Android Studio 2.2 に関するお知らせ

このたびは「はじめての Android プログラミング」をお買い上げいただきまして、ありがとうございます。

本書は Android Studio 2.0 を利用して動作検証を行ってありますが、2016 年 9 月 19 日に Android Studio 2.2 がリリースされました。このリリースは、レイアウトエディタが大幅に変更されているので、本書の記述とは以下のところが違っています。

P22 Live Template の設定変更

P28 作成する Activity の名前を決める画面。Backwards Compatibillity(AppCompat)というチェックボックスが増えています。これは後方互換に関する設定です。チェックをいれたままにしておいてください。

P49 レイアウトエディタの画面が新しくなり、Design ペインの左側にプレビュー表示、右側にブループリント（青い画面）が追加され、レイアウトが確認しやすくなっています。また「Component Tree」ペインのデフォルト位置が右上から左下に変更されています。

P51

「Pallette」ペインの「Widgets」にあったテキストビューのバリエーションがなくなり、「TextView」のみとなりました。本書で「Plain TextView」「Large Text」「Medium Text」「Small Text」と記載されている箇所は、「TextView」を配置し、画面右側の「Properties」ペインで TextView の「textAppearance」を次のように変更してください。

「Large Text」→textAppearance を AppCompat.Large に設定

「Medium Text」→textAppearance を AppCompat.Medium に設定

「Small Text」→textAppearance を AppCompat.Small に設定

また、「Palette」ペインでウィジットを選択して「Design」ペインに配置する時、黄色のラベルに「centerHorizontal,alignParentTop」といった配置位置の情報が表示されていましたが、表示がなくなっています。ブループリント（青い画面）に配置位置を示す線が表示されていますので、書籍のイメージを参考にしながら目的の位置に配置するようにしてください。

Android Studio 2.1 までは配置したビュー（TextView など）をダブルクリックすると、ダイアログが表示されていましたが、Android Studio 2.2 では右側の Properties に簡易表示として表示されます。以降、ラベルやテキスト、ボタンなどをダブルクリックしてダイアログを表示して操作する箇所は同様に読み替えてください。

従来ダイアログでの「id:」は、Properties の「ID」になります。同じく text:は「text」になります。スパナマークのついた「text」と間違えないようにしましょう。スパナマークの方には、プレビュー表示時にテスト表示したい文字列で、プログラム実行には反映されませ

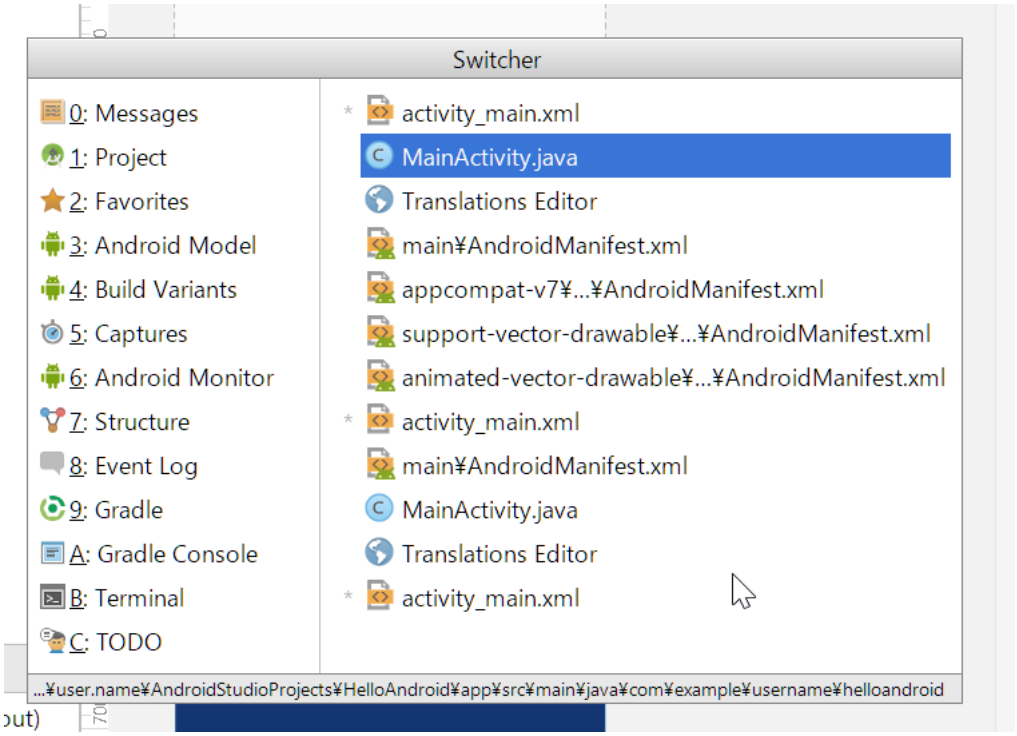
ん。
ただ、この 2.2 からの Properties の簡易表示はまだバグがあるようで、onClick プロパティを設定しても正しく動作しないことがあります。すぐに修正されると思いますが、しばらくは Properties の下部にある「View all properties」をクリックして全てのプロパティを表示し、全てのプロパティの方から onClick を指定してみてください。

P52
リソースを選択するための「Resource」ウィンドウのボタン位置が変更されています。
リソースを新規作成するためには画面右の「Add new resource」ボタンから「New string Value…」をクリックします。

