

Rapport TD3 ASL GIRALDO BOTERO Sebastian

Marshmallow 6.0 API level 23

1. Flickr L'idée est de vérifier régulièrement si des nouvelles données sont disponibles et de mettre à jour les données pour l'utilisateur.

Exercice 1 Regardez la documentation de l'API Flickr.

flickr Inscription Explorer Tirages Passez à la version Pro

Photos, personnes et

App Garden
Créer une application Documentation API Fils Qu'est-ce que App Garden ?

Les développeurs externes peuvent utiliser l'API Flickr à des fins non commerciales. Toute utilisation commerciale doit faire l'objet d'une autorisation préalable.

Consultez les rubriques suivantes :

- Guide du développeur
- Présentation
- Contage
- Authentification de l'utilisateur

- Dates
- Tags
- URL
- Avatars

- Conditions d'utilisation de l'API Flickr

- Clés API
- Liste de diffusion des développeurs

Méthodes API

activity

- flickr.activity.setUserComments
- flickr.activity.setUserPhotos

auth

- flickr.auth.checkToken
- flickr.auth.getFrob
- flickr.auth.getFullToken
- flickr.auth.getToken

auth.oauth

- flickr.auth.oauth.checkToken
- flickr.auth.oauth.getAccessToken

blogs

- flickr.blogs.getList
- flickr.blogs.getServices
- flickr.blogs.postPhoto

cameras

- flickr.cameras.getBrandModels
- flickr.cameras.getBrands

collections

- flickr.collections.getInfo
- flickr.collections.getTree

commons

- flickr.commons.getInstitutions

contacts

- flickr.contacts.getList
- flickr.contacts.getListRecentlyUploaded
- flickr.contacts.getPublicList
- flickr.contacts.getTaggingSuggestions

favorites

- flickr.favorites.add
- flickr.favorites.getContext
- flickr.favorites.getList
- flickr.favorites.setPublicList

API de chargement de photos

- Chargement de photos
- Remplacement de photos
- Exemple de requête
- Chargement asynchrone

Formats de requête

- REST
- XML-RPC
- SOAP

Formats de réponse

- REST
- XML-RPC
- SOAP
- JSON
- PHP

Kits API

Remarque : Les kits API ne sont pas entretenus par Flickr et Flickr n'endosse aucune responsabilité quant à leur utilisation. Pour obtenir de l'aide sur le kit API, rejoignez la liste de diffusion des développeurs ou contactez directement les personnes chargées de la maintenance.

