Ćwiczenia 24 – Android studio – AsyncTask, ExecutorService, DownloadManager

Na koniec zajęć prześlij pliki źródłowe (.xml, .java)+ obrazek do zasobu w teams.

1. Utwórz projekt o nazwie MyAsyncTask na podstawie Empty Activity, dobierz odpowiednie API.
2. Otwórz dokumentację:

   https://developer.android.com/reference/android/os/AsyncTask

   https://developer.android.com/reference/android/app/PendingIntent

   https://developer.android.com/training/notify-user/build-notification#progressbar

   https://developer.android.com/reference/java/util/concurrent/package-summary

   https://developer.android.com/reference/java/util/concurrent/Executor

3. activity_main.xml:

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:background="@drawable/ic_launcher_foreground" />

<ProgressBar
    android:id="@+id/progress_bar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:indeterminateTint="@color/purple_200">

</ProgressBar>
```

4. Zadeklaruj potrzebne stałe, np.: (adres wybierz dowolny)

```
imageSrc = "https://img.freepik.com/free-photo/painting-mountain-lake-with-mountain-background_188544-9126.jpg";
// imageSrc = "https://cdn.hejto.pl/uploads/posts/images/1200x900/f69f5ab680de25da695a0828ea674538.jpg";
 // imageSrc = getString(R.string.image_src);
```

5. Utwórz klasę MyAsyncTask

```
public class MyAsyncTask2 extends AsyncTask<String,String, Bitmap> {

    @Override
    protected void onPreExecute() {
        Log.v(AC.TAG, msg: "---------------------------> Start onPreExecute()");
        super.onPreExecute();
        dialog = new ProgressDialog( context: MainActivity.this);
        dialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        dialog.show();
    }
```

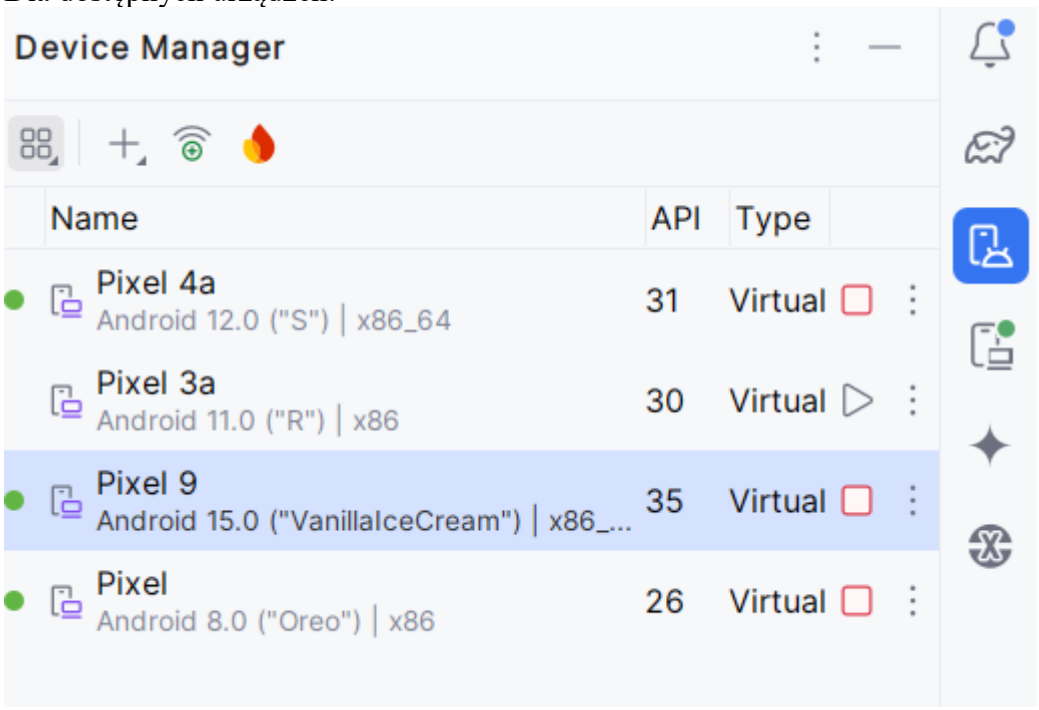
6. Dodaj metodę doInBackground:

```
@Override
protected Bitmap doInBackground(String... strings) {
    image = null;
    try{
        URL url = new URL(strings[0]);
        HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
        httpURLConnection.connect();
        image = BitmapFactory.decodeStream(httpURLConnection.getInputStream());
    }  catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return image;
}
```

7. Ustaw pobrane zdjęcie:

```
@Override
protected void onPostExecute(Bitmap bitmap) {
    super.onPostExecute(bitmap);
    dialog.cancel();
    imageView.setImageBitmap(bitmap);
}
```

8. Dla dostępnych urządzeń:



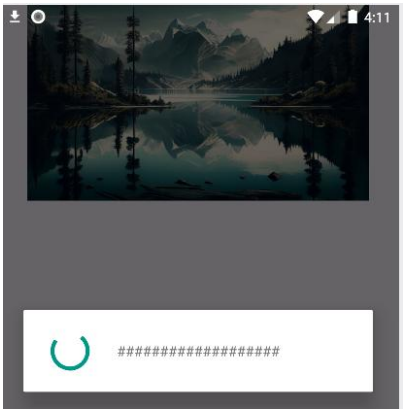
9. Uruchom Pixel i wywołaj instancję klasy:

```
if (Build.VERSION.SDK_INT <= Build.VERSION_CODES.O) {
    MyAsyncTask task = new MyAsyncTask( context: MainActivity.this, findViewById(R.id.image_view));
    task.execute(imageSrc);
    Log.v(TAG, msg: "<====================Q");
} else {
    Log.v(TAG, msg: "<==================== inne");
  //  executeMyService();
  //  executeMyService2();
  executeMyServiceWithDownloadManager();
}
```

10. Sprawdzenie logCata:

```
I  Kanał powiadomień został utworzony.
I  Tworzenie powiadomienia z paskiem postepu.
D  onPreExecute() called
V  <====================Q
D  doInBackground() called with: strings = [[Ljava.lang.String;@1a857a]
D  onPostExecute() called with: bitmap = [android.graphics.Bitmap@e3c39
```

11. Uzyskany efekt (tylko zdjęcie u dołu):

Widok z ProgressDialog: 

12. Klasa ExecutorService:

a) zrealizuj powyższe ćwiczenie z wykorzystaniem Klasy ExecutorService:

```
if (Build.VERSION.SDK_INT <= Build.VERSION_CODES.O) {
    MyAsyncTask task = new MyAsyncTask( context: MainActivity.this, findViewById(R.id.image_view));
    task.execute(imageSrc);
    Log.v(TAG, msg: "<====================Q");
} else {
    Log.v(TAG, msg: "<==================== inne");
    executeMyService();
  //  executeMyService2();
 // executeMyServiceWithDownloadManager();
}
```

```
}else{
    ExecutorService service = Executors.newSingleThreadExecutor();
        service.execute(new Runnable(){
            @Override
            public void run() {
                // onPreExecute
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        dialog = new ProgressDialog( context: MainActivity.this);
                        dialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
                        dialog.show();
                    }
                });
                // doInBackground
                try{
                    URL url = new URL(imagepath);
                    HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
                    httpURLConnection.connect();
                    image = BitmapFactory.decodeStream(httpURLConnection.getInputStream());
                }  catch (MalformedURLException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                // onPostExecute
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        dialog.cancel();
                        imageView.setImageBitmap(image);
                    }
                });
```

13. Sprawdź pobranie zdjęcia:

14. Zastąp przestarzały DialogProgress na rzecz ProgressBar: 

15. Utwórz klasę :

```
public class DownloadFromNetwork {

public static Bitmap downloadImage(String url){
    Bitmap bm = null;
    try{
        InputStream inputStream = new URL(url).openStream();
        bm = BitmapFactory.decodeStream(inputStream);
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return bm;
}
}
```

16. Sprawdź uprawnienia od wersji 33+ tiramisu


```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
    ActivityCompat.requestPermissions( activity: this, new String[]{
                    Manifest.permission.POST_NOTIFICATIONS},
            requestCode: 1);
}
```

17. Wywołaj

18. Sprawdź logcata:

```
I  Kanał powiadomień został utworzony.
I  Tworzenie powiadomienia z paskiem postepu.
V  <==================== inne
D  Rozpoczęto pobieranie pliku: downloadedImage_1738765515290.png
D  downloadImageWithDownloadManager() called with: url = [https://img.freepik.com/free-photo/painting-mountain-lake-with-mountain-background_188544-9126.jpg], con
I  Rozpoczęto pobieranie. ID: 46
I  Pobieranie zakończone! Plik: file:///storage/emulated/0/Android/data/gac.andrzej.cw24asynctaskexecutorservice/files/Download/downloadedImage_1738765515290.png
I  Wstawiam obrazek do imageView:
```

19. Metoda executeMyService2():

```
private void executeMyService2() {
    ExecutorService service2 = Executors.newSingleThreadExecutor();
    service2.execute(new Runnable() {
        @Override
        public void run() {
            // onPreExecute
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    progressBar.setVisibility(View.VISIBLE);
                }
            });
            // doInBackground
            image = DownloadFromNetwork.downloadImage(imagepath);
            // onPostExecute
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    progressBar.setVisibility(View.GONE);
                    if(image != null){
                        imageView.setImageBitmap(image);
                    }
                }
            });
        }
    });
}
```

20. Powiadomienie: dodaj powiadomienie z paskiem postępu, w stylu:


```
...
NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID);
builder.setContentTitle("Picture Download")
        .setContentText("Download in progress")
        .setSmallIcon(R.drawable.ic_notification)
        .setPriority(NotificationCompat.PRIORITY_LOW);

