Android The Rocket App

Getting data from the internet **List Views**

cs380 Nov 19

The Rocket Database App

- Every rocket launch from 1957 June 2020
- Goal
 - provide a user friendly portal into querying this database
 - Or at least a subset of the database
- Idea
 - allow users to look at the data starting from different slicing approaches
 - I have only implemented two slices

Tables_in_rocket

launch site vehicle

describe launch;

	Field Typ		ре	Null		Ke	Key D		efault	E×	<t< th=""></t<>
	Tag JD Date Vehicle Flight Mission LaunchSite LaunchPad Apogee Category	va va va va va va va va va	rchar(10) rchar(12) te rchar(20) rchar(20) rchar(20) rchar(10) diumint(9) rchar(10)	har(10)NOPRINULLhar(12)NOPRINULLYESNULLNULLhar(20)YESMULNULLhar(30)YESNULLhar(10)YESMULNULLhar(10)YESMULNULLhar(10)YESNULLhar(10)YESNULLhar(10)YESNULL							
-	Field		+ Туре		⊦ Nu]	+ ll	 Ke	+ ey	Defau	+ lt	
	<pre>Name Name FamilyName MainStageName Manufacturer VehicleVariant Length Diameter LaunchMass LEOPayload GTOPayload Thrust MaxApogee</pre>		varchar(20) varchar(20) varchar(20) varchar(10) varchar(20) float float float float float float float float		NO YES YES YES YES YES YES YES		PF	RI	NULL NULL NULL NULL NULL NULL NULL NULL		



Slice / App navigation



This feels to me like a natural fit for fragment navigation, and for use of the android system back button

List of Launches

Use the Android back button to reverse course

Rocket Details

Getting data from internet

Just write some standard java code to do this see "webInvokeJSON" in starter

NetworkSecurityConfig: No Network Security Config specified, using platform default E/THISS: android.os.NetworkOnMainThreadException android.os.NetworkOnMainThreadException

Into Android Manifest:
<pre><uses-permission android:name="android.permission.INTERNET"></uses-permission></pre>
android:usesCleartextTraffic="true"

With permissions fixed:

E/THISS: android.os.NetworkOnMainThreadException android.os.NetworkOnMainThreadException at android.os.StrictMode\$AndroidBlockGuardPolicy.onNetwork(StrictMode.java:1450) "Uses-permission" is fairly standard in android. Any time you use a subsystem, you have to ask for permission. Allows the user to say no — for instance, "no the app can't use the camera".

Allows http connections. Android would far prefer https

DBs can be set up to allow direct connections. The DB on comet does not allow connections except from localhost. So, need to write some PHP. PHP or direct to DB, you still need this



Main Thread??

- ""When an application is launched in Android, it creates the first thread of execution, known as the "main" thread. The main thread is responsible for dispatching events to the appropriate user interface components as well as communicating with components from the Android UI toolkit.""
 - So, the main thread is what users interact with and see
 - The goal is to always have the main thread be responsive to the user. Even if sometimes that means just telling the user "wait a second"

• Network operations are NOT allowed on the main thread

Running away from the Main Thread

```
Reusing webInvokeJSON. These
 Again, just plain Java
                                                           params are meaningless
   • Create a class that implements Runnable
                                                                     Use Java Executor rather than just creating a
   • Execute that class in a separate thread
                                                                        thread on the fly ... executors do thread
                                                                                       pooling etc.
private class GetLaunchersRunner implements
                                                               Executor exec = new Executor() {
Runnable {
                                                                   @Override
                                                                   public void execute(Runnable command) {
    @Override
                                                                       new Thread(command).start();
    public void run() {
               final String[][] ee =
                                                               };
                {"ActionPageSize", "1"},
                {"ActionPageNo", "1"},
                                                               try {
        };
                                                                   exec.execute(new GetLaunchersRunner());
        String res = webInvokeJSON("http://
comet.cs.brynmawr.edu/~gtowell/380/rocket/
                                                               catch (RejectedExecutionException | NullPointerException
sounders.php", ee);
                                                               exception) {
        Log.i("THISS", res);
                                                                   Log.e("THISS", "Thread problem", exception);
```