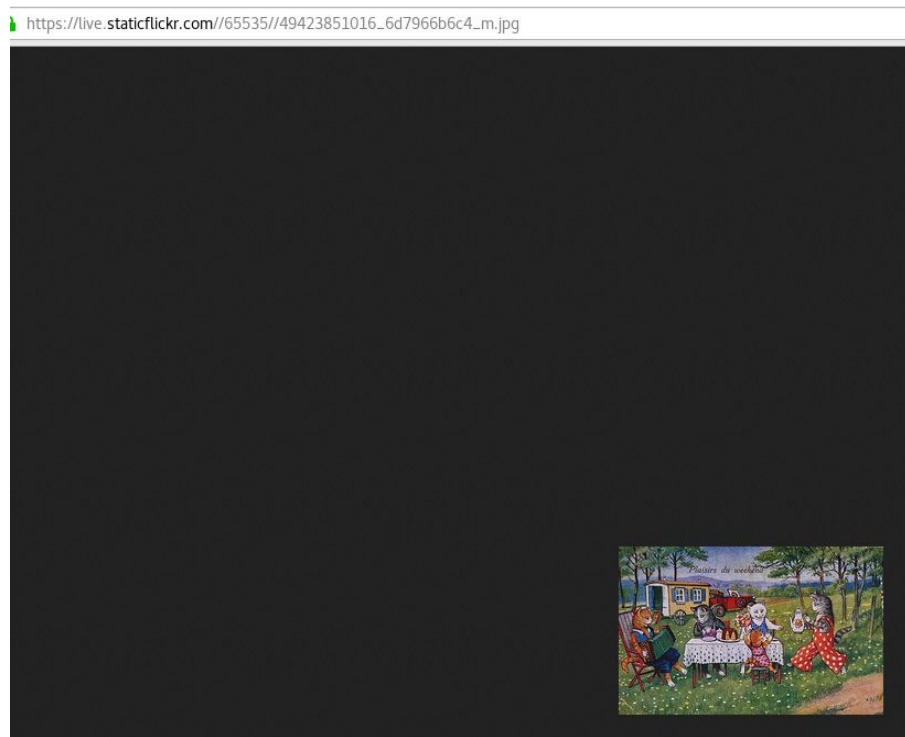
ActionScript

- api flickr (docs)
- Flashr

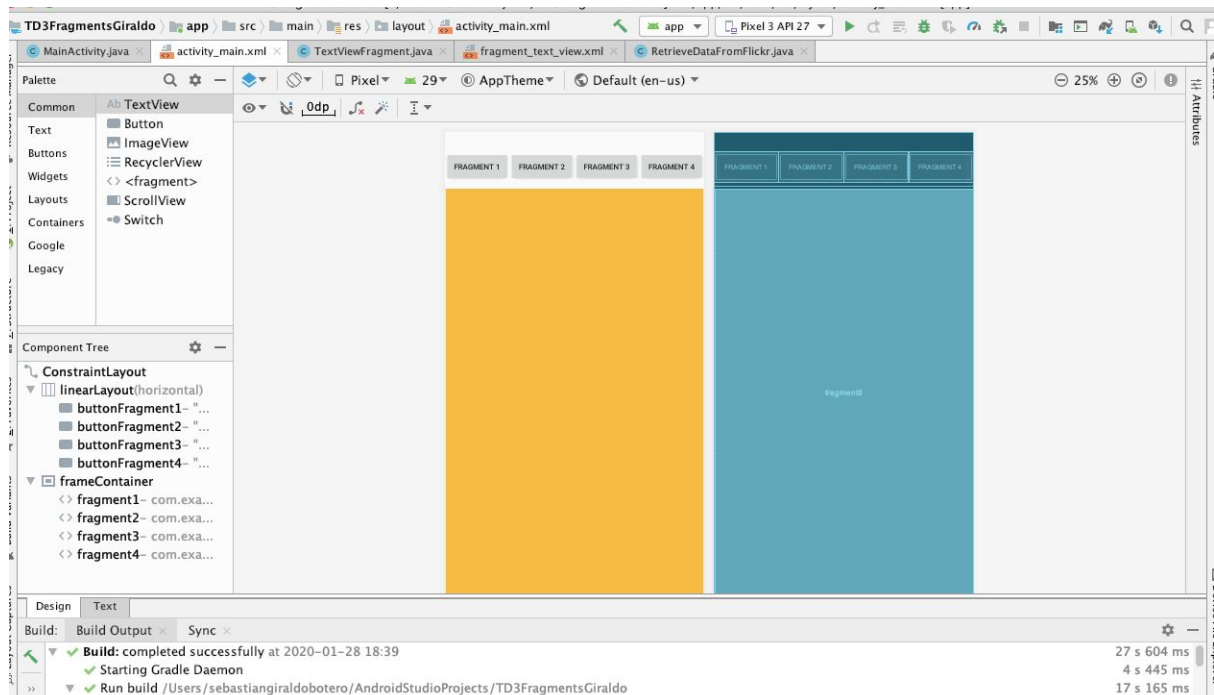
Exercice 2 Regardez le résultat de la requête ci-dessous. L'idée est donc de mettre à jour vos données locales régulièrement pour pouvoir présenter ces données à l'utilisateur.

```
jsonFlickrFeed({
  "title": "Recent Uploads tagged cats",
  "link": "https://www.flickr.com/photos/tags/cats/",
  "description": "",
  "modified": "2020-01-22T13:40:40Z",
  "generator": "https://www.flickr.com",
  "items": [
    {
      "title": "Diego et Charlotte ! ->",
      "link": "https://www.flickr.com/photos/joellisahttp://www.flickr.com/photos/49424539697/",
      "media": {
        "m": "https://live.staticflickr.com/65535/4942453969_374b1895bf_m.jpg",
        "date_taken": "2020-01-21T16:25:36-08:00"
      },
      "description": " ",
      "published": "2020-01-22T13:40:40Z",
      "author": "nobody@flickr.com (Joëlle Lisa)",
      "author_id": "248331370888",
      "tags": "janvier2020 diego charlotte deux two chats cats rb bw piconkey"
    },
    {
      "title": "Sleeping cats",
      "link": "https://www.flickr.com/photos/885261970800/49424138452/",
      "media": {
        "m": "https://live.staticflickr.com/65535/49424138452_7e3d46553d_m.jpg",