P53
「Properties」ペインの表示が変更されました。プレビュー上でダブルクリックした時には、よく使うプロパティだけが表示されます。全てのプロパティを表示するには、下にある「View all properties」をクリックして表示を切り替えます。「View fewer properties」をクリックすると、よく使うプロパティ表示に切り替わります。「Properties」ペインの上部の左右の矢印「⇄」ボタンを押すと全プロパテイ表示と、よく使うプロパティ表示を切り替えることができます。
以降、ラベルやテキスト、ボタンなどを選択して Properties を表示して操作する箇所は同様に読み替えてください。

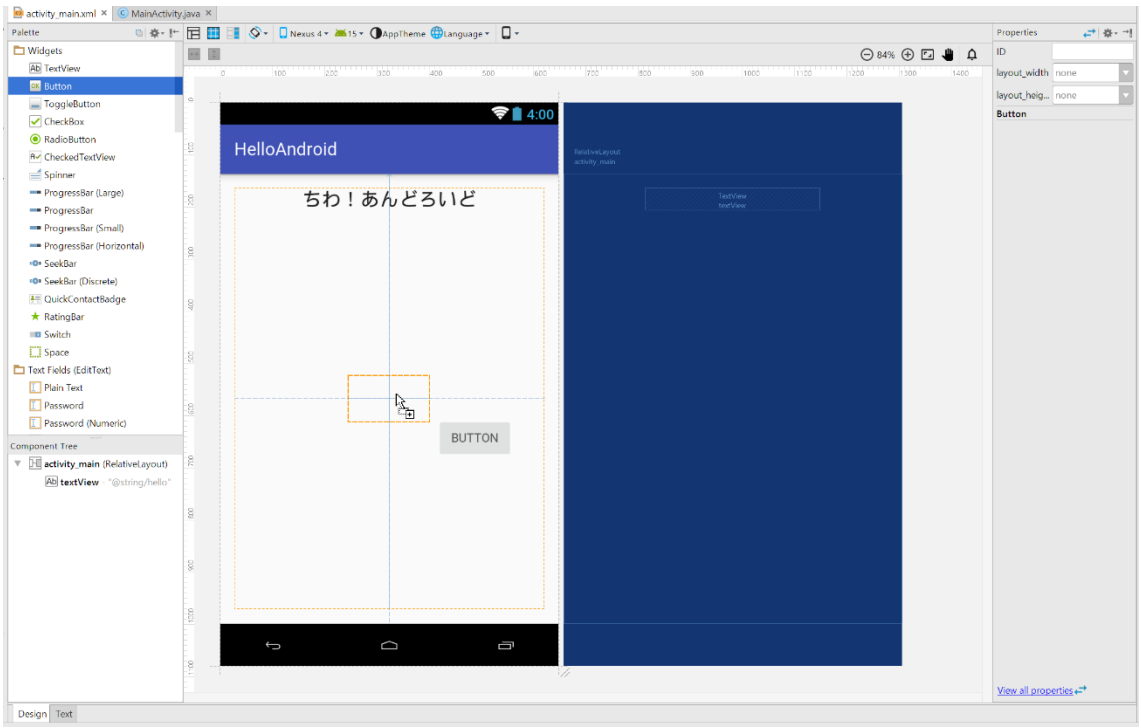
また、レイアウトに関するプロパティ表示が変更されました。レイアウトに関するプロパティは「layout_」から始まるプロパティのチェックを切り替えるようになります。ここでは「layout_alignParentTpo（親ビューの上部に合わせる）」と「layout_centerHorizontal（水平方向に中央）」にチェックが入っています。

P56
レイアウトエディタからアクティビティを開くボタンがありません。プロジェクトツールウィンドウから app→java→com.example.username.helloandroid→MainActivity をダブルクリックして MainActivity.java を開きます。または、Ctrl キー＋Tab キーを押して Switcher 画面を開き、MainActivity.java を選択して開くと素早く目的のアクティビティを開くことができます。



P57

ボタン配置時に「centerHorizontal」、「centerVertical」は表示されません。中央位置を示すガイドラインが表示されますのでこれを目印にボタンが中央になるように配置してください。「BUTTON」というラベルのボタンが配置されます。



P60

前述の通り、2.2 からの Properties の簡易表示はまだバグがあるようで、onClick プロパティを設定しても実行時に「ここをタップ！」ボタンを押すとエラーになることが有ります。これは、Properties の簡易表示画面で onClick プロパティを設定すると、不正な XML が生成されるためのようです。すぐにバグ修正されるとは思いますが、それまでは Properties の下部にある「View all properties」をクリックして全てのプロパティを表示してから、onClick プロパティを設定してください。

P61

2.1 先ほど追加した onClickButton メソッドを削除して、レイアウトエディタの「Properties」ペインで「onClick」を<unset>に戻しておいてください。

2.2 先ほど追加した onClickButton メソッドを削除して、レイアウトエディタの「Properties」ペインで「onClick」を none に戻しておいてください。

P65

画像・動画に関するビューのフォルダが「Widgets」から「Images & Media」に変更になっています。以降、ImageView、ImageButton を選択する場合、「Palette」ペインの「 Image & Media」から選択してください。

P66

ImageView を配置すると、自動で「Resources」ダイアログが開くようになりました。「lion」を選択して「OK」をクリックします。今まではここで設定した画像リソースは src プロパティに反映されていましたが、2.2 からはより汎用的になった srcCompat というプロパティに反映されます。ただし、ここではソースの基本形の説明のため AppCompatActivy を Activity に変更したので srcCompat プロパティは使えません。Properties を開き、srcCompat に設定されている値を削除し、改めて src プロパティの「…」ボタンをクリックして「Resources」ダイアログを開いて「lion」の画像リソースを指定し直します。

