УДК 004.514, 004.424.7

І.В.Богач
Є.В.Сакун
Н.О.Іонова
А.С.Васюра

ДОСЛІДЖЕННЯ МЕТОДІВ ПРОГРАМНОГО ЗАВАНТАЖЕННЯ ЗОБРАЖЕНЬ НА ПЛАТФОРМІ ANDROID

Вінницький національний технічний університет

Анотація

В роботі досліджено методи програмного завантаження зображень на платформі Android, особливу увагу приділено методам завантаження зображень з ресурсів додатку, методам завантаження з віддалених серверів та роботі з бібліотекою Picasso. За результатами дослідження обрано оптимальний метод для вирішення поставленої задачі.

Ключові слова: Android, завантаження зображень, Picasso, Input Stream.

Abstract

This article explores how to programmatically upload images to the Android platform, paying particular attention to methods of downloading images from application resources, methods of downloading from remote servers, and working with the Picasso library. As a result of the research, the optimal method for solving the task was selected.

Keywords: Android, download photo, Picasso, Input Stream.

Вступ

З появою смартфонів сфера розробки програмного забезпечення суттєво розширилась. З'явилося багато різноманітних задач під нові операційні системи, такі як Android, iOS та інші. Актуальними знову стали задачі, які вже давно були вирішені під операційні ситеми Windows та Linux, але для нових операційних систем знову стали актуальними.

Дуже часто для додатків виникає необхідність завантаження зображень для інтерфейсу користувача таких як, наприклад, фон додатку, фон кнопок інтерфейсу, інтерфейс додаткових об'єктів і таке інше. Саме тому актуальним є дослідження існуючих методів завантаження зображень та вибір найбільш оптимального рішення в залежності від поставленої задачі з врахуванням часу завантаження та ресурсів, що є в наявності.

Android підтримує дві мови програмування: Java та Kotlin. Для проведення дослідження будемо використовувати мову програмування Java. В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Операційна система Android дає можливість підтримки зображень в форматі "*.png".

Для завантажень зображень в Android існує три основних метода:

- витягування даних з ресурсів;
- синхронне завантаження зображень (InputStream);
- асинхронне завантаження зображень (Picasso).

1 Завантаження зображень з ресурсів додатку

Даний метод завантаження зображень вимагає попереднє завантаження будь-яким зручним для користувача засобом файлу.

Android генерує ідентифікатори ресурсів для файлів зображень, розташованих в підкаталозі / res / drawable. Підтримуються файли PNG. Для кожного файлу зображення, який знаходиться в цьому каталозі, генерується унікальний ідентифікатор на основі імені файлу без розширення. Наприклад, якщо в нас є файл з ім'ям "cat.jpg", то для нього буде створено ідентифікатор ресурсу R.drawable.cat. Але потрібно додатково вручну стежити за унікальністю назв, інакше ми отримаємо помилку при компіляції додатку. Всі зображення потрібно зберігати тільки в одній основній папці (res / drawable), так як зображення з вкладених та інших папок не будуть зчитуватися.

Перевагою даного методу є те, що не потрібно завантажувати файли зі сторонніх ресурсів, дістати зображення можна за мінімальну кількість операцій процесора.

Недолік даного методу полягає в тому, що ресурси потрібно мати заздалегідь, та їх не можна гнучко замінювати. Для того, щоб використати інше зображення, яке не було задано до моменту компіляції продукту, потрібно додатково перезібрати apk файл, що вимагає додаткового часу.

2 Синхронне завантаження зображень з віддалених серверів

Даний метод є універсальним методом для завантажень зображень як із внутрішнього сховища, так і з інтернету, у випадку коли файл знаходиться у доступі для прямого завантаження.

Для початку скачування потрібно відкрити сам Input Stream за допомогою класу URL. Даному класу при створені об'єкта передається посилання на файл в сховищі, або на файл в інтернеті. В даному прикладі ми використовуємо editText з якого витягується посилання, яке вводить користувач.

URL url = new URL(editText.getText().toString());

Після цього використовуючи BitmapFactory.decodeStream(InputStream is) який «вміє» отримавши стрім перетворити його в Bitmap повернути його.

bitmap = BitmapFactory.decodeStream(url.openConnection().getInputStream());

Після даних дій bitmap можна використовувати для виведення на екран, відправки, обрізки, тощо.

Перевагою даного методу є те що фото зручно змінювати, замінивши файл на сервері.

Недоліками даного методу є: потреба в підключенні до інтернету, у разі завантажень з сервера; синхронність, та навантаження UI-потока, що робить не зручною у використанні програму, якщо завантажується велика кількість фото великого розміру.

3 Асинхронне завантаження зображень за допомогою бібліотеки Picasso

Даний метод використовує бібліотеку Picasso, що призначена для асинхронного завантаження зображень з мережі, ресурсів або файлової системи, їх кешування і відображення.

Приклад коду для завантаження зображення:

```
Picasso.with(context)
    .load(url)
    .placeholder(R.drawable.user_placeholder)
    .error(R.drawable.user_placeholder_error)
    .into(imageView);
```

Вказуються адреса картинки (url), заглушка (placeholder) та заглушка для помилки після трьох невдалих спроб завантаження (error); у методі into() вказуєте компонент ImageView, в який завантажується зображення.

