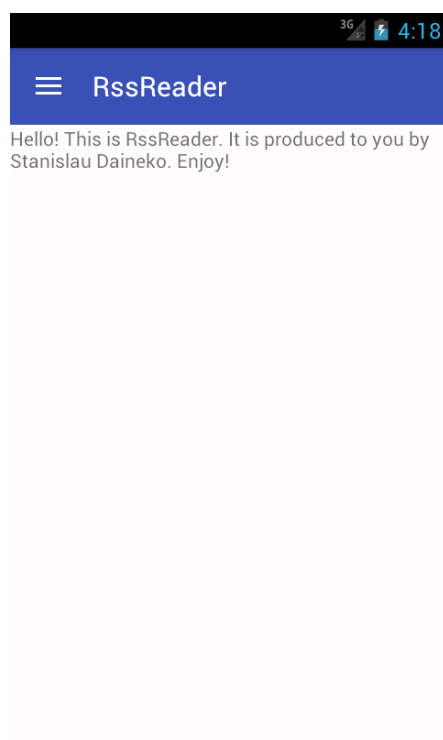


RssReader (Просмотрщик новостей)

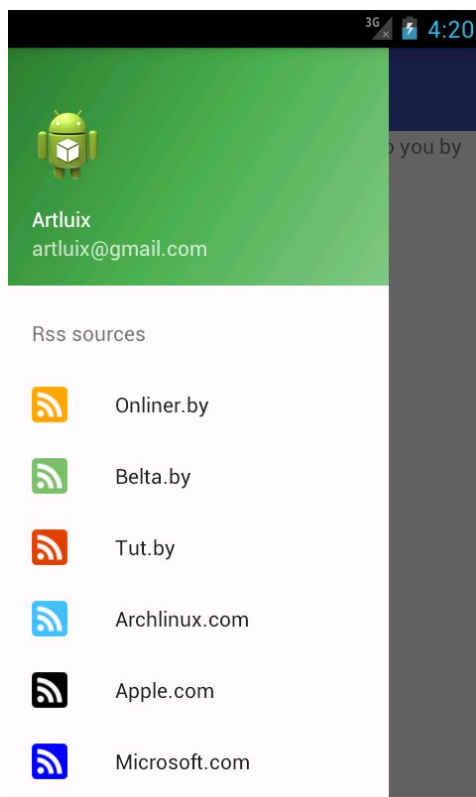
1. Описание приложения

Представляет собой обычный список новостей с некоторых источников. Каждый источник дает свой список новостей. В списке отображается название и дата. Новости сортируется по дате добавления. Пользователь может сохранять интересные ему новости в приложении, а также удалять новости по 1-й, или же, когда не в настроении, все сразу. И, разумеется, может просматривать новости в этом же приложении.

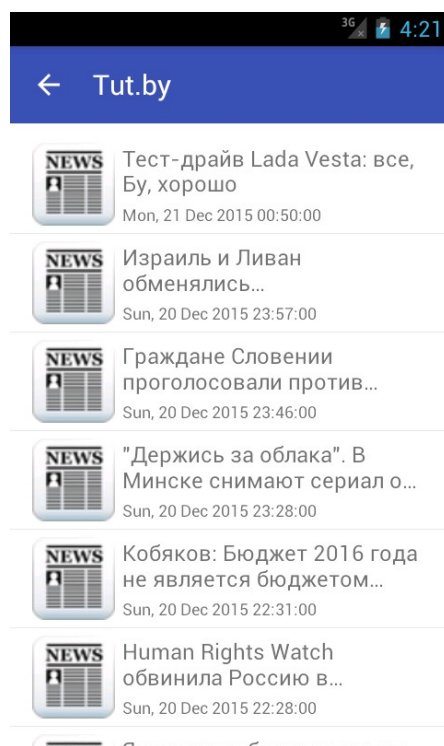
На экране приветствия имеется информация об авторе и его пожелания.