P73

「centerHorizontal」「alignParentTop,margin=100dp」の表示がありません。ガイドラインをみならが左右に中央になるようにしてください。上部とのマージンはおおよその位置へ配置した後、Properties の「Layout_Margin」の左の三角マークをクリックして展開し「layout_marginTop（上部マージン）」の値を 100dp に設定します。

P75

LinearLayout を画面中央に配置したら、「Properties」より「gravity」を開き、「center_vertical」にチェックを入れます。こうすることで、このレイアウト内に配置したビューを、「垂直方向で中央」に揃えることが出来ます。

P76、P77
2.2 より操作手順が若干変わります。次の手順で操作してください。
「Palette」ペインの「Images & Media」より「Image Button」を先ほど配置した LinearLayout にドラッグ&ドロップします。配置すると画像選択画面になるのでグーの画像を選択して「OK」をクリックします。「ImageButton」をダブルクリックして「Properties」で「ID」を「gu」に設定します。全プロパティ表示に切り替え、「layout_width」と「layout_height」の値を 100dp に設定、「scaleType」を「fitCenter」に設定、「layout_weight」に設定されている 1 を消しておきます。
チョキ、パーも同じように設定します。

P79
P80 の「テキストビューの上部にイメージビューを配置する」のイメージをみながら、イメージビューをテキストビューの上部に配置します。「Resources」ウィンドウが開きます。ここでは何か画像を選択しないとイメージビューを配置できません。とりあえず「com_pa」の画像を選択肢「OK」ボタンをクリックして閉じてください。
配置したイメージビューをダブルクリックし「Properties」より ID を「my_hand_image」に、「layout_width」「layout_height」を 150dp に設定します。「srcCompat」は値を消して空にしておきます。

P101
linerLayout2 の中にボタンを 1 つ配置すると、「layout_weight」の値に 1 が設定されるため、ボタンが幅一杯に表示されています。ボタンを 1 つ配置したら「layout_weight」の 1 の値を消してから、次のボタンを配置してください。

P105
コードの入力時、DialogFragment を入力すると、
「android.support.v4.app.DialogFragment?(multiple choice…)Alt+Enter」
と表示されます。これは、DialogFragment という名前のクラスが複数有りますよ、どれをインポートしますか？と訊いている訳です。Alt+Enter を押すと、候補が表示されますので、そのうち目的のクラスを選択すると、そのクラスをインポートしてくれます。
ここでは
android.support.v4.app.DialogFragment と android.app.DialogFragment が選択できま

す。android.support.～の方は、古いバージョンの Android で動作させるためのサポートライブラリになります。本書では、android.app.DialogFragment を選択してください。
AlertDialog も同様に、android.app.～の方を選択してください。

P108
コードの入力時 BigDecimal を入力すると
「java.math.BigDecimal?(multiple choice…)Alt+Enter」
と表示されます。Alt+Enter を押して、リストより BigDecimal(java.math)を選択してインポートしてください。
同じように Calender は java.util.Calendar を選択してインポートしてください。
android.icu.util.～は、Android Nougat Unicode と多言語化対応サポートの入ったクラス群です。本書では使用しません。

P110
実行してみよう
ここまできたら、再度アプリを実行してみましょう。
実行の前に、一旦

```
((MainActivity)getActivity())
        .setCalendar(year, monthOfYear, dayOfMonth);
```

をコメントアウトして実行しましょう。実行後は忘れずコメントアウトをはずしておいてください。

P115
RadioGroup を配置したら、Properties で「layout_height」を「wrap_content」に設定してください。
RadioButton を配置したら、Properties の「ID」を「man」に設定します。表示する文字列は「text」プロパティで設定します。

P118
TextView を配置し、ID を「height」に、「layout_width」を 100dp に設定した後、「layout_weight」の 1 を削除してください。

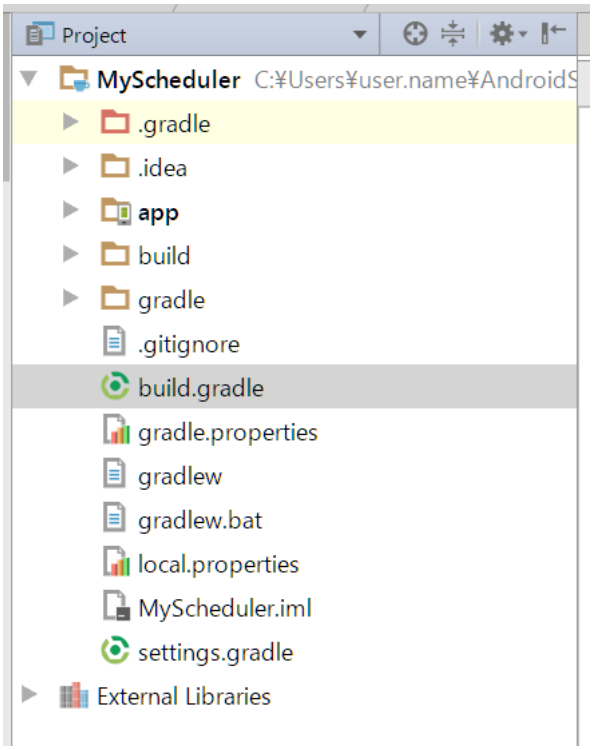
P132
サーフェスビューのフォルダが変更になっています。
「Palette」ペインの「Advanced」の「SurfaceView」を画面中央に配置してください。

P133
「padding」の値「left」「top」「right」「bottom」はそれぞれ「paddingLeft」「paddingTop」「paddingRight」「paddingBottom」と表記が変更されています。

