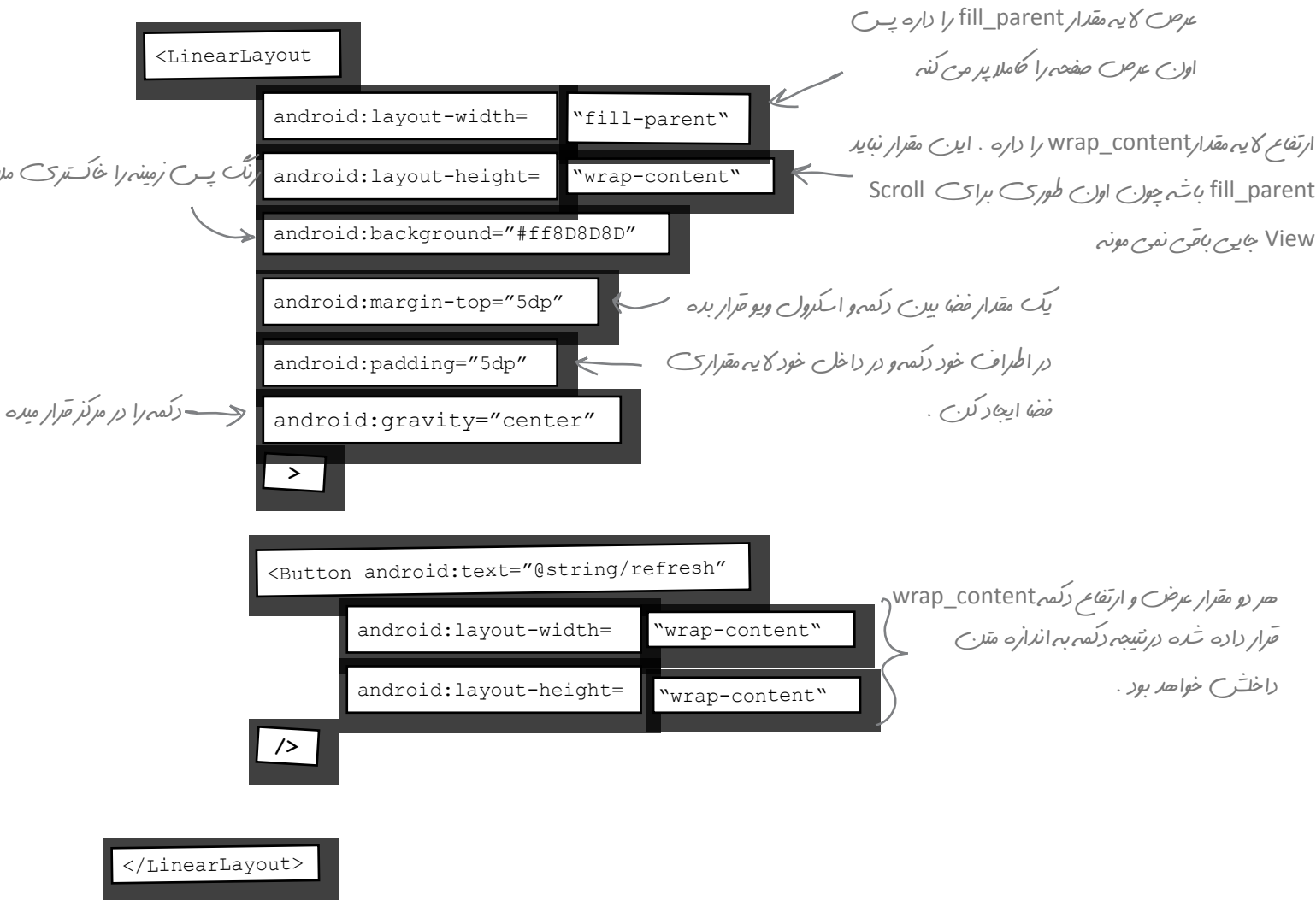
لایه های برنامه بدون Button Bar



راه حل مربوط به قطعات Button Bar



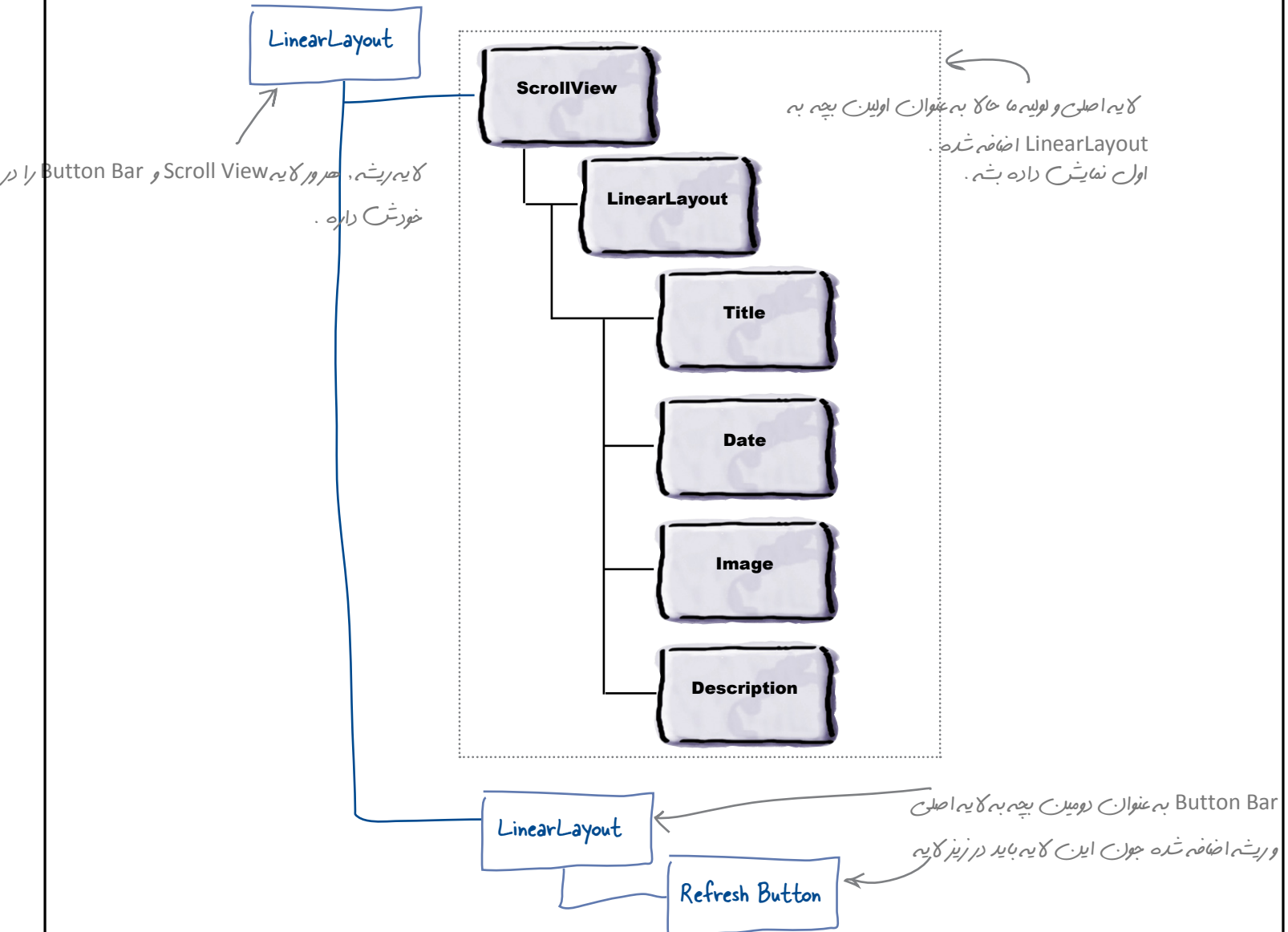
لایه Button Bar را با استفاده از قطعاتی که در زیر داده شده بساز .
به این فکر کن که عرض و ارتفاع دکمه و لایه ای که اون را در بر میگیره چقدر باید باشه . و نگران نباش در آخر چندتا قطعه باقی خواهد ماند که استفاده نمیکنی .





Exercise Solution

حالا که تو لایه مربوط به Button Bar را داری تو باید اون را به صفحه ات اضافه کنی . در زیر یک شمای گرافیکی از سلسله مراتب لایه هات میبینی . لایه مربوط به Button Bar را در اون بخش تا لایه ات کامل بشه . همچنین یادت باشه که همون طور که Scroll View می تونه تنها یک زیر لایه یا به اصطلاح بچه داشته باشه , تو تنها میتونی یک لایه ریشه داشته باشی .



لایه برنامه ات را ابدیت کن

نوار دکمه خودت را به لایه برنامه در فایل main.xml اضافه کن , همچنین Linear Layout در بر گیرنده هر دو لایه ی

Scroll View and Button Bar را به عنوان لایه ریشه به اون اضافه کن .