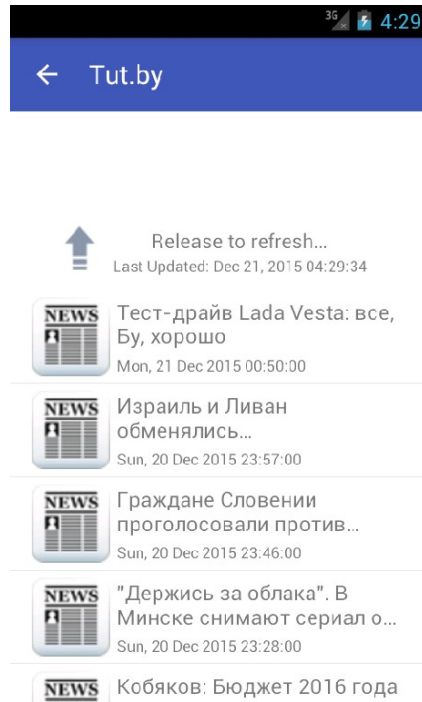
Для того, чтобы выбрать новостной портал, нужно открыть боковое меню. Там мы выбираем ресурс или сохраненные наши новости.



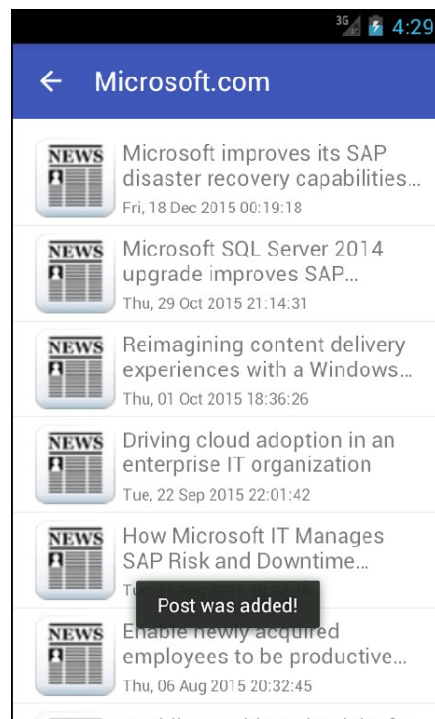
Далее автоматически новости подгружаются. Что выбрать желаемую новость из списка достаточно пролистать вниз список и сделать тап.



Чтобы обновить список нужно потянуть список вниз (drag-n-drop).



Чтобы сохранить новость нужно сделать длительное касание до появления уведомления в месте касания.

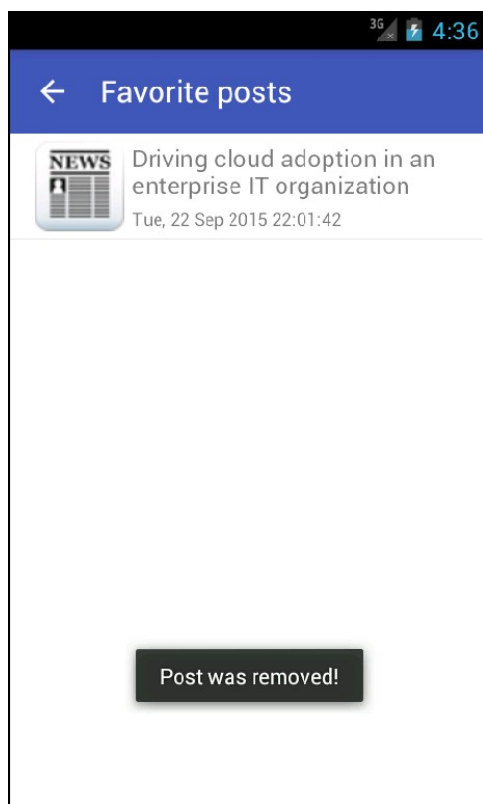


После нажатия на желаемую новость, будет выведено окно загрузки ее в окне. Далее вы можете спокойно просматривать новость.



Чтобы вернуться на предыдущее окно можно воспользоваться кнопкой back, а также кнопкой на toolbar'e.

В списке сохраненных новостей можно прочесть новость, удалить ее свайпом влево или вправо 1 пальцем. А также можно удалить все новости свайпом двумя пальцами в любую сторону.