// Issue the initial notification with zero progress
int PROGRESS_MAX = 100;
int PROGRESS_CURRENT = 0;
builder.setProgress(PROGRESS_MAX, PROGRESS_CURRENT, false);
notificationManager.notify(notificationId, builder.build());

// Do the job here that tracks the progress.
// Usually, this should be in a
// worker thread
// To show progress, update PROGRESS_CURRENT and update the notification with:
// builder.setProgress(PROGRESS_MAX, PROGRESS_CURRENT, false);
// notificationManager.notify(notificationId, builder.build());

// When done, update the notification one more time to remove the progress bar
builder.setContentText("Download complete")
        .setProgress(0,0,false);
notificationManager.notify(notificationId, builder.build());
```

21. Fragment z realizacji:

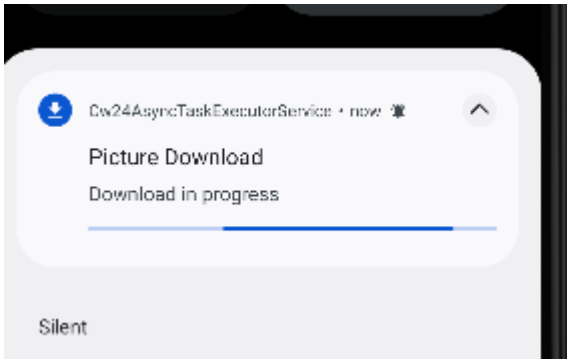
```
private void createNotificationChannel() {
    // Create the NotificationChannel, but only on API 26+ because
    // the NotificationChannel class is new and not in the support
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = getString(R.string.channel_name);
        String description = getString(R.string.channel_description
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
        NotificationChannel channel = new NotificationChannel(CHANN
        channel.setDescription(description);
        // Register the channel with the system; you can't change t
        // or other notification behaviors after this
        NotificationManager notificationManager = getSystemService(NotificationManager.class);
        notificationManager.createNotificationChannel(channel);
    }
}
```

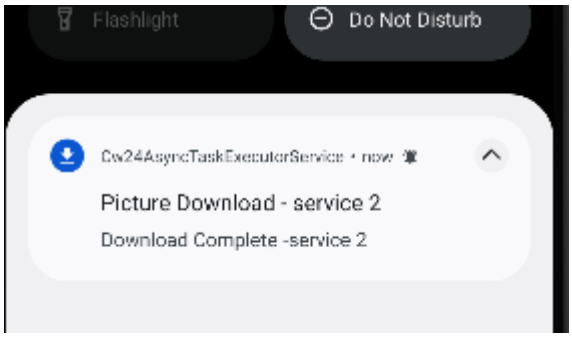
• Alerting the user by flashing t
Each of the notify methods take
parameter, which may be null.
id) if tag is unspecified. This pai
that pair should be unique withi
id) pair that is currently active a
For example, if you pass a new
replaced with the new one. This
or cancel(java.lang.String, int
Developer Guides
For a guide to creating notificati
See also:
• Notification ↗

22. Wywołanie na Pixel 4a:

```
if (Build.VERSION.SDK_INT <= Build.VERSION_CODES.O) {
    MyAsyncTask task = new MyAsyncTask( context: MainActivity.this, findViewById(R.id.image_view)
    task.execute(imageSrc);
    Log.v(TAG, msg: "<====================Q");
} else {
    Log.v(TAG, msg: "<==================== inne");
   // executeMyService();
    executeMyService2();
 // executeMyServiceWithDownloadManager();
}
```

23. Sprawdzenie:



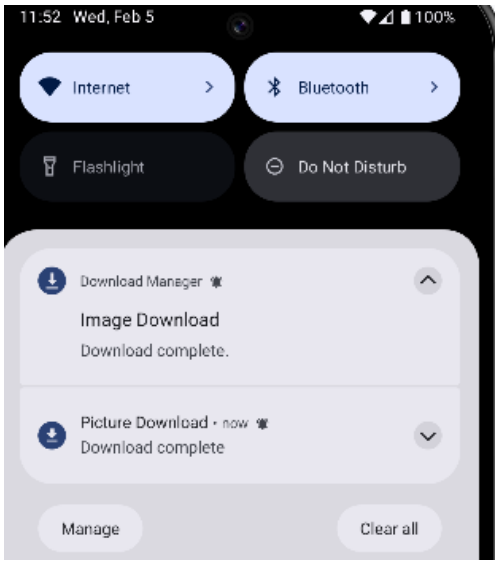


24. LogCat:

```
I  Kanał powiadomień został utworzony.
I  Tworzenie powiadomienia z paskiem postepu.
V  <==================== inne
I  set imageView OK - service 2
I  Download Complete - service 2
```

25. Użyj:
    https://developer.android.com/reference/android/app/DownloadManager

26. Efekt: (środkowe po kliknięciu w powiadomienie DownloadManagera)



27. KONIEC.