شروع لایه اصلی

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
```

Xmlns به لایه ریشه منتقل شده

لایه های موجود

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        >
```

Height changed to wrap-content; otherwise, it would fill the screen, leaving no room for the button bar.

```
</LinearLayout>
</ScrollView>
```

همه کارهای مربوط به پانل دکمه

```
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#ff8D8D8D"
    android:layout_marginTop="5dp"
    android:padding="5dp" >
    <Button android:text="@string/refresh"
        android:onClick="onRefresh"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

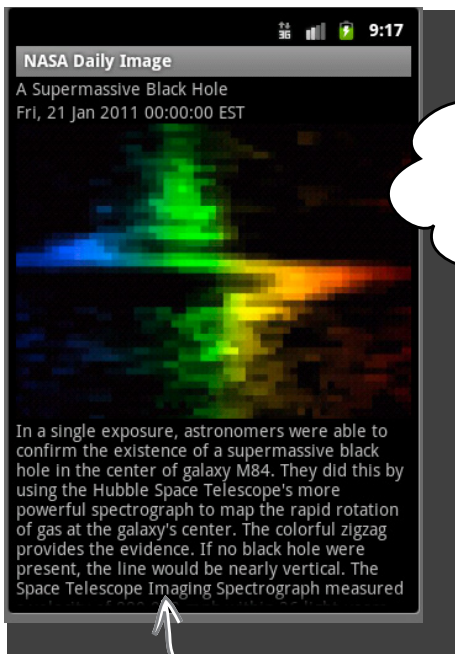
```
</LinearLayout>
```

پایان لایه اصلی



Test Drive

بعد از این که لایه اصلی را در فایل main.xml بروزرسانی کردی ، برنامه ات را اجرا کن تا بفهمیم که لایه درست کار میکنه یا نه ؟



خالا این دکمه ما دقیقاً کجاست؟
این همه روشن کار کردیم حالا
همه اش کشت؟!



پانل دکمه باید اینجا می بود... که نیست .

BRAIN POWER

بک چیزی این وسط درست نیست . عرض و ارتفاع که به نظر درسته و LinearLayout به نظر اندازه اش

درست باید باشه ... درسته ؟ پس مشکل کار کجاست؟

از خصوصیت وزن (weight) برای LinearLayout استفاده کن

LinearLayout به تو این اجازه را میدهد که خاصیتی به اسم وزن را به اون نسبت بدی که این خاصیت عملیات تغییر اندازه را برای View های بچه اش کنترل میکنه . برای پانل دکمه , شما میخواهید که این پانل فقط به اندازه ای باشه که نیاز دارید و بقیه صفحه مال ScrollView باشه .

خاصیت وزن که با کد android:layout_weight در فایل XML مشخص میشه میتونه دو مقدار صفر و یک را داشته باشه . زمانی که این خصوصیت یک هست باعث میشه که View کش بیاد اما وقتی که صفر باشه باعث می شه که اون View به همون اندازه ای باشه که نیاز هست .

```
<ScrollView

    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
```

وزن که یک شده به این معنی هست که این View کش بیاد (مترجم : یا به قول معروف جمع و جور تر بشینه تا جا برای پانل دکمه باز بشه)

کدهای معروف پانل دکمه

```
<LinearLayout

    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0"
```

وزن به مقدار صفر به این معنی که پانل دکمه به اندازه ای باشه که نیاز هست .



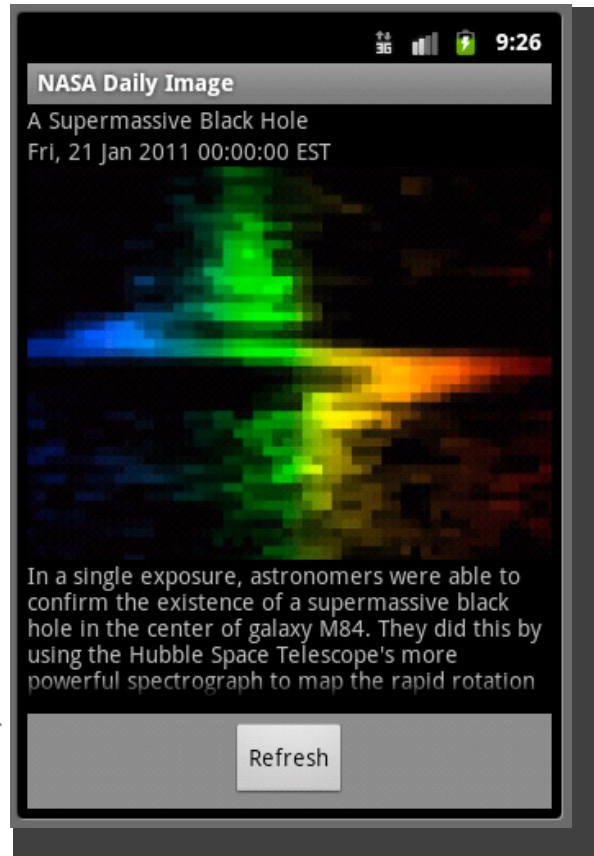
از کجا در مورد این خصوصیات خبر دار بشم ؟

همه این خصوصیتی که در اینجا استفاده شده (و خصوصیات خیلی خیلی بیشتر) همه و همه در مستندات آنلاین اندروید هستند . برای یاد گیری بیشتر به مستندات مربوط به لایه مورد نظرت و همچنین راهنماهای اونها یک نگاهی بنداز .
با یک جستجوی سریع در developer.android.com اونها را پیدا میکنی .
(مترجم : من زیاد اخبار را دنبال نمی کنم اما تا جایی که من خبر دارم الان که این مطالب را مینویسم هم خبرهای خوشحال کننده ای در راه هست و توافق نزدیکه و هم اختلافات بر سر مسایل مهم همچنان باقی هست پس به نظر میرسه سایت بالا همچنان تحریم باشه , از کجا معلوم شاید هم فردا این قفل این سایت توسط یک شاه کلید باز بشه و اون کلید , کلیدی نیست جز کلید اوه اوه داره بحث سیاسی میشه بهتره بریم سراغ کارمون , اخه یکی نیست بگه تو رو چه به سیاست؟!؟)



— Test Drive

برنامه را دوباره اجرا کن و ببین که چطور تغییر دادن خصوصیت وزن لایه ، این لایه را تغییر داده .



کارت عالی بود!

برنامه خوب و رویایی به نظر میرسه . حالا فقط دکمه سالم سازی را به کدهای جاوا وصل کن و اون را به بای نشون بده .

دکمه تازه سازی را وصل کن

تو الان یک کد برای اداره کردن فید خودت از فصل قبل داری . برای این که کدهات خوب و هوشمندانه به نظر برسن و همچنین یک کد را دو بار ننویسی , کدهای اداره کننده فید خودت را به تابع جدیدی به نام `refreshFromFeed()` منتقل کن . سپس تو میتونی این تابع را هم در تابع `onCreate()` و هم در تابع `onRefresh()` فراخوانی کنی .

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    iotdHandler = new IotdHandler();
    iotdHandler.processFeed();
    resetDisplay(iotdHandler.getTitle(),
        iotdHandler.getDate(),
        iotdHandler.getUrl(),
        iotdHandler.getDescription());
    refreshFromFeed();
}
```

کدها را به تابع جدیدی که `refreshFromFeed()` نام داره منتقل کن

تابع `refreshFromFeed()` را داخل متد `onCreate()` فراخوانی کن

```
private void refreshFromFeed() {
    iotdHandler = new IotdHandler();
    iotdHandler.processFeed();
    resetDisplay(iotdHandler.getTitle(),
        iotdHandler.getDate(),
        iotdHandler.getUrl(),
        iotdHandler.getDescription())
}
```

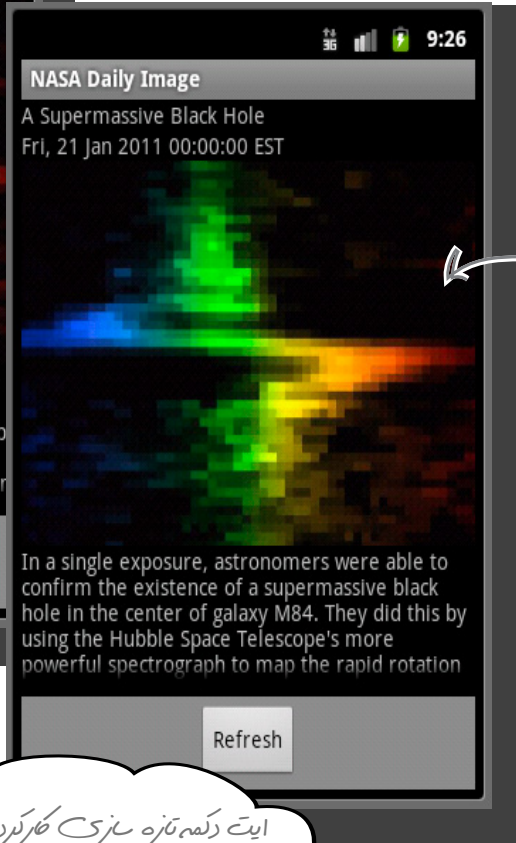
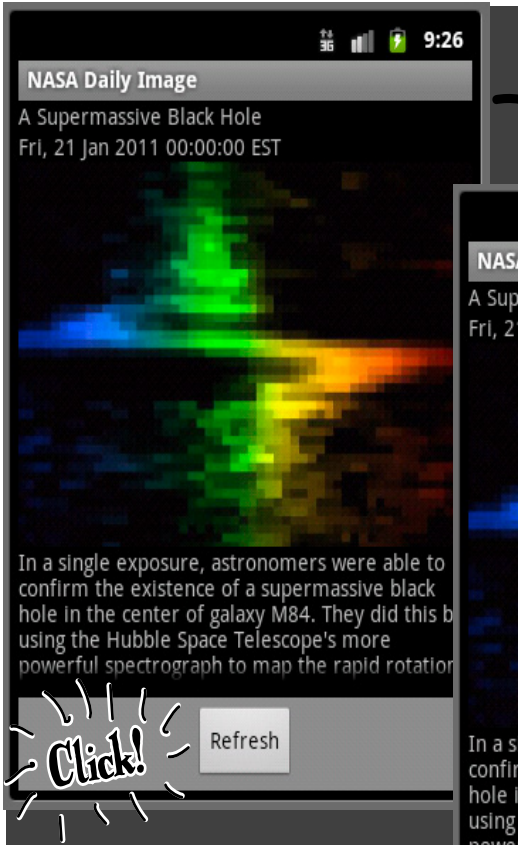
```
public void onRefresh(View view) {
    refreshFromFeed();
}
```

دوباره تابع `refreshFromFeed()` را داخل تابع `onRefresh()` فراخوانی کن .



Test Drive

برنامه را دوباره اجرا کن و روی دکمه refresh کلیک کن . این کار باید برنامه را از فید ابدیت میکند .



ایت دکمه تازه سازی کار نکرد من که ندیدم اتفای روی صفحه نیفتد ...