Wait a second (or 10) or for something to complete

```
final int delay = 10;
final RelativeLayout rl = findViewById(R.id.mainlayout);
final TextView tv = new TextView(this);
tv.setText(""+delay);
tv.setGravity(Gravity.CENTER);
tv.setBackgroundColor(Color.MAGENTA);
tv.setTextSize(128);
tv.setTypeface(null, Typeface.BOLD);
ViewGroup.LayoutParams lpp = new
ViewGroup.LayoutParams(ViewGroup.LayoutParams.MATCH PARENT,
ViewGroup.LayoutParams.MATCH PARENT);
rl.addView(tv, lpp);
tv.setRotation(0);
tv.animate().rotation(2880).setDuration(delay*1000).setListener(new
AnimatorListenerAdapter() {
    @Override
    public void onAnimationEnd(Animator animation) {
        super.onAnimationEnd(animation);
});
```

This example is fairly fake because you usually have no idea how long you will have to wait. So, you put up a "wait a minute" sign, then take it down when something completes. No fancy countdown.

Counterexample: some downloads you know the size (from http header) and know how much you have gotten ...

Starter



Updating the UI

```
private class GetLaunchersRunnerV2 implements Runnable {
    @Override
    public void run() {
        final String[][] ee = {
                {"ActionPageNo", "1"},
        };
        String res = webInvokeJSON("http://comet.cs.brynmawr.edu/~gtowell/380/rocket/
sounders.php", ee);
        TextView tv = (TextView)MainActivity.this.findViewById(R.id.textview);
        tv.setText(res);
```

E/AndroidRuntime: FATAL EXCEPTION: Thread-2

Process: edu.brynmawr.starter, PID: 31639

android.view.ViewRootImpl\$CalledFromWrongThreadException: Only the original thread that created a view hierarchy can touch its views.

at android.view.ViewRootImpl.checkThread(ViewRootImpl.java:7313)

at android.view.ViewRootImpl.requestLayout(ViewRootImpl.java:1161)

• Only the UI thread can update the UI thread!!

				9:53	
Rockets!!!					
name	cou nt	first	recent		
Ariane 5G	16	1996-06-04	2003-09-27	<u>details</u>	
Atlas D	135	1959-04-14	1967-11-07	<u>details</u>	
Atlas E	48	1960-10-11	1995-03-24	<u>details</u>	
Atla C Open app again Atla					
Atla	0	pen app aga	in		
Atla Chang Zheng 2C	0 57	pen app aga 1975-11-26	in 2020-06-10	details	
Chang Zheng Chang Zheng Chang Zheng D	0 57 46	pen app aga 1975-11-26 1992-08-09	in 2020-06-10 2020-05-31	<u>details</u> <u>details</u>	
Atla Chang Zheng 2C Chang Zheng 2D Chang Zheng 2F	0 57 46 12	pen app aga 1975-11-26 1992-08-09 1999-11-19	in 2020-06-10 2020-05-31 2016-10-16	details details details	
Atla Chang Zheng 2C Chang Zheng 2D Chang Zheng 2F Chang Zheng 4B	0 57 46 12 35	pen app aga 1975-11-26 1992-08-09 1999-11-19 1999-05-10	in 2020-06-10 2020-05-31 2016-10-16 2019-12-20	details details details details details	
Chang Zheng Chang Zheng Chang Zheng Dhang Zheng F Chang Zheng B Chang Zheng IB	0 57 46 12 35 27	pen app aga 1975-11-26 1992-08-09 1999-11-19 1999-05-10 2006-04-26	in 2020-06-10 2020-05-31 2016-10-16 2019-12-20 2019-11-27	details details details details details details	

UI updates must be from the UI thread

• So you need to get back there

This gets you to the main (UI) thread and executes code there

private class GetLaunchersRunnerV3 implements Runhable /{



The text is too big!! Scrolling

- When everything does not fit, wrap it in a ScrollView.
 - Usually a good idea to do so because you can never be sure that someone does not have a tiny screen

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"</pre>

xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<ScrollView

android:layout_width="match_parent" android:layout_height="match_parent" android:fadeScrollbars="false" tools:layout_editor_absoluteX="200dp" tools:layout_editor_absoluteY="299dp">

<LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:orientation="vertical" >

<TextView android:id="@+id/textview" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Hello World!" /> </LinearLayout> </ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