При завантаженні картинка кешується і при повторному запиті на скачування, бібліотека може дістати картинку з кешу, а не завантажувати її з Інтернету, що суттєво прискорює роботу програми. Якщо кеш буде переповнений або видалений користувачем, тоді зображення знову скачується з мережі, що в більшості випадків є дуже зручним.

Якщо потрібно зберегти великі картинки в ресурсах або на зовнішньому накопичувачі, тоді краще використовувати окремий процес для завантаження.

Оскільки бібліотека вже налаштована на роботу в асинхронному режимі, тому ви можете використовувати її і в цих випадках.

Також дану бібліотеку зручно використовувати для завантаження з:

-ресурсів

```
Picasso.with(context).load(R.drawable.landing_screen).into(imageView1);
```

-внутрішнього сховища

```
Picasso.with(context).load(new File(...)).into(imageView2);
```

Метод бібліотеки fit () зменшує розмір картинки перед розміщенням в ImageView . Це корисно для економії ресурсів, якщо реальності потрібна маленька картинка, а не оригінал.

Перевагою даного методу є: асинхронність, що значно збільшує зручність додатку, його використання; гнучкість, де можливо налаштовувати як місце куди буде завантажуватись зображення, так і обробляти різні сценарії розвитку подій при завантаженні.

Недоліком даного методу є некерованість розміру закешованих даних, шо не завжди зручно для розробників.

Висновки

Було досліджено різноманітні методи завантаження зображень на платформі Android з використанням мови програмування Java.

В залежності від поставленої задачі з врахуванням часу, завантаження та ресурсів, що є в наявності, можна використовувати будь-який з описаних методів, що найбільше підходить під початкові умови, що є в наявності.

Але якщо є стабільний необмежений доступ до глобальної мережі Internet, найкращим варіантом для завантаження зображень є завантаження за допомогою бібліотеки Picasso. Саме такий метод дає найбільш гнучкі можливості для роботи з зображеннями, не навантажуючи UI-потік, а також дозволяє обробляти різні сценарії подій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Работа с изображениями. Ресурсы изображений. [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: https://metanit.com/java/android/2.9.php (дата звернення 26.12.2019) - Назва з екрана.
1. Як завантажити фотографії чи відео на пристрій.[Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: ttps://support.google.com/photos/answer/7652919?co=GENIE.Platform%3DAndroid&hl=uk (дата звернення 26.12.2019) - Назва з екрана.
2. ImageView. Выведение изображения [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: https://metanit.com/java/android/4.5.php (дата звернення 26.12.2019) - Назва з екрана.
3. Изучаем Android [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: http://developer.alexanderklimov.ru/android/ (дата звернення 26.12.2019) - Назва з екрана.
4. Picasso. A powerful image downloading and caching library for Android [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: https://square.github.io/picasso/ (дата звернення 26.12.2019) - Назва з екрана.
5. Picasso. Трасформация [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: http://developer.alexanderklimov.ru/android/library/picasso.php (дата звернення 26.12.2019) - Назва з екрана.
6. Работаем с графикой. Основы [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: http://developer.alexanderklimov.ru/android/simplepaint.php (дата звернення 26.12.2019) - Назва з екрана.
7. Папка res/values. Используем ресурсы приложения. [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: https://startandroid.ru/ru/uroki/vse-uroki-spiskom/18-urok-11-papka-resvalues-ispolzuem-resursy-prilozhenija.html (дата звернення 26.12.2019) - Назва з екрана.
8. Рисование. Bitmap. Чтение изображений большого размера [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: https://startandroid.ru/ru/uroki/vse-uroki-spiskom/372-urok-160-risovanie-bitmap-chtenie-izobrazhenij-bolshogo-razmera.html (дата звернення 26.12.2019) - Назва з екрана.
9. Рисование. Bitmap. Memory-кэш. Picasso [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: https://startandroid.ru/ru/uroki/vse-uroki-spiskom/376-urok-161-risovanie-bitmap-memory-kesh-picasso.html (дата звернення 26.12.2019) - Назва з екрана.
10. Предоставление ресурсов [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: https://developer.android.com/guide/topics/resources/providing-resources?hl=ru (дата звернення 26.12.2019) - Назва з екрана.

Богач Ілона Віталіївна, к.т.н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м.Вінниця, ilona.bogach@gmail.com.

Сакун Євгеній Вікторович, студент групи 1ІСТ-17б, кафедра автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м.Вінниця, fkca.1ict.sev@gmail.com.

Іонова Наталія Олександрівна, студентка групи 1ІСТ-17б, кафедра автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м.Вінниця, nataliionova@gmail.com.

Науковий керівник: Васюра Анатолій Степанович, професор кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м.Вінниця.

Bogach Ilona Vitaliivna, PhD, Associate Professor, Department of Automation and Intelligent Information Technologies, Vinnitsa National Technical University, Vinnytsia, ilona.bogach@gmail.com.

Sakun Evgeniy Viktorovych, student of group 1IST-17b, Department of Automation and Intelligent Information Technologies, Vinnitsa National Technical University, Vinnitsa, fkca.1ict.sev@gmail.com.

Ionova Natalia Alexandrovna, student of group 1IST-17b, department of automation and intellectual information technologies, Vinnitsa National Technical University, Vinnitsa, nataliionova@gmail.com.

Supervisor: Vasyura Anatoliy Stepanovich, Professor, Department of Automation and Intelligent Information Technology, Vinnitsa National Technical University, Vinnitsa