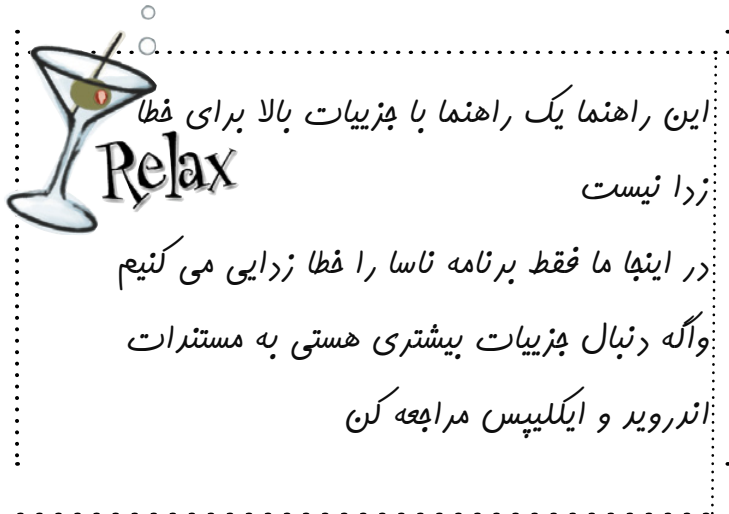
دقیقا معلوم نیست اینجا چه خبره ...

ایا دکمه به خوبی کار کرد؟ ایا فید به خوبی پردازش شد؟ کاملا مشخص نیست

که وقتی روی دکمه کلیک میکنی چه اتفاقی می افته .

از خطا زدا استفاده کن

خطا زدا یا debugger یک ابزار فوق العاده مفید هست برای این که بفهمی وقتی داده برنامه ات اجرا میشه ، دقیقا چه اتفاقی داده می افته . پلاگین اندروید برای Eclipse شامل ابزاری برای خطا زدایی برنامه ات هست هم برای شبیه ساز و هم برای دستگاه واقعی . به چیزهایی که در ادامه گفته میشه عمل کن تا بفهمی که آیا اصلا تابع refreshFromFeed() فراخوانی میشه .



1 یک breakpoint بزار

خطا زدا با استفاده از یک سری نقاط ترمز کار می کنه که به

اونها

Breakpoint می کن . یک breakpoint مثل یک توقف

میمونه که صبر می کنی و میبینی در اون محل چه اتفاقی

داده می افته .

```
input.close();
connection.disconnect();
return bitmap;
} catch (IOException ioe) {
    return null;
}
}

public void refreshFromFeed() {
    iotdHandler = new IotdHandler();
    iotdHandler.processFeed();
    resetDisplay(iotdHandler.getTitle(), iotdHandler.getDate(), iotdHandler.getU

}

public void onRefresh(View view) {
    refreshFromFeed();
}
```

→ در حاشیه خاکستری رنگ رو بار کلیک کن

خطا زدا به صورت خودکار

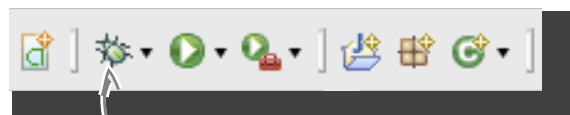
به برنامه ات بر روی

دستگاه یا شبیه ساز

متصل میشه .

2 Launch the debugger

The debug button is just to the left of the play button in the Eclipse toolbar. It uses the Android launch configurations you already set up. Press it to launch the debugger.



روی این دکمه کلیک کن تا خطا زدا اجرا بشه

برنامه ات را مشاهده کن و اون را خط زدایی کن.

پنجره خط زدا جایبه که تو میتونی حالت برنامه ات که داره اجرا میشه را ببینی.

وقتی تو برنامه ات را با استفاده از خط زدا اجرا کردی او خیلی سریع به

breakpoint برخورد میکنه. برای این که متد onCreate() متد

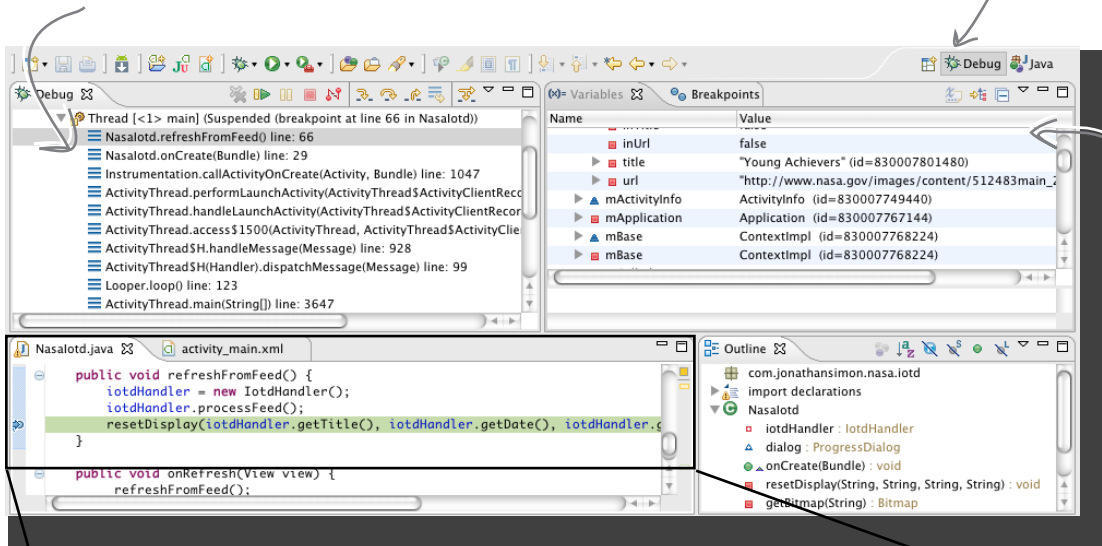
refreshFromFeed() را فراخوانی میکنه و این تابع جایبه که breakpoint تو در

اون قرار داره.

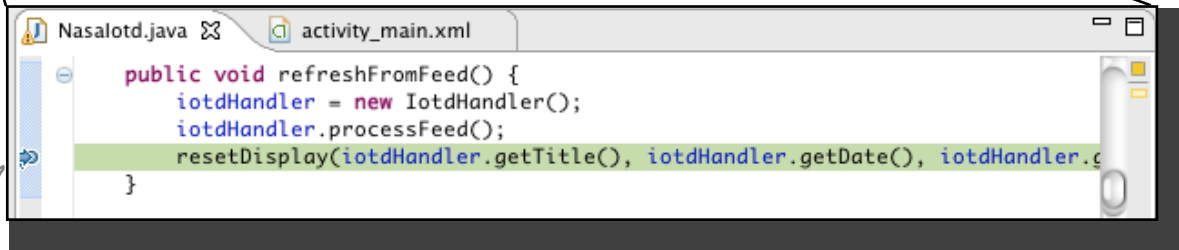
این انتخاب کننده از Java به debug تخییر میکنه و به تو اجازه میده

که بفهمی در پنجره خط زدا هستی. بر روی جاوا کلیک کن تا به جایی که لدهات هستند برسی.

این پنجره به تو thread stack traces را نشون میده.
(مترجم: در مورد Thread ها در آینده مفصل بحث خواهد شد)



این پنجره مقدار متغیرها را نشون میده.



این یه مکان که این طرف breakpoint هست به شما
نشون میده که کدام خط از برنامه داره اجرا میشه.

پس معلوم شد که برنامه به این خط از کد رسید خپ حالا کاربر چطور از این موضوع مطلع بشه ؟

یک progress dialog (نشان دهنده میزان پیشرفت) اضافه کن



Geek Bits

Modal به این معنی هست که کاربر نمیتونه هیچ تعاملی با برنامه داشته باشه و همه ورودی های کاربر نادیده گرفته میشن مثل کلیک (مترجم: در کتاب اصلی واژه modal به کار رفته بود که من اون را پنجره الکی ترجمه کردم)

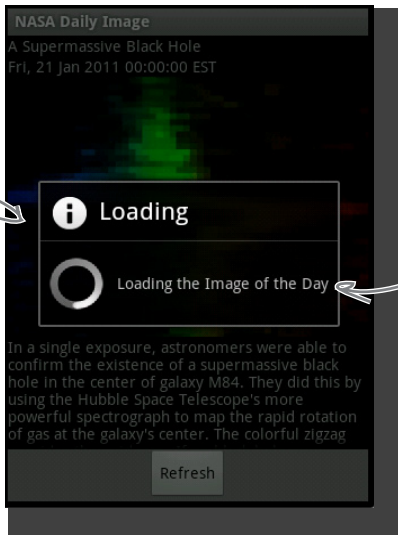
ProgressDialog یک پنجره الکی هست که یک هوایی ظاهر میشه و یک سری اطلاعات را نشون میده. ProgressDialog اینجا بدرد میخوره , چون میتونی به کاربرهات وضعیت برنامه را بگی و همچنین کاری میکنی که کاربرها مدام دکمه refresh را پست سر هم نزنن و مدام تابع مربوط به دکمه پشت سر هم اجرا نشه .

حالا این پنجره الکی ولی مفید را بطور نشون بدیم ؟

این کار با استفاده از فراخوانی تابع استاتیک یا ایستا show از کلاس ProgressDialog انجام میشه . تابع show یک ارجاع به نمونه ای از کلاس ProgressDialog برمیگردونه , تو باید این ارجاع را یک جایی ذخیره کنی چون که وقتی کارت با این پنجره تمام شد اون را ببندی .

این کدهای هست که progressDialog را نشون میده عنوان و متن را مطابق با نیاز خودت تغییر بده .

```
this,
"Loading",
"Loading the image of the Day");
```



Dismiss را بعد از dialog فراخوانی کن تا پنجره بسته بشه .

```
dialog.dismiss();
```

Call this to dismiss the dialog.

Sharpen your pencil



در زیر شما تابع `refreshFromFeed()` را میبینید که اجرای اون نیاز به زمان زیادی داره کدهای لازم را اضافه کن برای این که `Progressdialog` قبل از پردازش طولانی نشون داده بشه و و یادت نره که وقتی کارت تموم شد این پنجره را ازبین ببری .

```
public void refreshFromFeed() {
```

Show the dialog here. →

```
        iotdHandler = new IotdHandler();
        iotdHandler.processFeed();
        resetDisplay(iotdHandler.getTitle(),
        iotdHandler.getDate(),
                    iotdHandler.getUrl(),
                    iotdHandler.getDescription());
```

} The long-running work of the feed processing.