▼⊿ 🛔 11:38

Starter

🏟 🏐 🦱

1973-01-01 , recent : 2006-04-22 },{ name : Loki

Dart","count":"1565","first":"1956-05-09","recent":"1985-12-30"}, {"name":"M-100","count":"5805","first":"1957-07-11","re cent":"1986-12-01"},{"name":"M-100B","count":"1653","first ":"1976-01-07","recent":"2008-09-11"},{"name":"Martlet 2","count":"148","first":"1963-04-01","recent":"1967-06-22"}, {"name":"Matador","count":"311","first":"1948-12-01","recen t":"1961-06-01"},{"name":"Meteor-1","count":"110","first":"19 63-01-01","recent":"1973-09-15"},{"name":"Minuteman 1B","count":"298","first":"1962-12-07","recent":"1993-06-15"}, {"name":"Minuteman

2","count":"190","first":"1964-09-24","recent":"2001-07-15"}, {"name":"Minuteman

':"311","first":"1968-08-16","recent":"2020-02-05"}, "name":"MMR-06","count":"778","first":"1969-10-13","recen ":"1992-04-10"},{"name":"Molniya 8K78M","count":"272","fi rst":"1965-10-04","recent":"2010-09-30"},{"name":"MR-1 2","count":"697","first":"1962-05-01","recent":"1997-02-05"}, ["name":"MT-135P","count":"819","first":"1969-01-05","rec ent":"2000-12-20"},{"name":"Nike","count":"240","first":"19 53-10-23","recent":"1977-11-19"},{"name":"Nike Ajax","count":"1089","first":"1952-02-25","recent":"1960-05-21"} "name":"Nike Apache","count":"875","first":"1958-0 3-18","recent":"1980-12-18"},{"name":"Nike Cajun","count":"716","first":"1956-07-06","recent":"1976-10-06"}, "name":"Nike Hercules","count":"209","first":"1957-03-13 ,"recent":"1971-01-01"},{"name":"Nike Hydac","cou nt":"102","first":"1966-09-20","recent":"1983-06-16"}, "name":"Nike Iroquois","count":"211","first":"19 64-11-27","recent":"1979-02-26"},{"name":"Nike Orion","count":"176","first":"1977-02-26","recent":"2002-05-16"}



The interface still sucks ListView

- Could improved the text by inserting \n appropriately
 - maybe get rid of a lot of the JSON formatting also
- But it is still just dead text.
- How to make it possible to sect rows and drill down

Android ListView / RecyclerView

List Views

- When things might take a while, you need to tell users that things are happening
- Simplest ListView are for arrays of Strings
- grey: add the list view to the interface
- red: set up the listview adapter
- green: click on row to drill down in database

mainLayout.removeAllViews(); for (JSONObject jo : aList) aString.add(jo.toString()); mainLayout.addView(theLv, new ViewGroup.LayoutParams.MATCH PARENT)); ArrayAdapter<String> adapter = new ArrayAdapter<String>(getContext(), android.R.id.text1, aString); theLv.setAdapter(adapter);

theLv.setOnItemClickListener(new AdapterView.OnItemClickListener() {

```
public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
           JSONObject o = aList.get(position);
           Toast.makeText(getContext(),
o.toString(),Toast.LENGTH_SHORT).show();
            FragmentManager fragmentManager =
((AppCompatActivity)getContext()).getSupportFragmentManager();
            FragmentTransaction transaction =
fragmentManager.beginTransaction();
            transaction.replace(MainActivity.MAIN ACTIVITY ID,
new FragmentLaunches(0), null);
            transaction.addToBackStack(null);
            transaction.commit();
    });
```

```
private void gotData0(final ArrayList<JSONObject> aList) {
   ArrayList<String> aString = new ArrayList<>();
   ListView theLv = new ListView(getContext());
ViewGroup.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
           android.R.layout.simple list item 1,
```

Rockets!!!