        "date_taken": "2020-01-09T12:35:57-08:00"
      },
      "description": " ",
      "published": "2020-01-22T13:40:40Z",
      "author": "nobody@flickr.com (Joëlle Lisa)",
      "author_id": "885261970800",
      "tags": "postcards received bargo cats"
    },
    {
      "title": "Plaisirs du weekend",
      "link": "https://www.flickr.com/photos/885261970800/49423833841/",
      "media": {
        "m": "https://live.staticflickr.com/65535/49423833841_90e3f0e654_m.jpg",
        "date_taken": "2020-01-09T12:35:57-08:00"
      },
      "description": " ",
      "published": "2020-01-22T13:40:40Z",
      "author": "nobody@flickr.com (Joëlle Lisa)",
      "author_id": "885261970800",
      "tags": "received postcards vintage cats vehicles"
    },
    {
      "title": "2 cats under the stars",
      "link": "https://www.flickr.com/photos/885261970800/49423833841/",
      "media": {
        "m": "https://live.staticflickr.com/65535/49423833841_90e3f0e654_m.jpg",
        "date_taken": "2020-01-09T12:35:57-08:00"
      },
      "description": " ",
      "published": "2020-01-22T13:40:40Z",
      "author": "nobody@flickr.com (Joëlle Lisa)",
      "author_id": "885261970800",
      "tags": "received postcards vintage cats vehicles"
    }
  ]
})
```