2. Описание проекта

2.1. Модели

В качестве моделей здесь выступают новости (PostData).

```
public class PostData {
    private int id;
    private String link;
    private String title;
    private String date;
    public String getDate() {
        return date;
    }
    public void setDate(String date) {
        this.date = date;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getLink() {
        return link;
    }
    public void setLink(String link) {
        this.link = link;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
}
```

Все сохраненные новости хранятся в базе данных, взаимодействие с которой описано классом, наследником SQLiteOpenHelper

```
public class PostsDBHelper extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "postsReader.db";
    private String TABLE_NAME = "posts";
    private static final String COLUMN_NAME_ID = "id";
    private static final String COLUMN_NAME_TITLE = "title";
    private static final String COLUMN_NAME_LINK = "link";
    private static final String COLUMN_NAME_DATE = "date";
    public PostsDBHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_RSS_TABLE = "CREATE TABLE " + TABLE_NAME + "(" + COLUMN_NAME_ID
            + " INTEGER PRIMARY KEY," + COLUMN_NAME_TITLE + " TEXT,"
            + COLUMN_NAME_LINK + " TEXT," + COLUMN_NAME_DATE + " TEXT" + ")";
        db.execSQL(CREATE_RSS_TABLE);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
    public void addPost(PostData postData) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
    }
```

```

        values.put(COLUMN_NAME_TITLE, postData.getTitle());
        values.put(COLUMN_NAME_LINK, postData.getLink());
        values.put(COLUMN_NAME_DATE, postData.getDate());
        if (!isPostExists(db, postData.getLink())) {
            db.insert(TABLE_NAME, null, values);
            db.close();
        }
    }
    public ArrayList<PostData> getAllPosts() {
        ArrayList<PostData> listData = new ArrayList<PostData>();
        String selectQuery = "SELECT * FROM " + TABLE_NAME + " ORDER BY id DESC";
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery(selectQuery, null);
        if (cursor.moveToFirst()) {
            do {
                PostData postData = new PostData();
                postData.setId(cursor.getInt(0));
                postData.setTitle(cursor.getString(1));
                postData.setLink(cursor.getString(2));
                postData.setDate(cursor.getString(3));
                listData.add(postData);
            } while (cursor.moveToNext());
        }
        cursor.close();
        db.close();
        return listData;
    }
    public void deletePost(PostData postData) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_NAME, COLUMN_NAME_ID + " = " + String.valueOf(postData.getId()), null);
        db.close();
    }
    public void deleteAllPosts() {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_NAME, null, null);
        db.close();
    }
    public boolean isPostExists(SQLiteDatabase db, String link) {
        Cursor cursor = db.rawQuery("SELECT 1 FROM " + TABLE_NAME
            + " WHERE link = '" + link + "'", new String[] {});
        boolean exists = (cursor.getCount() > 0);
        return exists;
    }
}

```

2.2. Активити

В данном приложении всего 1 активити, в котором находится 3 фрагмента, причем все используются в полноэкранном режиме. Активити содержит боковое меню, которое является ключевым.

```

public class MainActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {
    private ActionBarDrawerToggle mDrawerToggle;
    private FragmentManager.OnBackStackChangedListener
        mOnBackStackChangedListener = new FragmentManager.OnBackStackChangedListener() {
        @Override
        public void onBackStackChanged() {
            syncActionBarArrowState();
        }
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
}

```

```

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
mDrawerToggle = new ActionBarDrawerToggle(
    this,
    drawer,
    R.string.navigation_drawer_open,
    R.string.navigation_drawer_close) {
    @Override
    public void onDrawerClosed(View drawerView) {
        syncActionBarArrowState();
    }
    @Override
    public void onDrawerOpened(View drawerView) {
        mDrawerToggle.setDrawerIndicatorEnabled(true);
    }
};
drawer.setDrawerListener(mDrawerToggle);
mDrawerToggle.syncState();
getFragmentManager().addOnBackStackChangedListener(mOnBackStackChangedListener);
NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);
navigationView.setItemIconTintList(null);
setTitle(getString(R.string.app_name));
}
@Override
protected void onDestroy() {
    getFragmentManager().removeOnBackStackChangedListener(mOnBackStackChangedListener);
    super.onDestroy();
}
private void syncActionBarArrowState() {
    int backStackEntryCount = getFragmentManager().getBackStackEntryCount();
    mDrawerToggle.setDrawerIndicatorEnabled(backStackEntryCount == 0);
}
}