{"name":"Ariane 5G","count":"16","first":"1996-06-04 ","recent":"2003-09-27","family":"Ariane5"}

{"name":"Atlas D","count":"135","first":"1959-04-14", "recent":"1967-11-07","family":"Atlas"}

{"name":"Atlas E","count":"48","first":"1960-10-11","r ecent":"1995-03-24","family":"Atlas"}

{"name":"Atlas F","count":"69","first":"1961-08-09","r ecent":"1981-06-23","family":"Atlas"}

{"name":"Atlas IIA","count":"23","first":"1992-06-10"," recent":"2002-12-05","family":"Atlas"}

{"name":"Atlas IIAS","count":"30","first":"1993-12-16 ","recent":"2004-08-31","family":"Atlas"}

{"name":"Chang Zheng 2C","count":"57","first":"197 5-11-26","recent":"2020-06-10","family":"DF5"}

{"name":"Chang Zheng 2D","count":"46","first":"199 2-08-09","recent":"2020-05-31","family":"DF5"}

{"name":"Chang Zheng 2F","count":"12","first":"199 9-11-19","recent":"2016-10-16","family":"DF5"}

{"name":"Chang Zheng 4B","count":"35","first":"199 9-05-10","recent":"2019-12-20","family":"DF5"}

{"name":"Chang Zheng 4C","count":"27","first":"200



More on ListViews

- 2 parts: the view and the adapter
 - The view specifies where it is and what the container looks like
 - The adapter specifies the list items and how they are rendered
 - So the code is:
 - create the ListView
 - create the adapter
 - attach the adapter to the listview

ListView theLv = new ListView(getContext()); mainLayout.addView(theLv, new ViewGroup.LayoutParams(ViewGroup.LayoutParams.MATC H_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));

ArrayAdapter<String> adapter = new ArrayAdapter<String>(getContext(), android.R.layout.*simple_list_item_1*, android.R.id.text1, aString);

theLv.setAdapter(adapter);

This creates the most basic adapter which only works for array (or arrayList) of strings



Improving on the Basic ListView

- Most improvements come in the for of customizing the adapter
 - There are several adapter types; in a decade I have only used ArrayAdapter
- Here, rather than using the basic adapter, I created a custom one so I could use an array of JSONobject

```
ArrayAdapter<JSONObject> aa = new
```

```
@NonNull
    @SuppressLint("InflateParams")
    @Override
ViewGroup parent) {
        View view = convertView;
        if (view == null) {
        ll.removeAllViews();
        ll.setLayoutParams(new
ViewGroup.LayoutParams.WRAP CONTENT));
        tv.setText(msg.toString());
(3 * density), (int) (5 * density));
ViewGroup.LayoutParams.WRAP_CONTENT, 1));
        return view;
```



Comparing

Rockets!!!

{"name":"Ariane 5G","count":"16","first":"1996-06-04 ","recent":"2003-09-27","family":"Ariane5"}

{"name":"Atlas D","count":"135","first":"1959-04-14", "recent":"1967-11-07","family":"Atlas"}

{"name":"Atlas E","count":"48","first":"1960-10-11","r ecent":"1995-03-24","family":"Atlas"}

{"name":"Atlas F","count":"69","first":"1961-08-09","r ecent":"1981-06-23","family":"Atlas"}

{"name":"Atlas IIA","count":"23","first":"1992-06-10"," recent":"2002-12-05","family":"Atlas"}

{"name":"Atlas IIAS","count":"30","first":"1993-12-16 ","recent":"2004-08-31","family":"Atlas"}

{"name":"Chang Zheng 2C","count":"57","first":"197 5-11-26","recent":"2020-06-10","family":"DF5"}

{"name":"Chang Zheng 2D","count":"46","first":"199 2-08-09","recent":"2020-05-31","family":"DF5"}

{"name":"Chang Zheng 2F","count":"12","first":"199 9-11-19","recent":"2016-10-16","family":"DF5"}

{"name":"Chang Zheng 4B","count":"35","first":"199 9-05-10","recent":"2019-12-20","family":"DF5"}

{"name":"Chang Zheng 4C","count":"27","first":"200



Rockets!!!