P151
ImageSwitcher のフォルダが変更になっています。
「Palette」ペインの「Transitions」より「ImageSwitcher」を「スライドショー」ボタンの上に配置してください。
ImageSwitcher の上端を nextButton の下端に揃えるには Properties の「layout_below」に nextButton を設定します。

P175
フローティングアクションボタンのアイコンを変更する場合、「srcCompat」より変更してください。

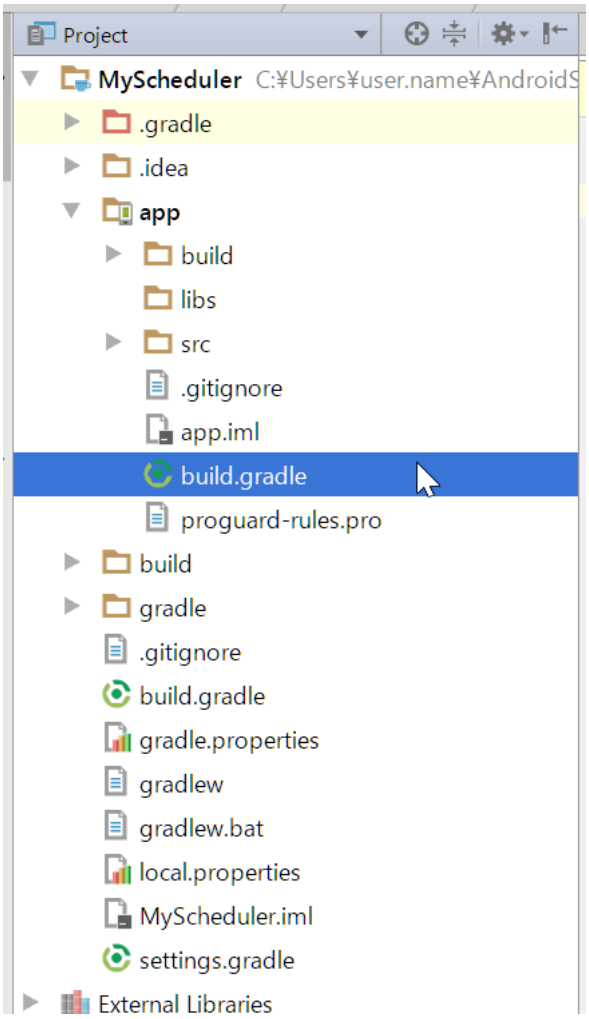
P176
Realm の本稿執筆時点の最新は 2.0.2 です。最新版の Realm を使用する場合、次の手順で導入してください。
プロジェクトツールウィンドウの表示を、「Project」ビューに変更してください。
MyScaduler 内の build.gradle を開きます。app フォルダ内にも同じ名前のファイルがあるので注意してください。



dependencies に「classpath "io.realm:realm-gradle-plugin:2.0.2"」を追記します。

```
buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.2.0'

        classpath "io.realm:realm-gradle-plugin:2.0.2"

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}
```

今度は、app フォルダ内の build.gradle を開きます。



上部に「apply plugin: 'realm-android'」を追記します。

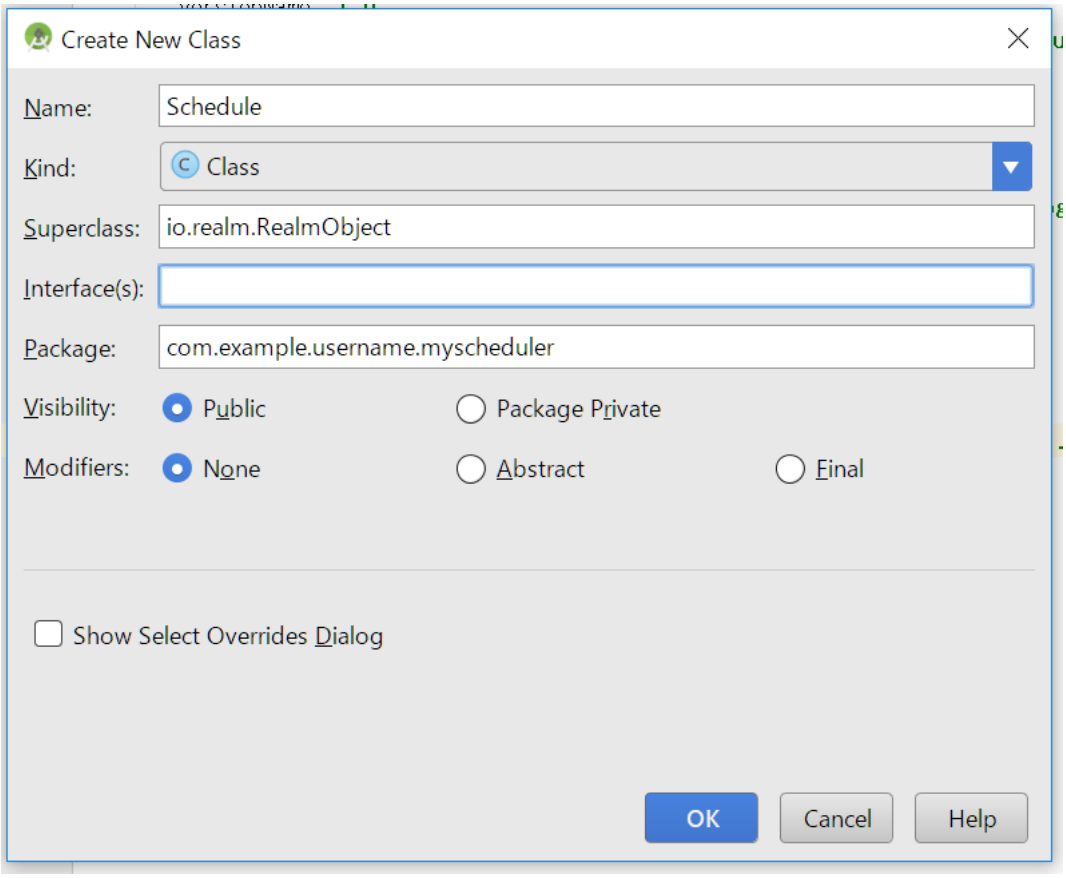
dependencies に「compile 'io.realm:android-adapters:1.3.0'」を追記します。

```
apply plugin: 'com.android.application'
apply plugin: 'realm-android'

android {
    compileSdkVersion 24
    buildToolsVersion "24.0.2"
    defaultConfig {
        applicationId "com.example.username.myscheduler"
        minSdkVersion 15
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
```

```
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:24.2.1'
    compile 'com.android.support:design:24.2.1'
    compile 'io.realm:android-adapters:1.3.0'
    testCompile 'junit:junit:4.12'
}
```