Dismiss the dialog here, now that all the work is done. →

```
}
```

Sharpen your pencil Solution

در زیر شما تابع refreshFromFeed() را میبینید که اجرای اون نیاز به زمان زیادی داره کدهای لازم را اضافه کن برای این که ProgressDialog قبل از پردازش طولانی نشون داده بشه و و یادت نره که وقتی کارت تموم شد این پنجره را ازبین ببری .

```
public void refreshFromFeed() {
```

```
    ProgressDialog dialog = ProgressDialog.show(  
        this,  
        "Loading",  
        "Loading the image of the Day");
```

Show the
progress
dialog.

```
        IotdHandler = new IotdHandler();  
        IotdHandler.processFeed();  
        resetDisplay(IotdHandler.getTitle(),  
                    IotdHandler.getDate(),  
                    IotdHandler.getUrl(),  
                    IotdHandler.getDescription());
```

The feed and
UI update
code remains
untouched.

```
        dialog.dismiss();
```

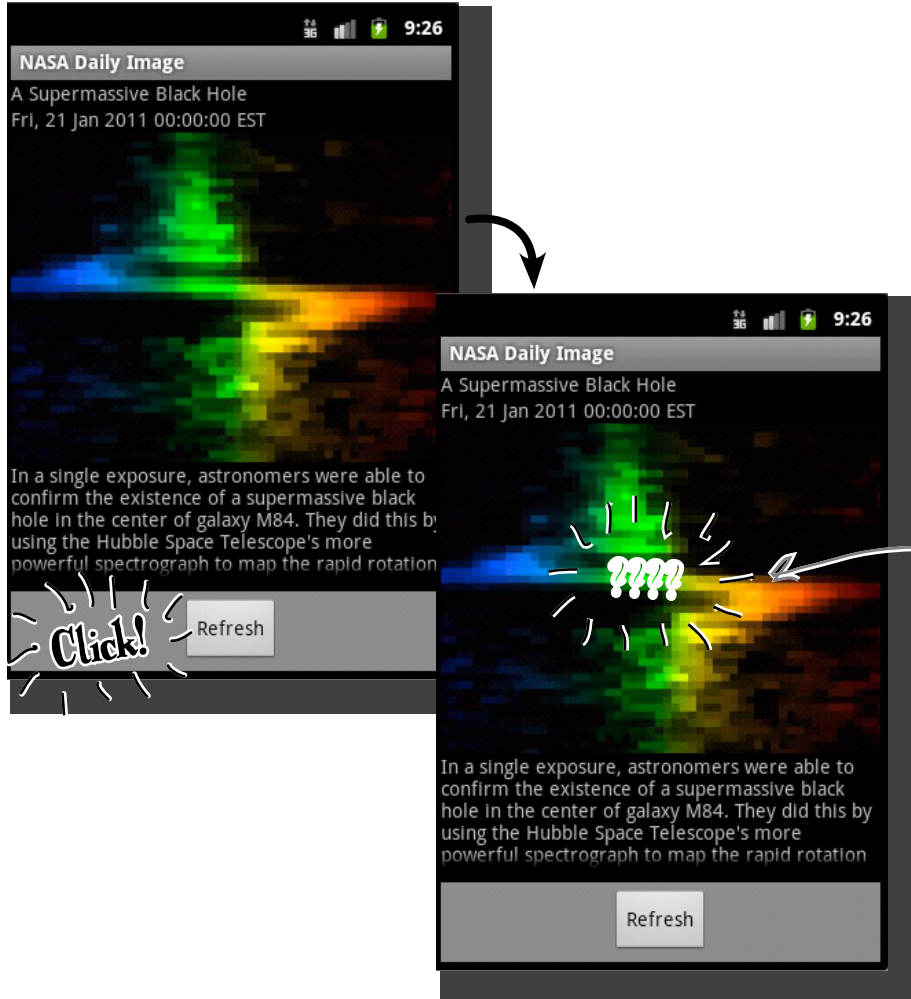
Dismiss the progress
screen, now that
the work is done.

```
}
```



Test Drive

برنامه را اجرا کن و روی refresh کلیک کن تا معلوم بشه Progressdialog کار میکنه .



خپ این اتفاق خوبی نیست

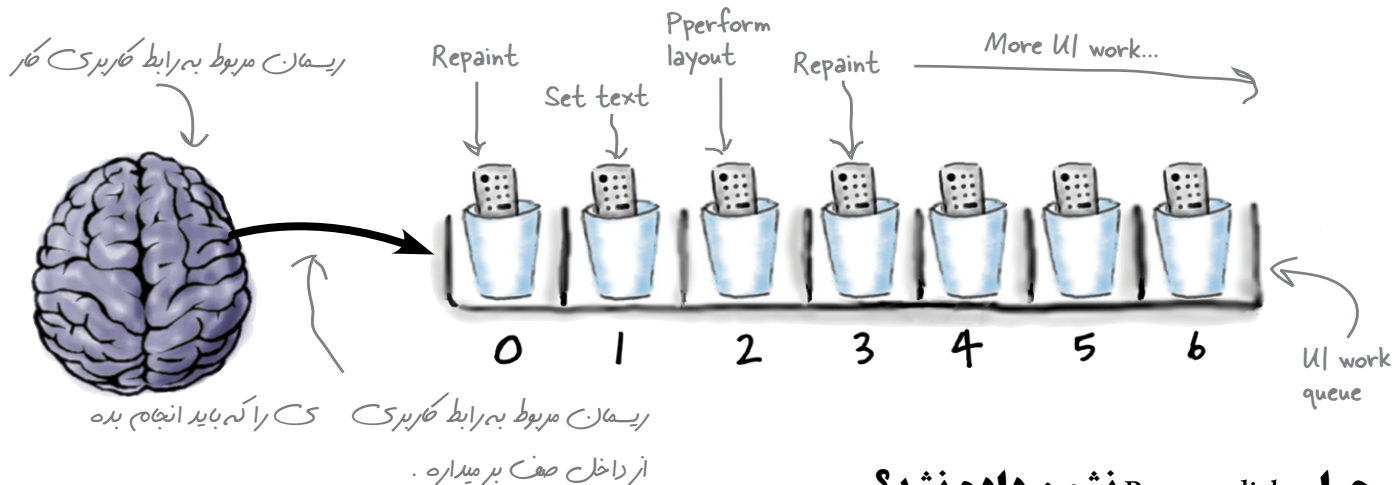
هدف ما از قرار دادن Progressdialog این بود که وقتی پردازش طولانی مربوط به پردازش فید اتفاق می افتد این پنجره نشون داده بشه . کدهای مربوط به این پنجره که در جای درست هستند , اما به دلایلی این پنجره نشون داده نمیشه . چه اتفاقی ممکنه افتاده باشه ؟

خپ باید بگم مشکل توی threading هست ...

Thread ای که مفهوس، رابط کاربری هست

(مترجم: thread یا ریسمان قابلیت است که سیستم عاملها و حتی CPU های جدید دارند و به این صورت هست که پردازشها به صورت یک ریسمان به پردازش گر که سیستم عامل میتونه بین این برنامه و پردازش گر باشه، عمل میکنن. یعنی با استفاده از ریسمانها شما میتونید چندین پردازش را با هم انجام دهید هم پردازشهای مربوط به یک برنامه و هم پردازشهای مربوط به برنامه های مختلف در ادامه تصویر واضح تری از ریسمان را پیدا خواهید کرد اما آگه نیاز به اطلاعات کاملتری دارید می تونید کتابی در مورد برنامه نویسی java یا c# بخوانید که من کتابهای Head first java programming and head first C# programming را که از همین انتشارات هست، توصیه میکنم.)

اندروید یک ریسمان مختص به اجرای رابط کاربری داره. این ریسمان کارش اینه که چیزهایی را روی صفحه بکشه مثل لایه ها و سایر عناصر گرافیکی که باعث میشه رابط کاربری را به این شکلی که میبینیم، باشه و همچنین انیمیشن ها به نرمی اجرا بشن. این ریسمان یک صف کاری داره که در این صف کارهایی که باید انجام بشن قرار میگیرن.



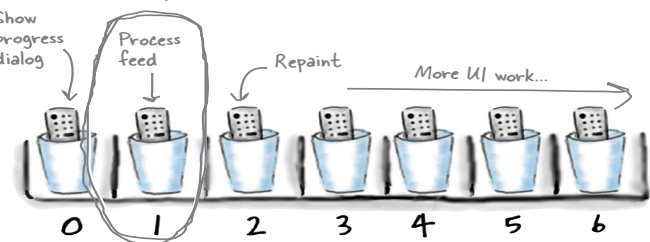
چرا Progressdialog نشون داده نشد؟

کاری که بعد از زدن روی دکمه انجام میشه در صف رابط کاربری قرار میگیره (یعنی این ریسمان مسول انجام این کار هست) زمانی که Progressdialog نشون داده میشه، مدام و پشت سر هم دستوراتی را فراخوانی میکنه (و باید فراخوانی کنه) تا انیمیشن مربوط به اون بر روی صفحه نشون داده بشه (منظور اون دایره هست که میچرخه) اما کد مربوط به فید هم در ریسمان مربوط به رابط کاربری اجرا میشه، که این به این معنی هست که زمان این ریسمان صرف انجام این پردازش میشه.