{"name":"Ariane 5G","cou 2003-09-27","family":"Ari

{"name":"Atlas D","count" 967-11-07","family":"Atla

{"name":"Atlas E","count": 95-03-24","family":"Atlas'

{"name":"Atlas F","count": 81-06-23","family":"Atlas"

{"name":"Atlas IIA","count 002-12-05","family":"Atla

{"name":"Atlas IIAS","cou 2004-08-31","family":"Atl

{"name":"Chang Zheng 2 cent":"2020-06-10","fami

{"name":"Chang Zheng 2 cent":"2020-05-31","fami

{"name":"Chang Zheng 2 cent":"2016-10-16","fami

{"name":"Chang Zheng 4 cent":"2019-12-20","fami

{"name":"Chang Zheng 4 cent":"2019-11-27","fami

nt":"16","first":"1996-06-04","recent":" ane5"} :"135","first":"1959-04-14","recent":"1
nt":"16","first":"1996-06-04","recent":" ane5"} :"135","first":"1959-04-14","recent":"1
:"135","first":"1959-04-14","recent":"1
s"}
"48","first":"1960-10-11","recent":"19 '}
"69","first":"1961-08-09","recent":"19 '}
:":"23","first":"1992-06-10","recent":"2 s"}
nt":"30","first":"1993-12-16","recent":" as"}
C","count":"57","first":"1975-11-26","re ly":"DF5"}
D","count":"46","first":"1992-08-09","re ly":"DF5"}
F","count":"12","first":"1999-11-19","re ly":"DF5"}
B","count":"35","first":"1999-05-10","re ly":"DF5"}
C","count":"27","first":"2006-04-26","re ly":"DF5"}

- Ignoring the background color there is little difference.
 - So a bunch of work, not much noise

My custom



Going further with Custom

- Note that the textview is put in a linearlayout.
 - We can put ANYTHING into that.
 - Pictures? (If I had them)
 - buttons (or other active elements)
 - More, we can render the information we have in a more user friendly way
 - Put data in columns with column dividers

Atlas IIAS	30	1993-12-16	2004-08-31	<u>vehicle details</u>
Chang Zheng 2C	57	1975-11-26	2020-06-10	<u>vehicle details</u>
Chang Zheng 2D	46	1992-08-09	2020-05-31	<u>vehicle details</u>

```
ArrayAdapter<JSONObject> aa = new ArrayAdapter<JSONObject>(getContext(), 0, aList)
{
```

```
@NonNull
    @SuppressLint("InflateParams")
    @Override
    public View getView(int position, View convertView, @NonNull ViewGroup parent)
         // Unchanged code not shown
       { TextView tv = new TextView(getContext());
            tv.setPadding((int)(3*density),0,0,0);
            tv.setText(msg.optString("name", "name"));
            tv.setGravity(Gravity.START | Gravity.CENTER_VERTICAL);
            ll.addView(tv, new LinearLayout.LayoutParams(0,
ViewGroup.LayoutParams.MATCH_PARENT, 3)); }
         { View v = new View(getContext());
            v.setBackgroundColor(ContextCompat.getColor(getContext(),
R.color.purple 500));
            ll.addView(v, new LinearLayout.LayoutParams((int)(0.5*density),
ViewGroup.LayoutParams.MATCH PARENT, 0)); }
//etc
   TextView m = new TextView(getContext());
   m.setGravity(Gravity.CENTER);
   SpannableString mySpannableString = new SpannableString("vehicle details");
   mySpannableString.setSpan(new UnderlineSpan(), 0, mySpannableString.length(), 0);
   m.setText(mySpannableString);
   m.setTextSize(10);
   m.setTextColor(Color.rgb(0, 122, 255));
   ll.addView(m, new LinearLayout.LayoutParams(0, ViewGroup.LayoutParams.WRAP_CONTENT, 3));
   m.setOnClickListener(new View.OnClickListener()
        @Override
       public void onClick(View view)
           Log.i("THISS", "launcher datails");
           Toast.makeText(getContext(), "Launcher details", Toast.LENGTH_SHORT).show();
           FragmentManager fragmentManager =
((AppCompatActivity)getContext()).getSupportFragmentManager();
           FragmentTransaction transaction = fragmentManager.beginTransaction();
           transaction.replace(MainActivity.MAIN_ACTIVITY_ID, new FragmentLauncherDetails(msg),
null);
           transaction.addToBackStack(null);
           transaction.commit();
   });
```