2 つの build.gradle を編集したら Tools→Android→Sync Project with Gradle Files を実行します。

「Create New Class」ダイアログが下記の画像のように変更されています。



Name：にクラス名「Schedule」を入力します。

Subclass：に「io.realm.RealmObject」と入力します。

OK をクリックして新しいクラスを作成します。

P182

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
```

```
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });
        Realm.init(this);
    }
```

Realm の初期化は init メソッドを使用します。

P183

「ScheduleAdapter」という名前のクラスを作成してください。

```
package com.example.username.myscheduler;

import android.content.Context;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import java.text.SimpleDateFormat;

import io.realm.OrderedRealmCollection;
import io.realm.RealmBaseAdapter;

import static android.R.attr.format;

public class ScheduleAdapter extends RealmBaseAdapter<Schedule> {

    public ScheduleAdapter(@NonNull Context context, @Nullable
OrderedRealmCollection<Schedule> realmResults) {
        super(context, realmResults);
    }

    private static class ViewHolder {
        TextView date;
        TextView title;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        ViewHolder viewHolder;

        if(convertView == null) {
            convertView = inflater.inflate(android.R.layout.simple_list_item_2,parent, false);
            viewHolder = new ViewHolder();
            viewHolder.date = (TextView)convertView.findViewById(android.R.id.text1);
            viewHolder.title = (TextView)convertView.findViewById(android.R.id.text2);
            convertView.setTag(viewHolder);
        } else {
            viewHolder = (ViewHolder)convertView.getTag();
        }
        Schedule schedule = adapterData.get(position);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd");
        String formatDate = sdf.format(schedule.getDate());
        viewHolder.date.setText(formatDate);
        viewHolder.title.setText(schedule.getTitle());
        return convertView;
    }
}
```

上記のように Realm 2.0.2 の RealmBaseAdapter では、対象データの取得に adapterData の get メソッドを使います（赤字部分）。またコンストラクタの引数が変更になっています。

P195

削除ボタンに赤色をつける方法です。「Resources」ダイアログを開き、画面左側で「Color」を選択した後「Add new resource」→「new color Value…」を選択します。
画面右側で好きな色を作成し、「OK」ボタンをクリックして色を指定します。

P196

Realm 2.0.2 では「Realm realm = Realm.getDefaultInstance();」として Realm インスタンスを取得します。

```
public void onSaveTapped(View view) {
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd");
    Date date = new Date();
    try {
        date = sdf.parse(mDateEdit.getText().toString());
    } catch (ParseException e) {
        e.printStackTrace();
    }

    Realm realm = Realm.getDefaultInstance();

    realm.beginTransaction();
    Number maxId = realm.where(Schedule.class).max("id");
    long nextId = 1;
    if (maxId != null) nextId = maxId.longValue() + 1;
    Schedule schedule = realm.createObject(Schedule.class);
    schedule.setId(nextId);
    schedule.setDate(date);
    schedule.setTitle(mTitleEdit.getText().toString());
    schedule.setDetail(mDetailEdit.getText().toString());
    realm.commitTransaction();

    Toast.makeText(this, "追加しました", Toast.LENGTH_SHORT).show();
    finish();

