# EPICTURE DOCUMENTATION

by Simon Bernabeu and Guillaume Corbet

## Summary

## Compile

We use gradle to compile our project, gradle will install all the dependencies we need to have to make our project run.

```
implementation 'androidx.core:core-ktx:1.1.0'
implementation 'androidx.appcompat:appcompat:1.1.0'
implementation 'com.google.android.material:material:1.0.0'
implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
implementation 'androidx.navigation:navigation-fragment-ktx:2.1.0'
implementation 'androidx.navigation:navigation-ui-ktx:2.1.0'
implementation "com.squareup.retrofit2:retrofit:2.3.0"
implementation "com.squareup.retrofit2:adapter-rxjava2:2.3.0"
implementation "com.squareup.retrofit2:converter-gson:2.3.0"
implementation "com.squareup.retrofit2:converter-moshi:2.3.0"
implementation 'com.squareup.picasso:picasso:2.5.2'
implementation 'com.android.support:recyclerview-v7'
implementation 'com.android.support:cardview-v7'
implementation 'androidx.recyclerview:recyclerview:1.1.0'
implementation 'androidx.legacy:legacy-support-v4:1.0.0'
```

## Retrofit

We use Retrofit library to make all the request to Imgur API.

We have an Imgur interface with all the request and the param we want.

```
@GET("/3/gallery/{section}/{sort}/{pages}/week")
fun getGallery(
    @Header("Authorization") authHeader: String,
    @Path("section") section: String,
    @Path("sort") sort: String,
    @Path("pages") pages: Int,
    @Query("showViral") showViral: Boolean,
    @Query("mature") mature: Boolean,
    @Query("album_previews") album_previews: Boolean
): Call<ModelImgur.SearchData>
```

We also have an class with all the model of response of imgurApi.

```
data class DataSearch(
    val id: String,
    val title: String,
    val description: String,
    val datetime: Int,
    val type: String,
    val animated: Boolean,
    val width: Int,
    val height: Int,
    val size: Int,
    val views: Int,
    val bandwidth: Long,
    val vote: Int,
    val favorite: Boolean,
    val nsfw: Boolean,
    val section: String,
    val account_url: String,
    val account_id: String,
    val is_ad: Boolean,
    val in_most_viral: Boolean,
    val has_sound: Boolean,
    val tags: List<DataTags>,
    val images: List<ImageData>,
    val link: String
)
```

Retrofit will just send the request, read the response and put in the data class all the info he have from the response.

## Picasso

We use Picasso library to display image, it just need a link to display an image.

> Picasso.with(activity?.applicationContext).load(link).into(holder.photo)

## Imgur

We use these request of Imgur API.

- Upload image

  > https://api.imgur.com/3/upload

- Get images Gallery

  > https://api.imgur.com/3/gallery/{section}/{sort}/{pages}/week

- Get account posted images

  > https://api.imgur.com//3/account/me/images

- Get Favorite images

  > https://api.imgur.com/3/account/{username}/gallery_favorites/{page}/{favoriteSort}

- Favorite image

  > https://api.imgur.com/post/v1/posts/{imageHash}/favorite