(مترجم: رون اجرای برنامه به این شکل هست که چون قرار هست پردازشها توسط یک ریسمان انجام بشوند همان طور که در تصویر میبینید و در کدهایی هم که نوشتیم به این شکل بوده کد مربوط به progressdialog اجرا میشه و شاید هم روی صفحه چیزی نمایش داده بشه اما به سرعت پردازش مربوط به فید آغاز میشه و کار progressdialog که repaint کردن مداوم صفحه است انجام نمیشه)

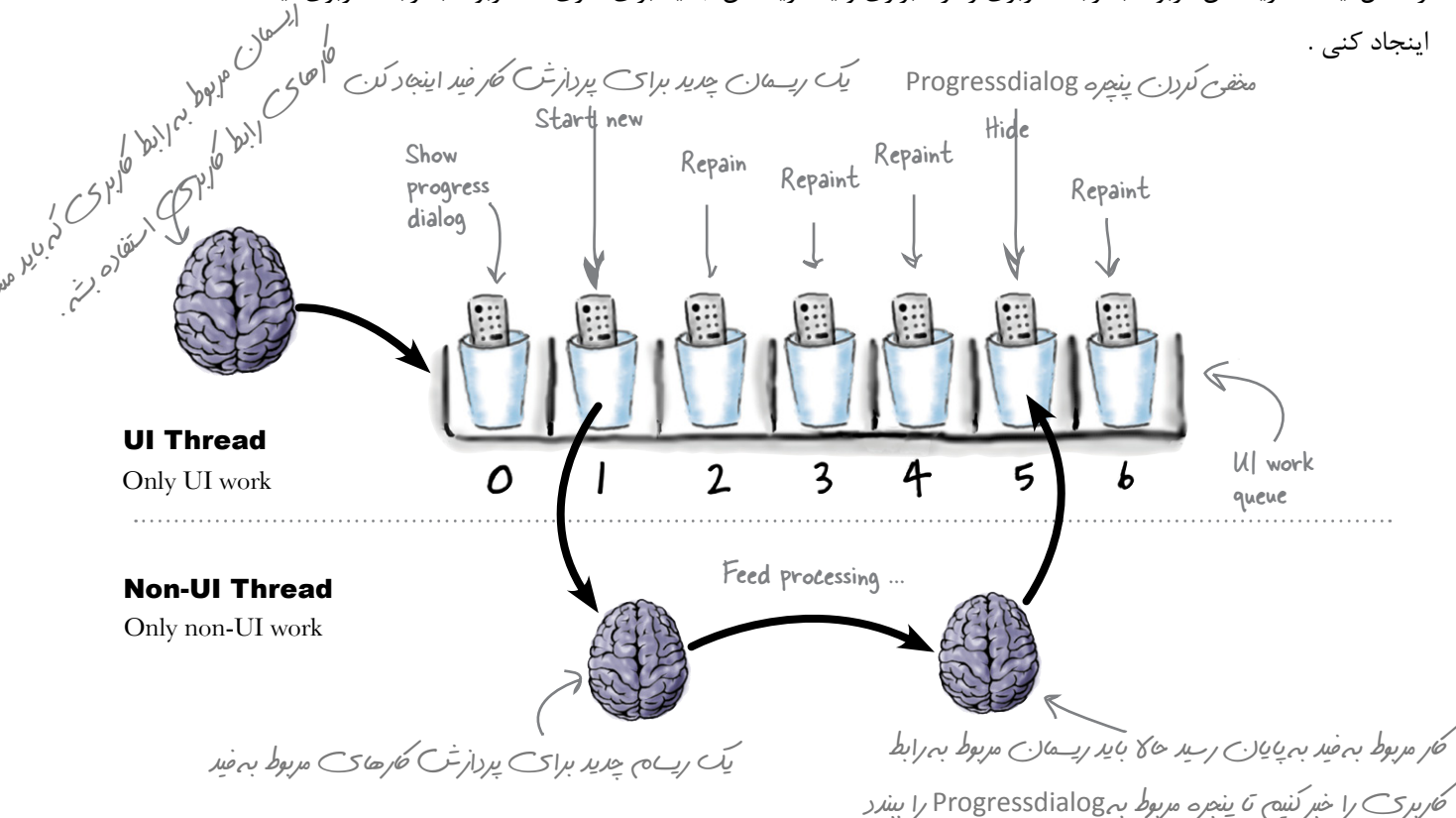
فراخوانی و اجرای این پردازش طولانی ریسمان مربوط به

تنها کاری که باید بکنیم اینه که یک ریسمان جدا برای پردازش مربوط به فید ایجاد کنیم رابط کاربری را از انجام سایر کارها باز میداره



می‌توانی بدونی چطور این مسئله را باید حل کنی؟

راه حل اینه که ریسمان مربوط به رابط کاربری را ازاد بزاری و یک ریسمان جدید برای کاری که مربوط به رابط کاربری نیست ایجاد کنی.



ریسمان رابط کاربری را از انجام پردازشهای سنگین و طولانی ازاد بزار تا رابط کاربری تو همیشه پاسخگو باشه.

انتقال دادن پردازش مربوط به فید از ریسمان رابط کاربری به ریسمان دیگه باعث میشه که ریسمان رابط کاربری روی نمایش انیمیشن تمرکز کنه. اولین دستور دوباره کشیدن یا repaint پنجره Progressdialog را نشان میدهد و دستورات بعدی repaint باعث ایجاد انیمیشن می شوند. و زمانی که کار پردازش فید به پایان رسید حاصل کار در صف کاری ریسمان رابط کاربری قرار داده میشه و باعث میشه که پنجره Progressdialog ناپدید بشه. عملیات تغییر از ریسمانی که کار پردازش فید را انجام میده به ریسمان رابط کاربری امر مهمی هست چرا که ریسمانهای دیگه نمی توانند عناصر این ریسمان را مستقیما دست کاری کنند.

یک ریسمان جدید برای پردازش طولانی ایجاد کن

بهترین راه برای این که یک ریسمان جدید ایجاد کنی و ریسمان رابط کاربری را از انجام پردازش طولانی خلاص کنی اینست که یک کلاس جدید ایجاد کنی که از کلاس پایه Thread ارث بری میکنه و در اون تابع run را پیاده سازی کنی.

ملیونها راه برای تعامل با ریسمانها هست هدف ما اینجا بحث بر سر اونها نیست فقط می خواهیم نشون بدیم که چطور با ریسمان رابط کاربری در اندروید کار کنیم.

```
public void refreshFromFeed() {
    dialog = ProgressDialog.show(
        this,
        "Loading",
        "Loading the Image of the Day");
}

Thread th = new Thread() {
    public void run() {
        if (iotdHandler == null) {
            iotdHandler = new IotdHandler();
        }
        iotdHandler.processFeed();
        resetDisplay(
            iotdHandler.getTitle(),
            iotdHandler.getDate(),
            iotdHandler.getUrl(),
            iotdHandler.getDescription());
    }
};
th.start();
```

Leave this code on the UI thread.

Extend thread.

Implement run.

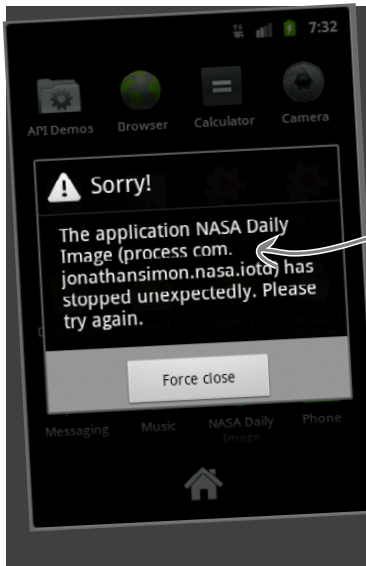
All of the feed-processing goes on the new thread.

Don't forget to start your new thread.



Test Drive

برنامه ات را دوباره اجرا کن و این بار که پردازش طولانی را به ریسمان دیگه ای محول کردی پنجره باید نمایش داده بشه
اما وقتی برنامه را اجرا کنی تو با یک خطا مواجه میشی .



You'll see this error dialog when you run the app.

توضیحات مربوط به خطای پیش آمده

```
FATAL EXCEPTION: Thread-11
android.view.ViewRoot$CalledFromWrongThreadException: Only the original thread that created a view hierarchy can touch its views.
    at android.view.ViewRoot.checkThread(ViewRoot.java:2932) at
    android.view.ViewRoot.requestLayout(ViewRoot.java:629) at
    android.view.View.requestLayout(View.java:8267)
```

مشکل کجاست ؟

مشکل در کد اخیری ، یعنی جایی هست که پنجره مجو میشه . مدیریت درست کار بر روی ریسمان مربوط به بابط کاربری فقط منحصر به واگذاری پردازش های سنگین به سایر ریسمانها نیست بلکه باید دقت کنید که همه کارهای مربوط به ریسمان بابط کاربری در خود این ریسمان انجام بشه .

```
iotdHandler.processFeed();

resetDisplay (
    iotdHandler.getTitle(),
    iotdHandler.getDate(),
    iotdHandler.getUrl(),
    iotdHandler.getDescription());
dialog.dismiss();
```

اینجا باید در ریسمان رابط کاربری اتفاق بیفتند

از Handler برای واگذاری کد به رابط کاربری استفاده کن

کد `dialog.dismiss()` باید به ریسمان رابط کاربری برگردد . کار واگذاری پردازش سنگین به ریسمان دیگه آسون بود اما اون ریسمان تازه ایجاد شده هیچ ارجاعی به ریسمان رابط کاربری نداشت تا بتونه بعد از پایان کار , کدهای رابط کاربری را ایجاد کنه . در اینجاست که handler استفاده میشه .

Handler یک ارجاع به ریسمانی که اون را ایجاد کرده در خودش نگه میداره . تو میتونی یک کاری را به Handler واگذار کنی و مطمئن باشی که این کار توسط همون ریسمانی انجام میشه که این handler در اون ساخته شده یا به عبارت بهتر نمونه سازی شده .

نمونه سازی یک handler در ریسمان رابط کاربری

تابع `onCreate()` در ریسمان رابط کاربری فراخوانی میشه . اگه تو یک handler را در این تابع نمونه سازی کنی اون وقت هر وقت که بخوای میتونی کارهات را با استفاده از این handler , روی ریسمان رابط کاربری انجام بدی .

```
Handler handler;

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    handler = new Handler();
    refreshFromFeed();
}
```

یک متغیر از کلاس handler ایجاد کن و هر وقت که خواستی میتونی نمونه ساخته شده را به اون ارجاع بدی .