}
```

P205
Realm 2.0.2 では

```
results.remove(0);
```

は非推奨になっています。代わりに

```
results.deleteFromRealm(0);
```

を使用してください。ほかにも、
最初のオブジェクトを削除 deleteFirstFromRealm()
最後のオブジェクトを削除 deleteLastFromRealm()
すべてのオブジェクトを削除 deleteAllFromRealm()
といったメソッドが用意されています。

P208

Resources ウィンドウが開いたら、左側の「Color」タブを選択し、好きな色を選択するか、右上の「Add new resource」→「New color value…」を選択し、好きな色を作成して、「OK」ボタンをクリックします。

P222

2.2 からは、Properties のエキスパート向けプロパティ表示は廃止されています。

「Component Tree」ペインで「activity_main」を選択します。Properties がよく使うプロパティ表示になっていたら、「View all properties」をクリックして全プロパティ表示に変更します。「keepScreenOn」を探してチェックを入れます。

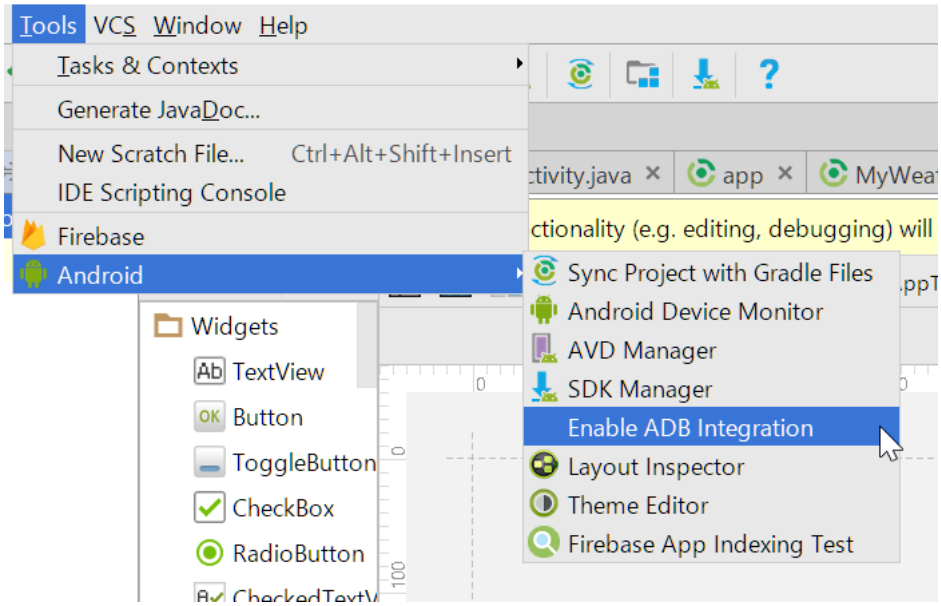
P228

Vollay を追加後、次のようなエラーが発生します。

Gradle 'MyWeather' project refresh failed

Error:Failed to delete original file 'C:¥Users¥user.name¥AppData¥Local¥Temp¥gradle_download2631086358563751438bin' after copy to 'C:¥Users¥user.name¥.gradle¥caches¥modules-2¥files-2.1¥com.android.tools.lint¥lint¥24.3.1¥57cb2a6361f32753bee3670f072f097840219e34¥lint-24.3.1.jar'

Instant Run を無効にします。メニューより Tools->Android->Enable ADB Integration を選択してチェックをはずします。