```

Здесь мы реализуем синхронизацию возврата в предыдущую активити. А чуть выше реализовывали методы, для синхронизации кнопки тулбара с боковым меню.

```

@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    }
    else {
        int count = getFragmentManager().getBackStackEntryCount();
        if (count == 0) {
            super.onBackPressed();
        } else {
            getFragmentManager().popBackStack();
        }
    }
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (mDrawerToggle.isDrawerIndicatorEnabled() && mDrawerToggle.onOptionsItemSelected(item)) {
        return true;
    }
    else if (item.getItemId() == android.R.id.home &&
getFragmentManager().popBackStackImmediate()) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

Тут мы видим переход в активности, с сохранением возврата.

```
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    setTitle(item.getTitle());
    int itemId = item.getItemId();
    item.setChecked(true);
    switch (itemId) {
        case R.id.nav_onliner_by:
        case R.id.nav_belta_by:
        case R.id.nav_tut_by:
        case R.id.nav_archlinux_com:
        case R.id.nav_apple_com:
        case R.id.nav_microsoft_com:
            String url = getResources().getStringArray(R.array.rss_urls)[itemId -
R.id.nav_onliner_by];

            FetchPostsFragment fetchPostsFragment = new FetchPostsFragment();
            Bundle args = new Bundle();
            args.putString(FetchPostsFragment.ARG_POSTS_SRC, url);
            fetchPostsFragment.setArguments(args);
            FragmentManager fragmentManager = getFragmentManager();
            fragmentManager.beginTransaction().replace(R.id.content,
fetchPostsFragment).addToBackStack(null).commit();
            break;
        case R.id.nav_favorite_posts:
            FavoritePostsFragment favoritePostsFragment = new FavoritePostsFragment();
            fragmentManager = getFragmentManager();
            fragmentManager.beginTransaction().replace(R.id.content,
favoritePostsFragment).addToBackStack(null).commit();
            break;
        default:
            break;
    }
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
}
```

2.3. Асинхронные задачи

В данном приложении есть поддержка асинхронного выполнения задач, в том числе загрузка формата xml новостей с сайтов. Вот пример кода загрузки и парсинга XML

```
private class RssDataLoader extends AsyncTask<String, Integer, ArrayList<PostData>> {
    private RSSXMLTag currentTag;
    @Override
    protected ArrayList doInBackground(String... params) {
        String urlStr = params[0];
        InputStream is;
        ArrayList<PostData> resultDataList = new ArrayList<>();
        try {
            URL url = new URL(urlStr);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setReadTimeout(10000);
            connection.setConnectTimeout(10000);
            connection.setRequestMethod("GET");
            connection.setDoInput(true);
```



```

connection.connect();
is = connection.getInputStream();
XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
factory.setNamespaceAware(true);
XmlPullParser xpp = factory.newPullParser();
xpp.setInput(is, null);
int eventType = xpp.getEventType();
PostData newPostData = null;
SimpleDateFormat dateFormat = new SimpleDateFormat("EEE, dd MMM yyyy HH:mm:ss",
Locale.ENGLISH);

while (eventType != XmlPullParser.END_DOCUMENT) {
    if (eventType == XmlPullParser.START_TAG) {
        if (xpp.getName().equals("item")) {
            newPostData = new PostData();
            currentTag = RSSXMLTag.IGNORETAG;
        }
        else if (xpp.getName().equals("title")) {
            currentTag = RSSXMLTag.TITLE;
        }
        else if (xpp.getName().equals("link")) {
            currentTag = RSSXMLTag.LINK;
        }
        else if (xpp.getName().equals("pubDate")) {
            currentTag = RSSXMLTag.DATE;
        }
    }
    else if (eventType == XmlPullParser.END_TAG) {
        if (xpp.getName().equals("item")) {
            Date newPostDate = dateFormat.parse(newPostData.getDate());
            newPostData.setDate(dateFormat.format(newPostDate));
            resultDataList.add(newPostData);
        }
        else {
            currentTag = RSSXMLTag.IGNORETAG;
        }
    }
    else if (eventType == XmlPullParser.TEXT) {
        String content = xpp.getText();
        content = content.trim();
        if (newPostData != null) {
            switch (currentTag) {
                case TITLE:
                    if (content.length() != 0)
                        newPostData.setTitle(content);
                    break;
                case LINK:
                    if (content.length() != 0)
                        newPostData.setLink(content);
                    break;
                case DATE:
                    if (content.length() != 0)
                        newPostData.setDate(content);
                    break;
                default:
                    break;
            }
        }
    }
    eventType = xpp.next();
}

```