Exercice 3 Analysez ce que contient la réponse JSON. Vous devriez retrouver les urls des images



Exercice 4 Créez une activité avec plusieurs boutons permettant de changer le fragment situé en bas de l'activité.



Class **RetrieveDataFromFlickr.java** fait avec la librairie Volley

- ```

66 RequestQueue queue = Volley.newRequestQueue(getActivity());
67 StringRequest stringRequest = new StringRequest(Request.Method.GET, flickr_reponse,
68 (response) → {
69 responseJSON.setText(response);
70 Log.i(tag: "SGB", response);
71 }, (error) → { Log.i(tag: "SGB", msg: "Error return request"); });
72 queue.add(stringRequest);
73
74
75
76
77
78
79
80
81

```

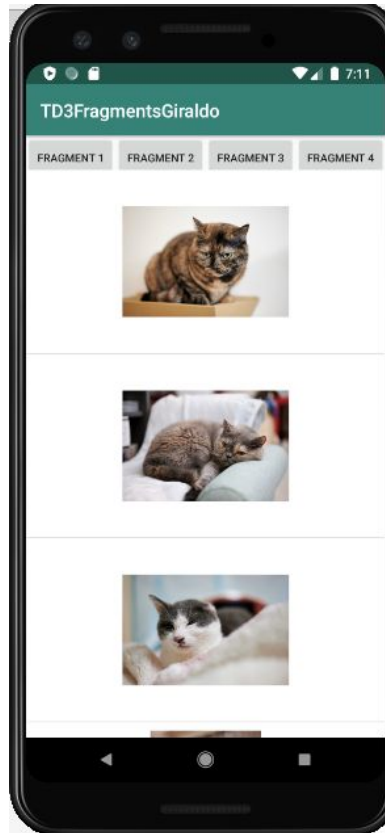
TD3FragmentsGiraldo

| FRAGMENT 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | FRAGMENT 2 | FRAGMENT 3 | FRAGMENT 4 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------|------------|
| [{"photos":1,"page":1,"pages":238556,"perpage":10,"total":238552,"photo":{"id":"4944313296b","owner":"158958545@N04","secret":"c6049a9e542","server":"65535","farm":66,"title":"Katzte mit Beute","ispublic":1,"isfriend":0,"id":"4944303598b","owner":"134266660@N04","secret":"b0c8ebc620","server":"65535","farm":66,"title":"MG_3703","ispublic":1,"isfriend":0,"isfamily":0,"id":"49443035348","owner":"134426660@N04","secret":"7f3b7d3e3e","server":"65535","farm":66,"title":"MG_3699","ispublic":1,"isfriend":0,"isfamily":0,"id":"494434445621","owner":"8185859@N08","secret":"17470ed3b7","server":"65535","farm":66,"title":"StVu00f0psef","ispublic":1,"isfriend":0,"isfamily":0,"id":"49443621242","owner":"119022625@N06","secret":"f3e3637b33","server":"65535","farm":66,"title":"523","ispublic":1,"isfriend":0,"isfamily":0,"id":"49443616467","owner":"154247468@N08","secret":"d0c489c0b","server":"65535","farm":66,"title":"K_1_1312","ispublic":1,"isfriend":0,"isfamily":0,"id":"49442909128","owner":"154247468@N08","secret":"1a5ba84b7a","server":"65535","farm":66,"title":"K_1_1325","ispublic":1,"isfriend":0,"isfamily":0,"id":"49442888223","owner":"87642046@N00","secret":"106cdf1bf9","server":"65535","farm":66,"title":"Day 20 - Brothers","ispublic":1,"isfriend":0,"isfamily":0,"id":"4944359397","owner":"87642046@N00","secret":"3f7bc634ae","server":"65535","farm":66,"title":"Day 19 - Bandit","ispublic":1,"isfriend":0,"isfamily":0,"id":"4944359397","owner":"87642046@N00","secret":"bc21aa02fe","server":"65535","farm":66,"title":"Day 21 - Bear","ispublic":1,"isfriend":0,"isfamily":0},"stat":"ck"}] |            |            |            |

**Fragment 2:** Fragment contenant une ListView qui permet d'afficher le résultat de la requête avec les titres des images(10 mais on peut changer paramètre pour charger davantage des images) ListTitleImages.java, fragment\_list\_title\_images.xml