P230 の手順通り Vollay のビルドに必要なコンポーネントをインストールすると、先ほど

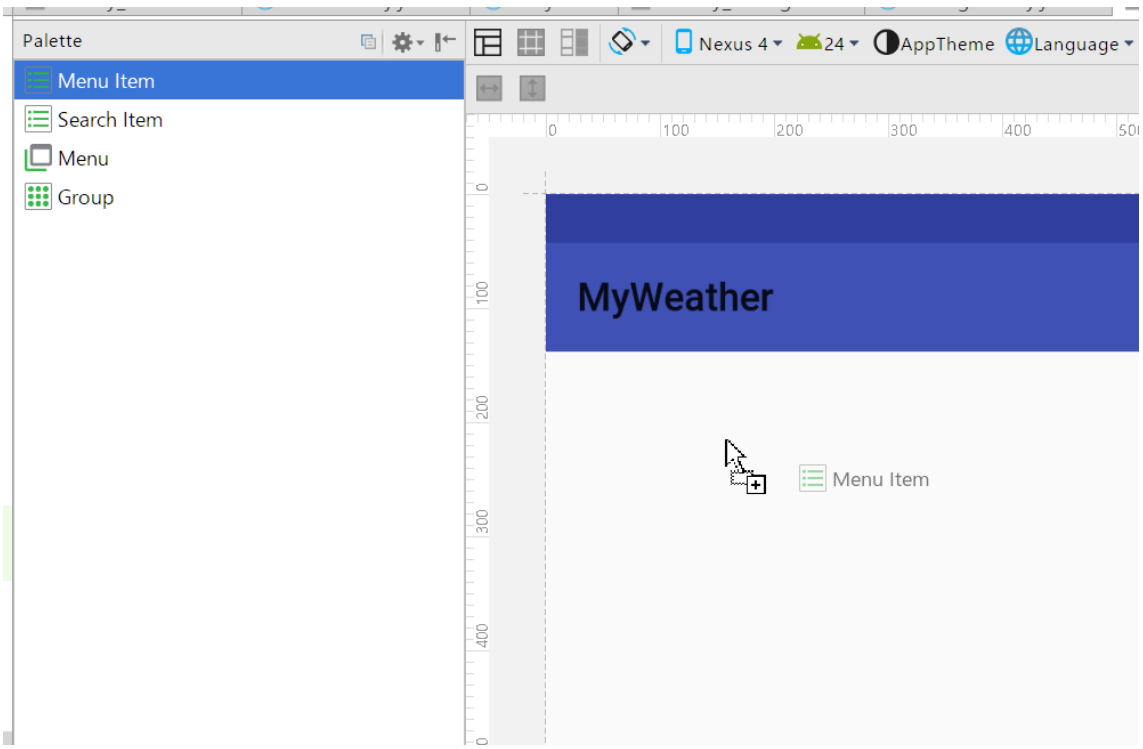
のエラーが再度表示されます。その場合は Tools->Android->Sync Project with Gradle Files を選択してビルドを行います。

ビルド時にエラーが発生しなければ OK です。Instant Run は再び有効にしてください。メニューより Tools->Android->Enable ADB Integration を選択してチェックを入れます。

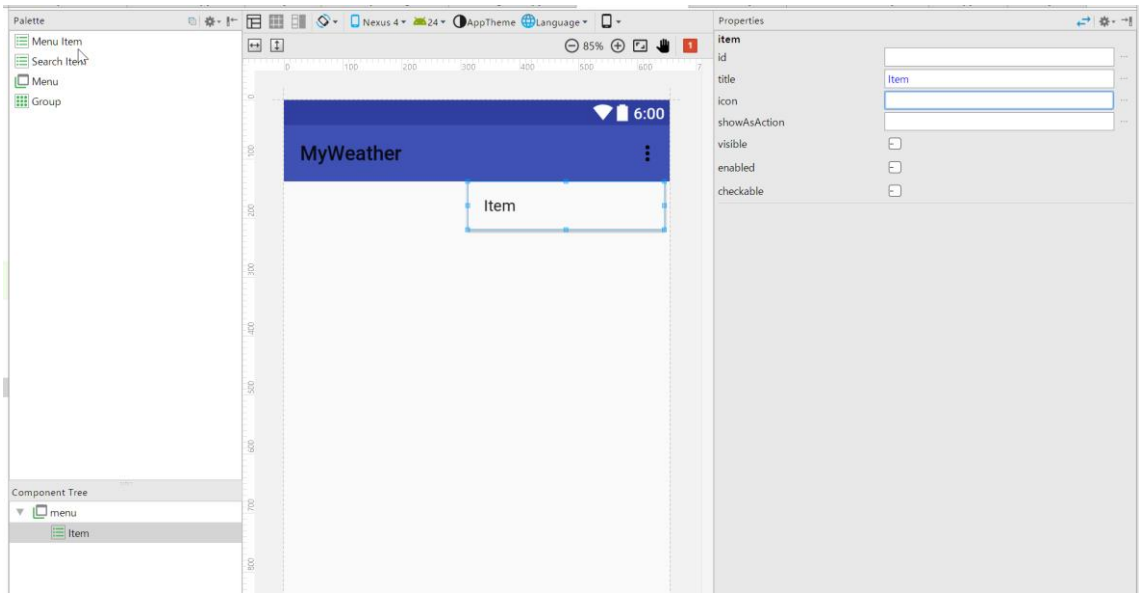
P250
ビュー選択ダイアログに NetworkImageView は表示されません。次ページ（P251）を参照して XML を手動で入力してください。

P261
build.gradle を修正した後、「Sync Now」をクリックするとエラーになります。その場合は Instant Run を無効にしてから「Sync Now」または「Sync Project with Gradle Files」を実行してください。プログラムを実行する時には Instant Run を再度有効にしてからプログラムを実行してください。

P274
グラフィカルなインターフェイスによるメニューの作成がサポートされています。Menu Item をドラッグ＆ドロップしてメニューを作成できます。
左側の Palette より「Menu Item」を中央のプレビュー上にドラッグ＆ドロップします。



アイテムが追加され、画面にプレビューが表示されます。

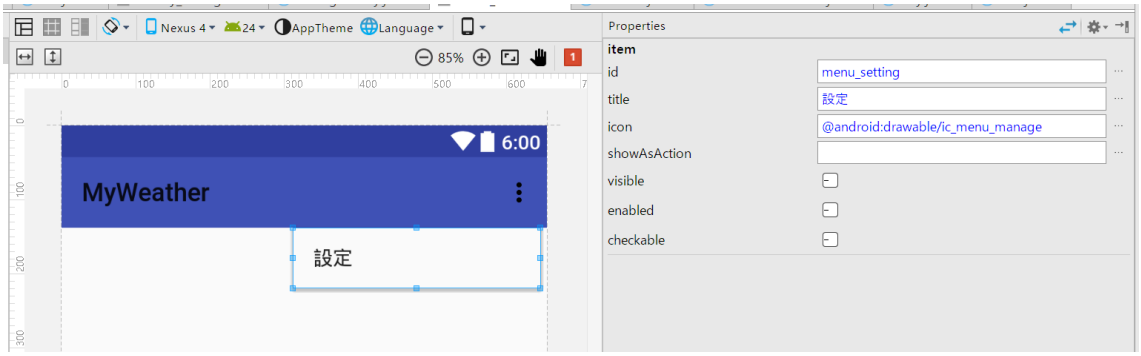


右側の Properties でドラッグした item のプロパティを設定します。

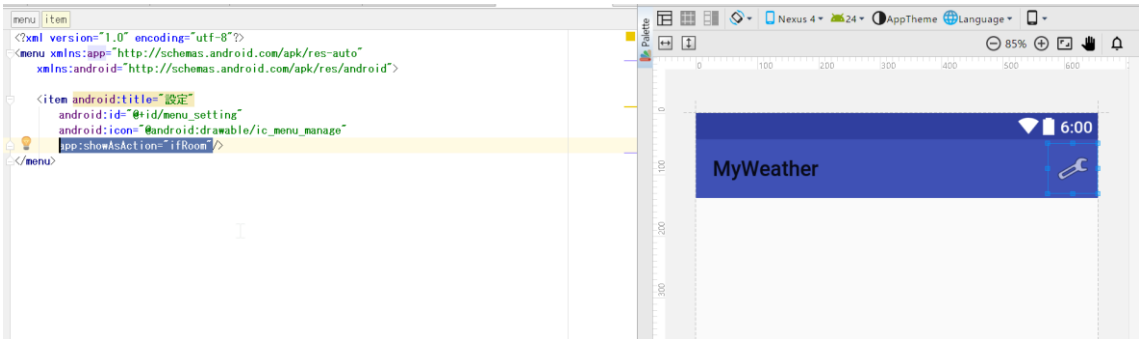
id：menu_setting

title：”設定”

ただし、showAsAction は何も設定せずにそのままにしておきます。



showAsAction だけ手動で設定してください。下にある Text タブをクリックして手入力します。

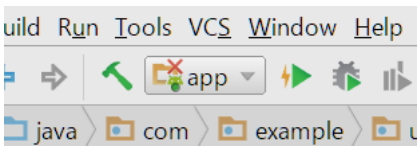