```

    }
}
catch (MalformedURLException e) {
    e.printStackTrace();
}
catch (IOException e) {
    e.printStackTrace();
}
catch (XmlPullParserException e) {
    e.printStackTrace();
}
catch (ParseException e) {
    e.printStackTrace();
}
return resultDataList;
}
@Override
protected void onPostExecute(ArrayList<PostData> resultList) {
    boolean b = false;
    int index = 0;
    for (int i = 0; i < resultList.size(); i++) {
        if (!guidList.contains(resultList.get(i).getLink())) {
            b = true;
            guidList.add(resultList.get(i).getLink());
            if (isRefreshLoading) {
                postList.add(index, resultList.get(i));
                ++index;
            }
            else {
                postList.add(resultList.get(i));
            }
        }
    }
    onDataLoadComplete(b);
    super.onPostExecute(resultList);
}
}

```

2.4. Анимация

При обновлении списка новостей используются 2 анимации с линейной интерполяцией

```

rotateAnimation = new RotateAnimation(0.0f, -180f, 1, 0.5f, 1, 0.5f);
rotateAnimation.setInterpolator(new LinearInterpolator());
rotateAnimation.setDuration(250);
rotateAnimation.setFillAfter(true);
reverseRotateAnimation = new RotateAnimation(-180.0f, 0.0f, 1, 0.5f, 1, 0.5f);
reverseRotateAnimation.setInterpolator(new LinearInterpolator());
reverseRotateAnimation.setDuration(1);
reverseRotateAnimation.setFillAfter(true);

```

2.5. Жесты

Как уже было сказано, есть жесты 1-пальцевые и 2-пальцевые. Интереснее всего будет рассмотреть реализацию удаления сохраненных новостей с помощью жестов.

```

private class PostItemClicksListener implements ListView.OnItemClickListener, ListView.OnTouchListener

```

```

private static final int MIN_DISTANCE_X = 100;
private static final int MAX_DISTANCE_Y = 40;
// for 1 finger
private float downX, downY, upX, upY;
// for 2 finger
private float downX_1, downY_1, upX_1, upY_1,
downX_2, downY_2, upX_2, upY_2;
// for 1 finger //gives as chance to separate click from touch
private boolean swipeDetected = false;
// for 2 fingers
private boolean swipeMode = false;
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    if (swipeDetected) {
        PostsDBHelper postsDBHelper = new PostsDBHelper(getActivity());
        postsDBHelper.deletePost(postList.get(position));
        Toast.makeText(getActivity(), "Post was removed!", Toast.LENGTH_SHORT).show();
        postList.remove(position);
        postItemAdapter.notifyDataSetChanged();
        swipeDetected = false;
    }
    else {
        Fragment webViewFragment = new PostWebFragment();
        Bundle args = new Bundle();
        args.putString(PostWebFragment.ARG_POST_URL, postList.get(position).getLink());
        webViewFragment.setArguments(args);
        FragmentManager fragmentManager = getFragmentManager();
        fragmentManager.beginTransaction().replace(R.id.content,
webViewFragment).addToBackStack(null).commit();
    }
}
@Override
public boolean onTouch(View v, MotionEvent event) {
    int pointerCount = event.getPointerCount();
    if (pointerCount == 1) {
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                downX = event.getX();
                downY = event.getY();
                swipeDetected = false;
                break;
            case MotionEvent.ACTION_MOVE:
                upX = event.getX();
                upY = event.getY();
                float deltaX = downX - upX;
                float deltaY = downY - upY;
                if ((Math.abs(deltaX) > MIN_DISTANCE_X) && (Math.abs(deltaY) < MAX_DISTANCE_Y)) {
                    swipeDetected = true;
                    return true;
                }
                break;
            default:
                return false;
        }
    }
    else if (pointerCount == 2) {
        switch (event.getAction() & MotionEvent.ACTION_MASK) {
            case MotionEvent.ACTION_POINTER_DOWN:
                downX_1 = event.getX(0);
                downY_1 = event.getY(0);
                downX_2 = event.getX(1);
                downY_2 = event.getY(1);
                swipeMode = true;
                break;
            case MotionEvent.ACTION_MOVE:
                if (swipeMode) {
                    upX_1 = event.getX(0);
                    upY_1 = event.getY(0);
                    upX_2 = event.getX(1);
                    upY_2 = event.getY(1);
                }
        }
    }
}

```