تابع `onCreate()` روی ریسمان رابط کاربری اجرا میشه

چون تابع `onCreate()` در داخل ریسمان رابط کاربری ایجاد میشه , ساختن یک handler در اون این قابلیت را با اون میده که هر وقت خواستی میتونی کدهات را در ریسمان رابط کاربری اجرا کنی

Pass work to the Handler using post

Once you have a Handler instance, you can call `post`, passing it a Runnable to execute on the desired thread.

```
handler.post(Runnable runnable)
```

This is a standard Runnable, nothing Android specific.

Get ready to fix `refreshFromFeed()` with correct threading...



از قطعات زیر استفاده کن تا تابع `refreshFromFeed()` را بعد از تغییراتی که در ریسمانها دادی درست کنی . پردازش سنگین فید را به ریسمان دیگه ای ببر و کدهای مخفی کردن پنجره را با استفاده از `handler` در ریسمان رابط کاربری اجرا کن . فرض کن که `handler` تو در متد `onCreate()` نمونه سازی شده .

Here are your magnets.

```
resetDisplay(iotdHandler.getTitle(),  
            iotdHandler.getDate(), iotdHandler.getUrl(),  
            iotdHandler.getDescription());
```

```
handler.post({
```

```
    dialog.dismiss();
```

```
dialog = ProgressDialog.show(this,  
    "Loading", "Loading the Image of the Day");
```

```
new Runnable () {
```

```
    public void run() {
```

```
        public void run() {
```

```
        }});
```

```
Thread th = new Thread() {
```

```
    th.start();
```

```
        iotdHandler.processFeed();
```

```
        if (iotdHandler == null) {  
            iotdHandler = new IotdHandler();  
        }
```

Handler Magnet Solution



از قطعات زیر استفاده کن تا تابع `refreshFromFeed()` را بعد از تغییراتی که در ریسمانها دادی درست کنی . پردازش سنگین فید را به ریسمان دیگه ای ببر و کدهای مخفی کردن پنجره را با استفاده از `handler` در ریسمان رابط کاربری اجرا کن . فرض کن که `handler` تو در متد `onCreate()` نمونه سازی شده .

پنجره `ProgressDialog` در خود ریسمان رابط کاربری

فراخوانی میشه

```
dialog = ProgressDialog.show(this,
    "Loading", "Loading the Image of the Day");
```

```
Thread th = new Thread() {
```

```
public void run() {
```

```
if (iotdHandler == null) {
    iotdHandler = new IotdHandler();
}
```

```
iotdHandler.processFeed();
```

```
handler.post(
```

```
new Runnable () {
```

```
public void run() {
```

```
resetDisplay(iotdHandler.getTitle(),
    iotdHandler.getDate(), iotdHandler.getUrl(),
    iotdHandler.getDescription());
```

```
dialog.dismiss();
```

```
}});
```

```
th.start();
```

یک ریسمان جدید برای فید آغاز کن

این `Runnable` جدید را به `handler` بفرست

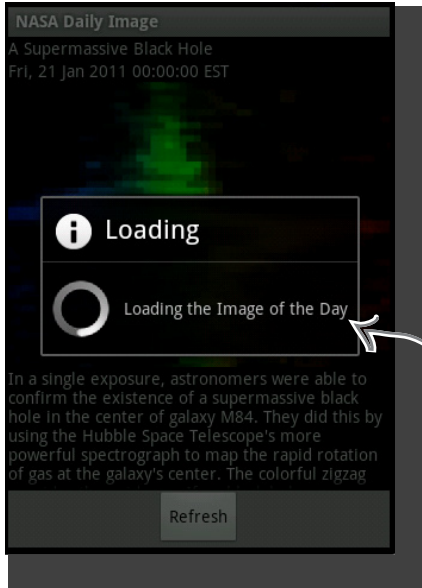
در اینجا `resetDisplay` and `dismiss` را با استفاده از `handler` در ریسمان رابط کاربری اجرا کن



Test Drive

حالا برنامه را اجرا کن و می بینی که وقتی تابع onCreate() فراخوانی میشه progressdialog هم به نمایش در میاد همچنین اگه روی دکمه refresh کلیک کنی این پنجره را دوباره میبینی.

1 شروع برنامه



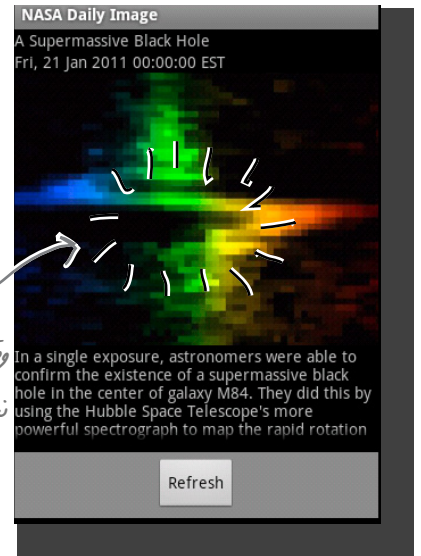
به برنامه چند ثانیه فرصت بده تا فید را بارگذاری کنه

نگاه کن که پنجره ProgressDialog جطور ناپدید میشه



On app startup, the progress dialog will show

وقتی پردازش فید کامل شد ریله اثری از Progressdialog نخواهد بود

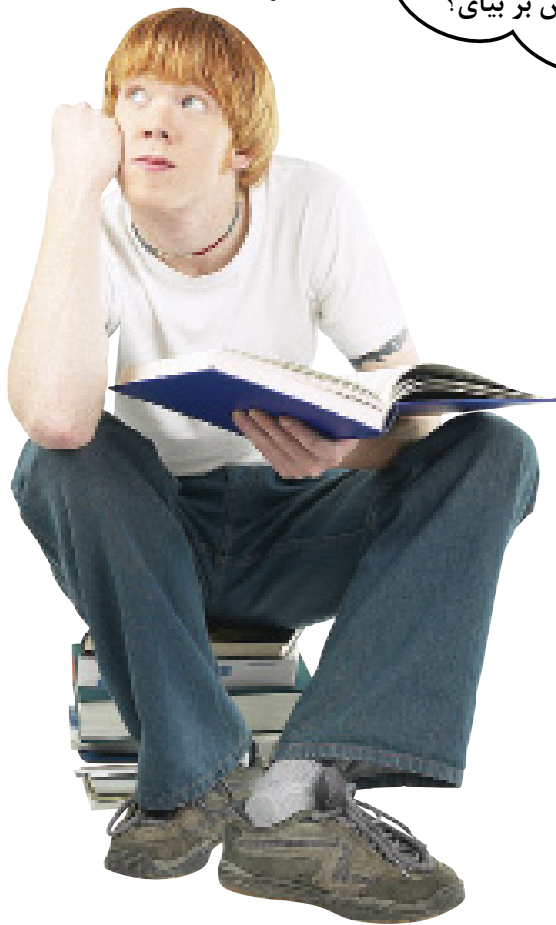


کارت عالی بود!

حالا کاربرها میدونن که برنامه داره کاری انجام میده .

Positive reinforcement goes a long way! (مترجم: به نظر شما معنی این چی میشه؟)

دیدن عکس روز ناسا خیلی با حاله ... اما زود گذره .
من دوست دارم که عکسی را که دوست دارم بتونم به
عنوان عکس پس زمینه ذخیره کنم , در نتیجه بعدا هم
میتونم اون را تماشا کنم . فکر میکنی از پیشش بر بیای؟

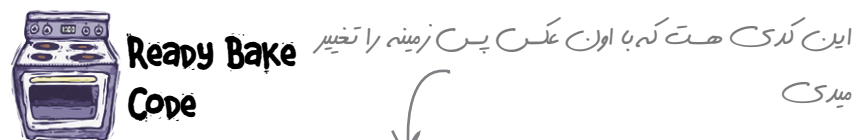


نباید زیاد سخت باشه .

عوض کردن عکس پس زمینه میتونه سریع انجام بشه .
تو الان عکس را در اختیار داری تنها کاری که باید بکنی
اینه که اون را به عنوان عکس پس زمینه قرار بدی همچنین
تو یک پانل دکمه داری و تنها کاری که باید بکنی اینه که
دکمه دیگه ای به اون اضافه کنی .

کد

تو این کار را میتونی با بازیابی کردن WallpaperManager و قرار دادن عکس پس زمینه با استفاده از Bitmap انجام بدی. تو الان یک ارجاع به Bitmap از فید خودت داری .



```
WallpaperManager wallpaperManager =  
    WallpaperManager.getInstance(this);  
wallpaperManager.setBitmap(bitmap);
```

تو میتونی عکس که از فید گرفتی را در این جا وارد کنی

کلمه this به آنتیوتی که در حال اجرا هست اشاره داره



طرح

تو تا الان یک خونه برای دکمه تازه سازی ساختی و تنها کاری که باید انجام بدی اینه که دکمه جدیدت را هم در این خونه جا بدی .

بابی حتما از این خوشش میاد! پس بیا شروع کنیم

دکمه set wallpaper را اضافه کن

پانل دکمه توسط بک LinearLayout ساخته شده پس تو میتونی دکمه جدیدت را مستقیما به اون اضافه کنی .
LinearLayout ها به صورت پیش فرض افقی هستند اما تو میتونی برای محکم کاری
android:orientation="horizontal" را به اون اضافه کنی یا به حالت پیش فرض اعتماد کنی .
دکمه جدید را به پانل دکمه در main.xml اضافه کن .

```
<LinearLayout
    android:orientation="horizontal" ←
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:gravity="center_horizontal"
    android:background="#ff8D8D" >

    <Button android:text="@string/refresh"
        android:onClick="onRefresh"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <Button android:text="@string/setwallpaper"
        android:onClick="onSetWallpaper"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

LinearLayout defaults to horizontal orientation, but it's a good idea to include the orientation attribute anyway. It makes your code easier to understand later and proprotects you in case defaults change.

دکمه set wallpaper را به عنوان زیر شمشیر فرزند دوم این دکمه اضافه کن . این دکمه در سمت راست دکمه قبلی قرار میگیرد

همچنین منبع string.xml را هم برای دکمه جدید بروز رسانی کن :

```
<string name="setwallpaper">Set Wallpaper</string>
```


برای فعالیت دکمه جدید اکتیویته خودت را بروزرسانی کن

فید ما عکس را از اینترنت میگیره و یک `Bitmap` میسازه. برای کامل کردن `onsetWallpaper` (که تابع کلیک شدن روی دکمه `setWallpaper` هست) `Bitmap` را بگیر و اون را به `WallpaperManager` بده.