More Customization

- Columns are cool, but they need column headers
- Column headers are cool, but they need sorting!

```
theLv.addHeaderView(headerRow(aList, aa));
```

```
private View headerRow(ArrayList<JSONObject> aList, ArrayAdapter<JSONObject> lv) {
    float density = getResources().getDisplayMetrics().density;
   LinearLayout ll = new LinearLayout(getContext());
   ll.setBackgroundColor(ContextCompat.getColor(getContext(), R.color.purple_200));
   ll.setOrientation(LinearLayout.HORIZONTAL);
   ll.setLayoutParams(new AbsListView.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, (int)(40*density)));
        TextView tv = new TextView(getContext());
       tv.setPadding((int)(3*density),0,0,0);
       tv.setText("name");
       tv.setGravity(Gravity.START | Gravity.CENTER_VERTICAL);
       ll.addView(tv, new LinearLayout.LayoutParams(0, ViewGroup.LayoutParams.MATCH PARENT, 3));
       tv.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                reSort(aList, "name");
                lv.notifyDataSetChanged();
       });
       View v = new View(getContext());
       v.setBackgroundColor(ContextCompat.getColor(getContext(), R.color.purple_500));
       ll.addView(v, new LinearLayout.LayoutParams((int)(0.5*density), ViewGroup.LayoutParams.MATCH PARENT, 0));
```

This goes into the method that creates the listview

Create a new view, actually a LinearLayout, that contains column headers. These need to be laid out exactly as in data.

click on a column header to sort the java using standard Java, in-place sorting (Not shown). Once the data is sorted, tell the listview that the data has changed





Before and After

Rockets!!!

{"name":"Ariane 5G","count":"16","first":"1996-06-04 ","recent":"2003-09-27","family":"Ariane5"}

{"name":"Atlas D","count":"135","first":"1959-04-14", "recent":"1967-11-07","family":"Atlas"}

{"name":"Atlas E","count":"48","first":"1960-10-11","r ecent":"1995-03-24","family":"Atlas"}

{"name":"Atlas F","count":"69","first":"1961-08-09","r ecent":"1981-06-23","family":"Atlas"}

{"name":"Atlas IIA","count":"23","first":"1992-06-10"," recent":"2002-12-05","family":"Atlas"}

{"name":"Atlas IIAS","count":"30","first":"1993-12-16 ","recent":"2004-08-31","family":"Atlas"}

{"name":"Chang Zheng 2C","count":"57","first":"197 5-11-26","recent":"2020-06-10","family":"DF5"}

{"name":"Chang Zheng 2D","count":"46","first":"199 2-08-09","recent":"2020-05-31","family":"DF5"}

{"name":"Chang Zheng 2F","count":"12","first":"199 9-11-19","recent":"2016-10-16","family":"DF5"}

{"name":"Chang Zheng 4B","count":"35","first":"199 9-05-10","recent":"2019-12-20","family":"DF5"}

{"name":"Chang Zheng 4C","count":"27","first":"200



Rockets!!!							
name	cou nt	first	recent				
Ariane 5G	16	1996-06-04	2003-09-27	<u>vehicle details</u>			
Atlas D	135	1959-04-14	1967-11-07	<u>vehicle details</u>			
Atlas E	48	1960-10-11	1995-03-24	<u>vehicle details</u>			
Atlas F	69	1961-08-09	1981-06-23	<u>vehicle details</u>			
Atlas IIA	23	1992-06-10	2002-12-05	<u>vehicle details</u>			
Atlas IIAS	30	1993-12-16	2004-08-31	<u>vehicle details</u>			
Chang Zheng 2C	57	1975-11-26	2020-06-10	<u>vehicle details</u>			
Chang Zheng 2D	46	1992-08-09	2020-05-31	vehicle details			
Chang Zheng 2F	12	1999-11-19	2016-10-16	vehicle details			
Chang Zheng 4B	35	1999-05-10	2019-12-20	vehicle details			
Chang Zheng	27	2006-04-26	2019-11-27	vehicle details			
	l		1				

ListView and RecyclerView

- Even after customization, a list view still looks like a list.
 - Sometimes this is exactly what you want
 - I would argue that this is correct for the first Rocket screen
- RecyclerView gives you a lot more
 - Horizontal scrolling of items
 - Layout items in grids
 - unequal sized items
 - ect.
- You can make a list view using a recycler, but it is harder.