```

    }
    break;
case MotionEvent.ACTION_POINTER_UP:
    swipeMode = false;
    float deltaX_1 = downX_1 - upX_1;
    float deltaY_1 = downY_1 - upY_1;
    float deltaX_2 = downX_2 - upX_2;
    float deltaY_2 = downY_2 - upY_2;
    if ((Math.abs(deltaX_1) > MIN_DISTANCE_X) && (Math.abs(deltaY_1) < MAX_DISTANCE_Y)
        && (Math.abs(deltaX_2) > MIN_DISTANCE_X) && (Math.abs(deltaY_2) <
MAX_DISTANCE_Y)) {
        removeAllPosts();
        return true;
    }
}
return false;
}
}
}

```

Как видно, здесь мы переопределили методы касания и клика.

Данное приложение изначально задумывалось с музыкальным сервисом внутри, но была сделана ставка на целенаправленность.

3. Разметка

В конце хотелось бы привести код разметки модели

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="5dp"
    android:paddingBottom="5dp">
    <ImageView
        android:id="@+id/post_thumb"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:src="@drawable/ic_rss_news"
        android:scaleType="centerCrop"
        android:layout_marginRight="5dp"/>
    <TextView
        android:id="@+id/post_title"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignTop="@id/post_thumb"
        android:layout_toRightOf="@id/post_thumb"
        android:maxLines="2"
        android:text="This is a good Post, I hope it will have 2 lines. Please give me 2 lines"
        android:textIsSelectable="false"
        android:textSize="16sp"
        android:ellipsize="end"/>
    <TextView
        android:id="@+id/post_date"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignBottom="@id/post_thumb"
        android:layout_toRightOf="@id/post_thumb"
        android:maxLines="1"

```

```

        android:text="April 12, 2013"
        android:textSize="12sp" />
    </RelativeLayout>

```

и драг-н-дропа обновления списка новостей

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:paddingLeft="20.0dip" android:paddingBottom="10.0dip"
    android:layout_width="fill_parent" android:layout_height="wrap_content"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <RelativeLayout android:layout_width="fill_parent" android:layout_height="wrap_content">
        <LinearLayout android:gravity="center_horizontal" android:orientation="vertical"
            android:id="@id/linearLayout1"
            android:paddingLeft="10.0dip"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true">
            <TextView android:textSize="15.0sp"
                android:id="@id/head_tipsTextView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/pull_to_refresh" />
            <TextView android:textSize="12.0sp"
                android:id="@id/head_lastUpdatedDateTextView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/pull_to_refresh" />
        </LinearLayout>
        <FrameLayout android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:minWidth="30.0dip"
            android:minHeight="30.0dip"
            android:layout_toLeftOf="@id/linearLayout1"
            android:layout_alignParentTop="true">
            <ImageView android:layout_gravity="center"
                android:id="@id/head_arrowImageView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:src="@drawable/ic_arrow_refresh"
                android:contentDescription="@string/head_arrowImage" />
            <ProgressBar android:layout_gravity="center"
                android:id="@id/head_progressBar"
                android:visibility="gone"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                style="?android:attr/progressBarStyleSmall" />
        </FrameLayout>
    </RelativeLayout>
</LinearLayout>

```