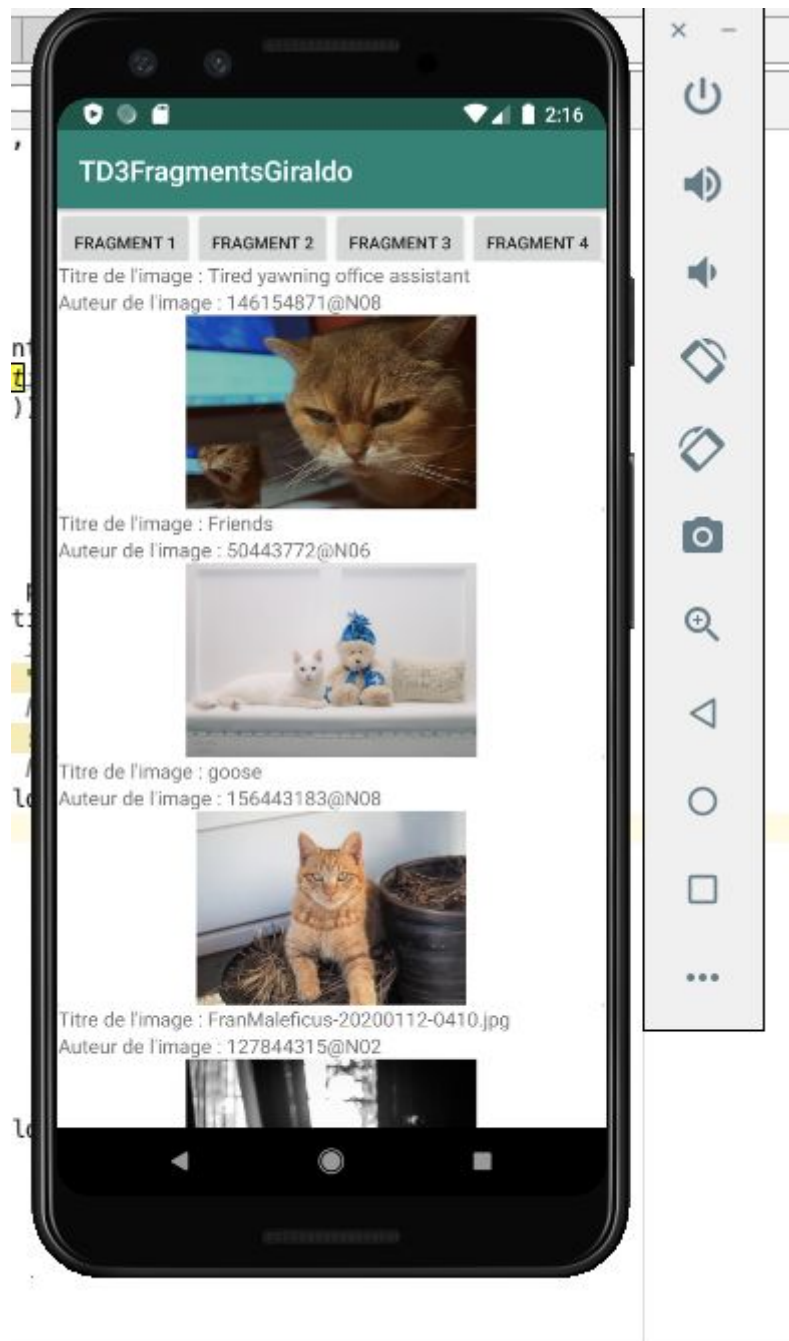
**Fragment 3 :** Fragment contenant la liste des images(AsyncBitmapDownloader.java, BitmapAdapter.java, ListImages.java, fragment\_list\_images.xml)



pour l'affichage d'une image à partir d'une URL j'ai utilisé la librairie Picasso parce qu'il propose:

- Gestion du cache d'une image
- Réduction de la taille d'une image
- Insertion de l'image dans une ImageView
- Téléchargement de l'image de façon asynchrone depuis une URL

**Fragment 4:** RecyclerView fait avec Volley et Picasso FragmentComplex.java, ImageFlickr.java, AdapterImages.java, fragment\_complex.xml, image\_flickr.xml



## Exercice 6 et 7

```
private void jsonParserFlickr(){
 //http://farm{farm-id}.staticflickr.com/{server-id}/{id}_{secret}.jpg
 mQueue = Volley.newRequestQueue(getActivity());
 JsonObjectRequest request = new JsonObjectRequest(Request.Method.GET, flickr_reponse, JsonRequest: null,
 (response) -> {
 try {
 JSONObject jsonArray = response.getJSONObject("photos");
 JSONArray Arrayphotos = jsonArray.getJSONArray(name: "photo");
 for (int i = 0; i < Arrayphotos.length(); i++) {
 JSONObject image = Arrayphotos.getJSONObject(i);

 String farm_id = image.getString(name: "farm");
 String server_id = image.getString(name: "server");
 String id = image.getString(name: "id");
 String secret = image.getString(name: "secret");
 String url = "http://farm"+farm_id+".staticflickr.com/"+server_id+"/"+id+"_"+secret+".jpg";
 ImageFlickr imageF = new ImageFlickr(image.getString(name: "title"),
 image.getString(name: "owner"),
 image.getString(name: "id"),
 image.getString(name: "secret"),
 image.getString(name: "server"),
 url,
 image.getString(name: "farm"));
 ImagesFlickr.add(imageF);
 adapterComplex = new AdapterImages(ImagesFlickr, getContext());
 }

 recyclerViewImages.setAdapter(adapterComplex);
 // Log.i("SGB", "inside Adapter Images constructor"+ adapterComplex.getItemCount());
 } catch (JSONException e) {
 e.printStackTrace();
 }
 }, new Response.ErrorListener() {
 @Override
 public void onErrorResponse(VolleyError error) {
 error.printStackTrace();
 }
 });
};
```