```
public class NasaIotd extends Activity {
    private IotdHandler iotdHandler;
    ProgressDialog dialog;
    Handler handler;
    Bitmap image;
}
```

یک متغیر از `Bitmap` بنر

```
iotdHandler.processFeed();
image = getBitmap(
    iotdHandler.getUrl());
```

In refreshFromFeed()

تابع `onSetWallpaper` را به اکتیویته در `Nasalotd.java` اضافه کن.

```
public void onSetWallpaper(View view) {
    Thread th = new Thread() {
        public void run() {
            try { WallpaperManager wallpaperManager =
                WallpaperManager.getInstance(NasaIotd.this);
                wallpaperManager.setBitmap(image);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    };
    th.start();
}
```

*عوض کردن پس زمینه ممکنه یک مدت طول بکشه
بنابراین مثل قبل یک ریسمان جدید برای پردازش اون
ایجاد کن.*

Since the current scope is an inner class; you can get a reference to "this" by preceding it with the class name.

This will do a default dump of the exception to LogCat.



Test Drive

برنامه را اجرا کن و مطمئن شو که دکمه درست در لایه قرار داده شده .



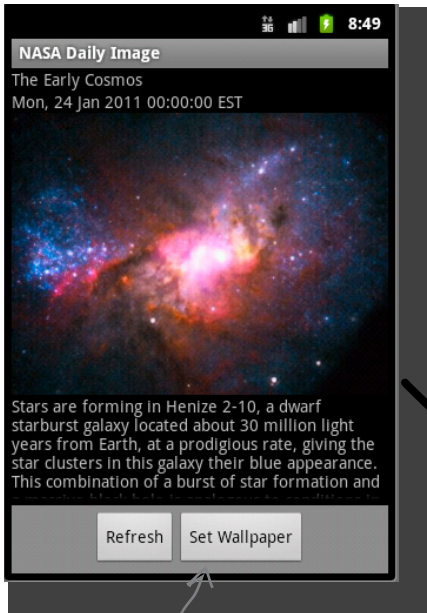
اول بررسی کن که دکمه درست سر جاش باشه

عوض کردن پس زمینه نیاز به کتب مجوز داره .
با نوشتن کد

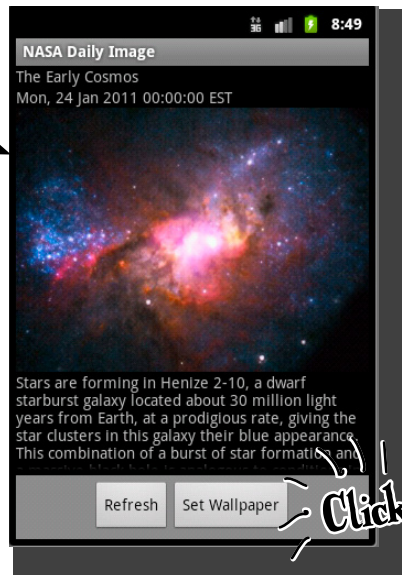
```
Android.permission.SET_WALLPAPER
```

در فایل AndroidManifest.xml مجوز لازم را بدست بیار .

دکمه که درست سر جاش هست حالا ببین که چطور کار میکنه ...



The button is displaying correctly, horizontally positioned next to the Refresh button.



چرا؟ هیچ اتفاق نیفتاد. من روی دکمه کلیک کردم و برنامه
همین طور ثابت مونده!



در حقیقت دکمه کار کرد اما ...

اگر تو همین الان به صفحه اصلی برگردی میبینی که عکس پس زمینه
تغییر کرده. اما این نشون میده که تعامل برنامه ات با کاربران افتضاحه.
به یاد داشته باش که کار کردن برنامه ات فقط بخشی از یک پازل بزرگ هست
برای این که برنامه ای موفق بسازی که مردم ازش استفاده کنن (و این باعث میشه
که جیبهای تو هم پر از پول بشن) باید تعامل کاربران با برنامه ات خوب باشه.
مشکل اینجا اینه که عکس پس زمینه دور از چشم کاربران عوض میشه.
تنها چیزی که نیاز داری یک تقویت مثبت یا عکس العمل مثبت به کاربران هست
تا اونها بدونن که برنامه کار کرده.

تو میتونی یک Progressdialog زمانی که عکس داره تغییر میکنه نشون بدی

اما یک راه بهتری هم هست

روی دکمه مربوط به صفحه اصلی یا خانه کلیک کن و ببین که

عکس تغییر کرده



از toast استفاده کن تا به کاربران اطلاع بدی

Toast

= passive
notifications

Progress Dialog

= active, blocking
notifications

تو میتونی از ProgressDialog استفاده کنی. اما یکی از خصوصیات به ارث برده توسط این پنجره اینه که ورودی های کاربر نادیده گرفته میشن, این برای زمانی که فید در حال پردازش بود, خوب بود چون کاربر را از فشار دادن مداوم دکمه تازه سازی منع میکرد.

اما تغییر عکس پس زمینه فرق میکنه: توفقط میخوای که کاربر را از تغییر عکس مطلع کنی, نه این که اون را از انجام کاری منع کنی. برای مثال وقتی کاربر دکمه را میزنه اون میتونه به پایین بره و توضیحات عکس را در مدتی که عکس تغییر میکنه مطالعه کنه (مترجم: کار تغییر عکس خیلی طول نمیکشه ژ فکر کنم بیشتر از یک ثانیه نشه, پس کاربر باید ابر تند خوان باشه که توی این زمان کم بتونه این همه متن را بخونه, در هر حال شاید دلیل کتاب جالب نباشه اما استفاده از toast عاقلانه ترین کاره).

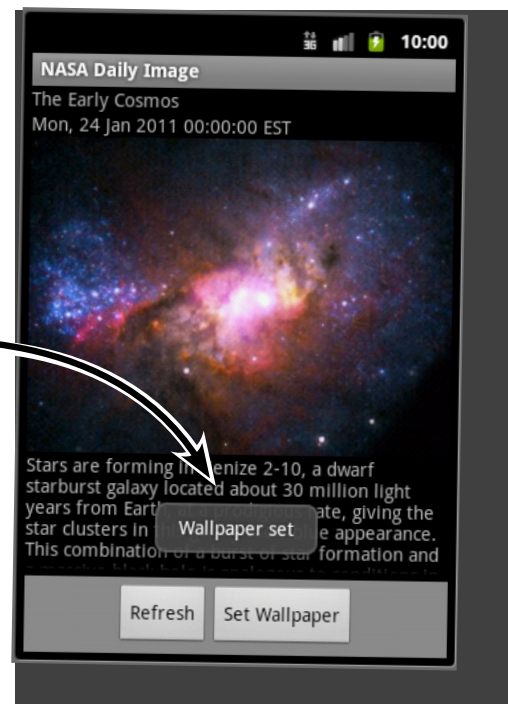
اندروید toast را برای همین مواقع پیش بینی کرده

Toast یک اعلام دهنده خنثی هست که کاربر را بلوک نمی کنه و تنها یک پیغام را در زیر صفحه نمایش میده. این پیغام تنها چند ثانیه نشون داده میشه و بعد ناپدید میشه و کاربر کاملاً میتونه با برنامه تعامل داشته باشه. در اینجا شما کدی را همراه با لحظه ای که این پیغام نمایش داده میشه میبینید.

```
Toast.makeText(this,  
    "Wallpaper set",  
    Toast.LENGTH_SHORT).show();
```

منظور از this آنتیویج حال حاضر هست
آنتیویج خودت را بهش بده

Time to display
the toast.





Exercise

متد `onSetWallpaper` را با اضافه کردن دو تا `Toast` کامل کن ، یکی برای موفقیت امیز بودن و دیگری برای عدم موفقیت ، در بلوک `catch` . اطلاع رسان `Toast` باید در ریسمان رابط کاربری پردازش شود . از `Handler` ای که قبلا ساختی استفاده کن تا هر دو `Toast` را در ریسمان رابط کاربری پردازش کنی .

```
public void onSetWallpaper(View view) {  
    Thread th = new Thread() {  
        public void run() {  
            WallpaperManager wallpaperManager =  
                WallpaperManager.getInstance(NasaIotd.this);  
            try {  
                wallpaperManager.setBitmap(image);
```

*Add the code here
to create the toast
confirmation message for
setting the wallpaper.*