app:showAsAction=”ifRoom”
を追加してください。即座にプレビューに反映され、メニューがアイコンに変わります。

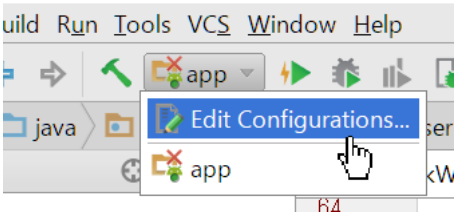
P305
左側の「Palette」の「Images&Media」から「ImageView」をドラッグ＆ドロップで配置します。配置すると Resources ダイアログが表示されます。任意の画像を選択して OK をクリックすると ImageView が配置されます。次の設定を行います。
ID：imageView
layout_width：80dp
layout_height：84dp
srcCompat：空にする
左側の「Palette」の「Widgets」で「TextView」をドラッグ＆ドロップで配置します。次の設定を行います。
ID：title
textAppearance：AppCompat.Medium
layout_alignTop：@+id/imageView
layout_toRightOf：@+id/imageView
layout_toEndOf：@+id/imageView

P306
左側の「Palette」の「Widgets」で「TextView」をドラッグ＆ドロップで配置します。次の設定を行います。
ID：telop
layout_centerVertical：チェックを入れる
textAppearance：AppCompat.Large
layout_toRightOf：@+id/imageView
layout_toEndOf：@+id/imageView

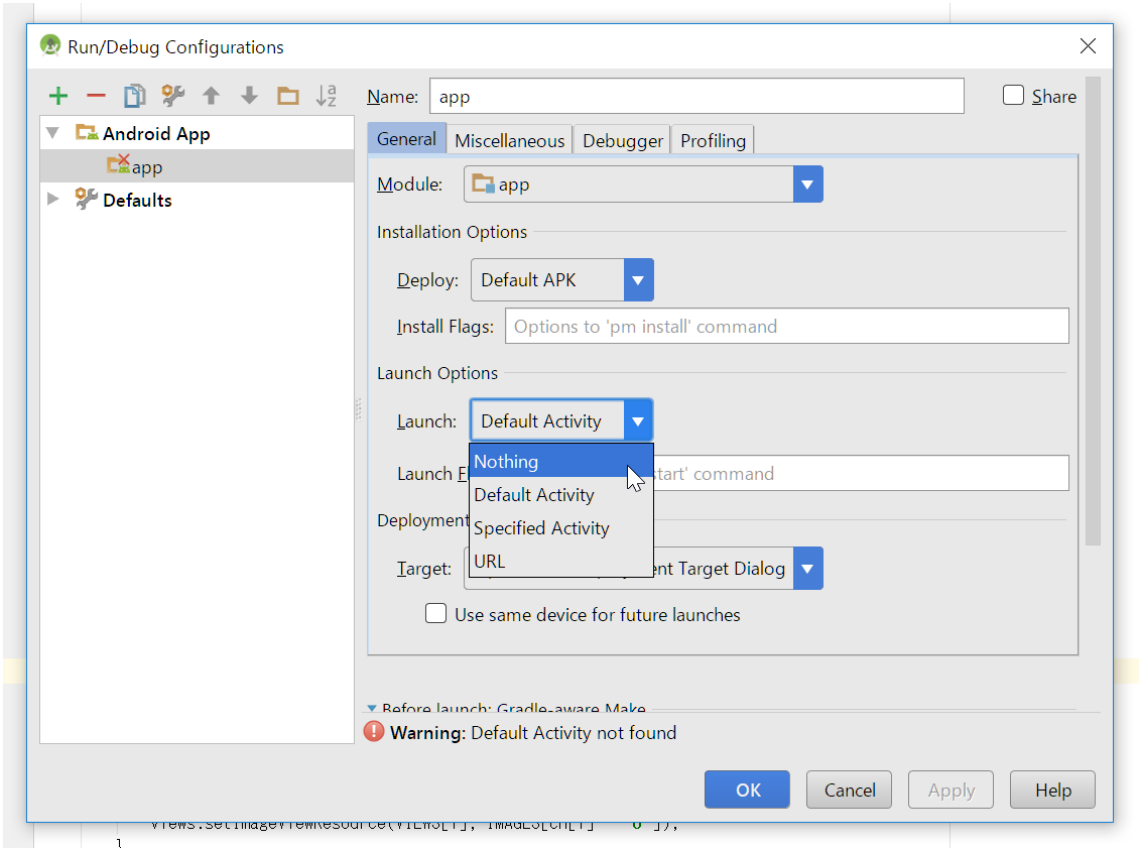
P324
プロジェクトにアクティビティが含まれないため、そのままでは実行することができません。



メニューバーの「app」に×が描かれ、エラーがあることが示されています。



「app」→「Edit Configurations…」をクリックします。



「Run/Debug Configurations」ダイアログが開きます。「Launch Options」の「Launch:」を「Nothing」に設定し、「OK」をクリックしてダイアログを閉じます。

エラーが消え、実行できるようになります。メニューバーの「app」からも×が消えています。