FragmentComplex > jsonParserFlickr() > new Listener > onResponse()

Build Profiler 4: Run Event Log



```

package com.example.td3fragmentsgiraldo;

public class ImageFlickr {
 private String title;
 private String author;
 private String id;
 private String secret;
 private String server;
 private String url;
 private String farm;

 public ImageFlickr(String title,String author, String id, String secret, String server, String url, String farm){
 this.title= title;
 this.author=author;
 this.id=id;
 this.secret=secret;
 this.server=server;
 this.url=url;
 this.farm=farm;
 }

 public String getTitle() { return title; }

 public void setTitle(String title) { this.title = title; }

 public String getAuthor() { return author; }

 public void setAuthor(String author) { this.author = author; }

 public String getId() { return id; }

 public void setId(String id) { this.id = id; }

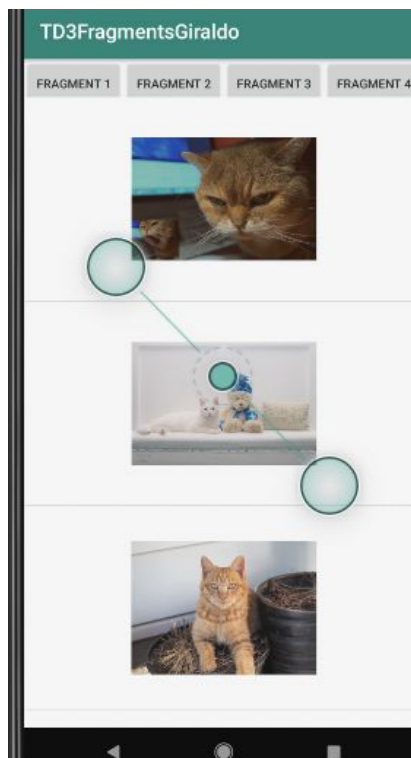
 public String getSecret() { return secret; }

 public void setSecret(String secret) {
 this.secret = secret;
 }

 public String getServer() {
 return server;
 }
}

```

## Exercice 8 et 9



```
1 package com.example.td3fragmentsgirald;
2
3 import android.graphics.Bitmap;
4 import android.os.AsyncTask;
5 import android.util.Log;
6
7 import com.squareup.picasso.Picasso;
8 import java.io.IOException;
9
10
11 public class AsyncBitmapDownloader extends AsyncTask<String, Void, Bitmap> {
12
13 BitmapAdapter adapter_ = null;
14 Bitmap bm = null;
15
16 public AsyncBitmapDownloader(BitmapAdapter adapter) { adapter_ = adapter; }
17
18 @Override
19 protected Bitmap doInBackground(String... strings) {
20
21 try {
22 bm = Picasso.get().load(strings[0]).get();
23 } catch (IOException e) {
24 e.printStackTrace();
25 }
26 return bm;
27 }
28
29 @Override
30 protected void onPostExecute(Bitmap bitmap) {
31 Log.i(tag: "SGB", msg: "Image received !");
32 adapter_.add(bitmap);
33 adapter_.notifyDataSetChanged();
34 }
35 }
36
37 }
```