```
        } catch (Exception e) {  
            e.printStackTrace();
```

*Add a toast message in
the catch block with the
message "Error setting
wallpaper."*



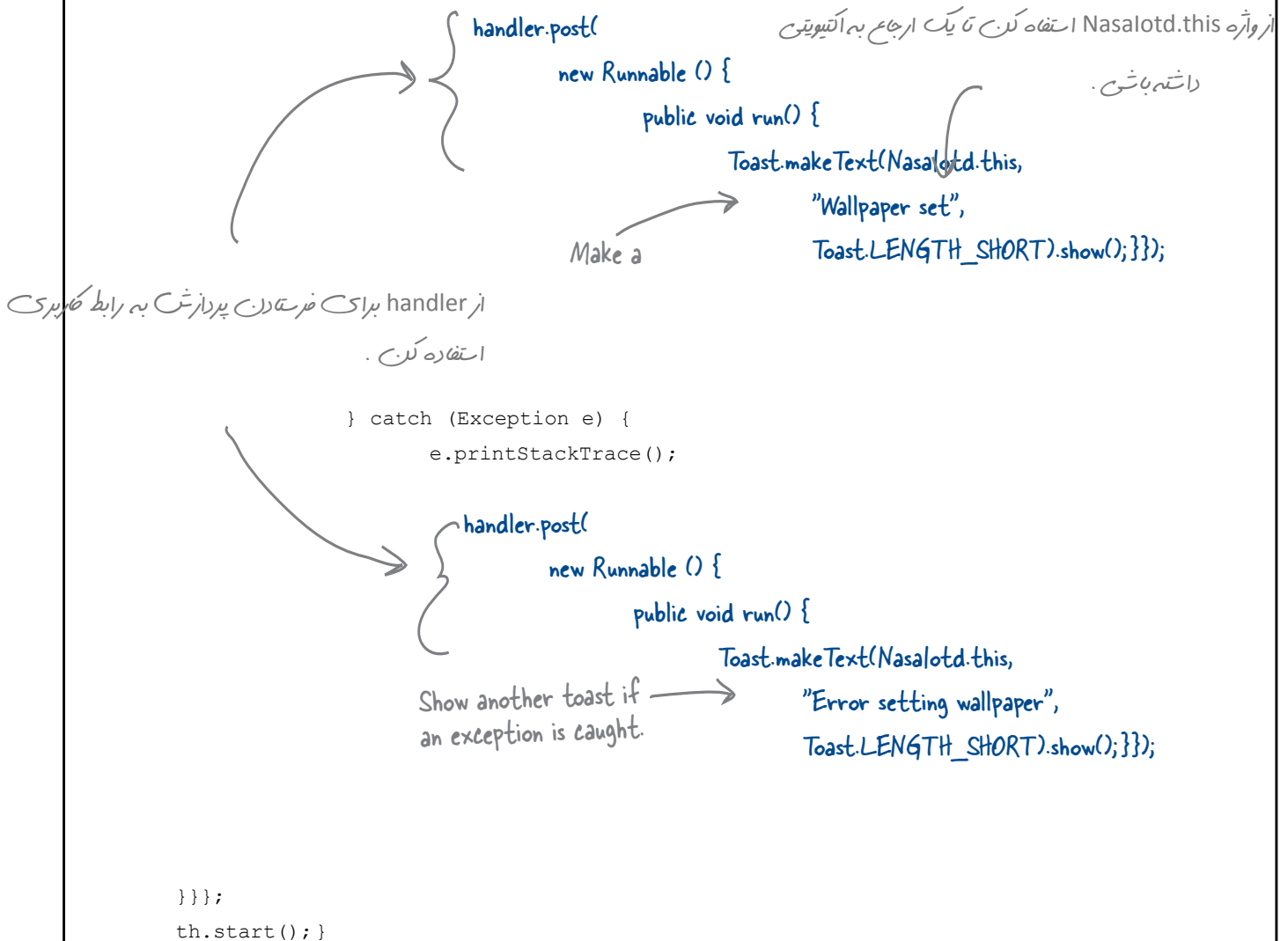
```
    }  
    }  
    th.start();  
}
```



Exercise Solution

متد `onSetWallpaper` را با اضافه کردن دو تا `Toast` کامل کن ، یکی برای موفقیت امیز بودن و دیگری برای عدم موفقیت ، در بلوک `catch` . اطلاع رسان `Toast` باید در ریسمان رابط کاربری پردازش شود . از `Handler` ای که قبلا ساختی استفاده کن تا هر دو `Toast` را در ریسمان رابط کاربری پردازش کنی .

```
public void onSetWallpaper(View view) {
    Thread th = new Thread() {
        public void run() {
            WallpaperManager wallpaperManager =
                WallpaperManager.getInstance(NasaIotd.this);
        }
    };
    try {
        wallpaperManager.setBitmap(image);
    }
}
```

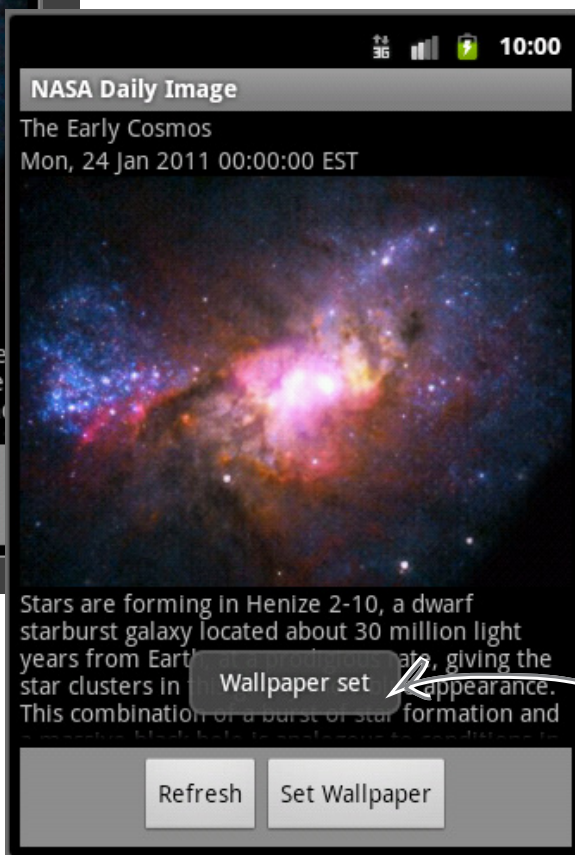
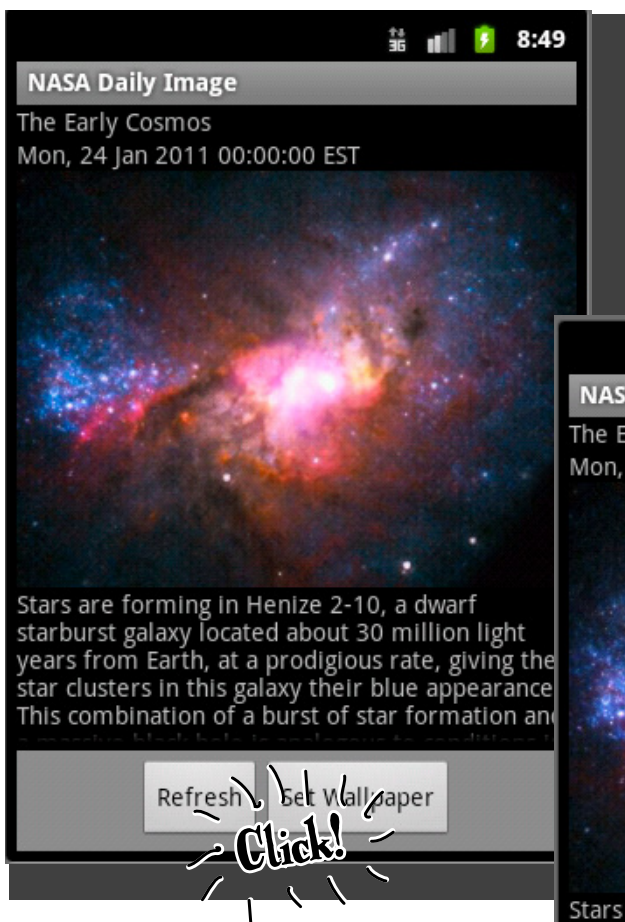




TEST DRIVE

برنامه را اجرا کن و روی دکمه Set Wallpaper کلیک کن . حالا میبینی که پس زمینه تغییر کرد و اطلاع رسان

Toast نمایش داده شد و این اجازه میدهد که کاربر از اتفاقات مطلع بشه .



The toast confirmation displays shortly after the click.

کارت عالی بود! بابی و همه دوستاش از این خوششون میاد!

برنامه کار میکنه... تو کاملا انجامش دادی! حالا فلتر می‌کنم

و قش ریده تا این برنامه را با مردم همی بشم. بینم

تو میتونی اون را روی فروشگاه بزاری؟



به نظر خوبه! ایستگاه بعد... فروشگاه اندروید!

تو در فصل بعدی این برنامه را روی فروشگاه اندروید قرار

میدی، پش با ما همون!



ANDROID

market

با استفاده از ریسمانها و باز خورد کاربران

تو میتونی به کاربرهات قول یک برنامه

قوی و کاربر پسند را بدی .



BULLET POINTS

از خصوصیات اضافی Linearlayout

استفاده کن تا صفحه ات را به خوبی

تنظیم کنی (padding , margin,background)

عرض و ارتفاع لایه را با مقادیر

Fill_parent and wrap_content

مقدار دهی کن , fill_parent باعث میشه لایه

اونقدر گش بیاد تا لایه بالای را کاملا پر کنه

و wrap_content باعث میشه که لایه فقط به

همان اندازه که نیاز داره کش بیاد و بزرگ بشه .

از واحد DIP برای زمانی که اندازه ای را می خواهی

تعریف کنی استفاده کن . این به تو این اطمینان را

میده که برنامه تو روی اکثر دستگاهها کارمیکنه .

لایه ها میتونن در هم دیگه قرار بگیرن یعنی یک

لایه را به عنوان View به لایه دیگر اضافه کنی

اما دقت کن که افراط در این کار باعث میشه

سرعت پاسخگویی لایه و مدت زمان رندر کردن

اون طول بکشه . پس مراقب باش(در مورد راههای

دیگه بعدا یاد میگیری)

از خطا زدا برای ردگیری کدهات در شبیه ساز

یا حتی دستگاه واقعی استفاده کن .

از ProgressDialog برای بلوک کردن و

نادیده گرفتن ورودی های کاربر استفاده کن

از Toast استفاده کن تا به طور خنثی به کاربر

اطلاع بدی (بدون بلوک کردن ورودی ها)

هم Toast و هم ProgressDialog میتونن

توسط کاربر سفارشی (customized) بشن.

پردازشهای سنگین را از ریسمان رابط کاربری

بردار کار های این ریسمان را فقط به خودش

محول کن .

از Handler استفاده کن تا کارهای رابط کاربری

را از سایر ریسمانها به ریسمان رابط کاربری منتقل

کنی .

ریسمان رابط کاربری

پردازشهای سنگین را از دوش ریسمان

رابط کاربری بردار در غیر این صورت

رابط کاربری خوبی نخواهی داشت .

مطمئن شو که همه کارهای مربوط به رابط

کاربری فقط بر روی ریسمان رابط کاربری

انجام میشه در غیر این صورت

این کار باعث ایجاد یک استثا میشه .

به کاربرهات باز خورد بده

از Toast استفاده کن تا پیام ها به صورت خنثی

نمایش داده شوند .

از ProgressDialog زمانی استفاده کن که

میتوانی ورودی کاربرها نادرده گرفته بشه و یک

پیغام را به اونها نشون بدی .