Mobile Application Development

Produced by

David Drohan (ddrohan@wit.ie)

Department of Computing & Mathematics
Waterford Institute of Technology
http://www.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Android & Firebase
Part 4

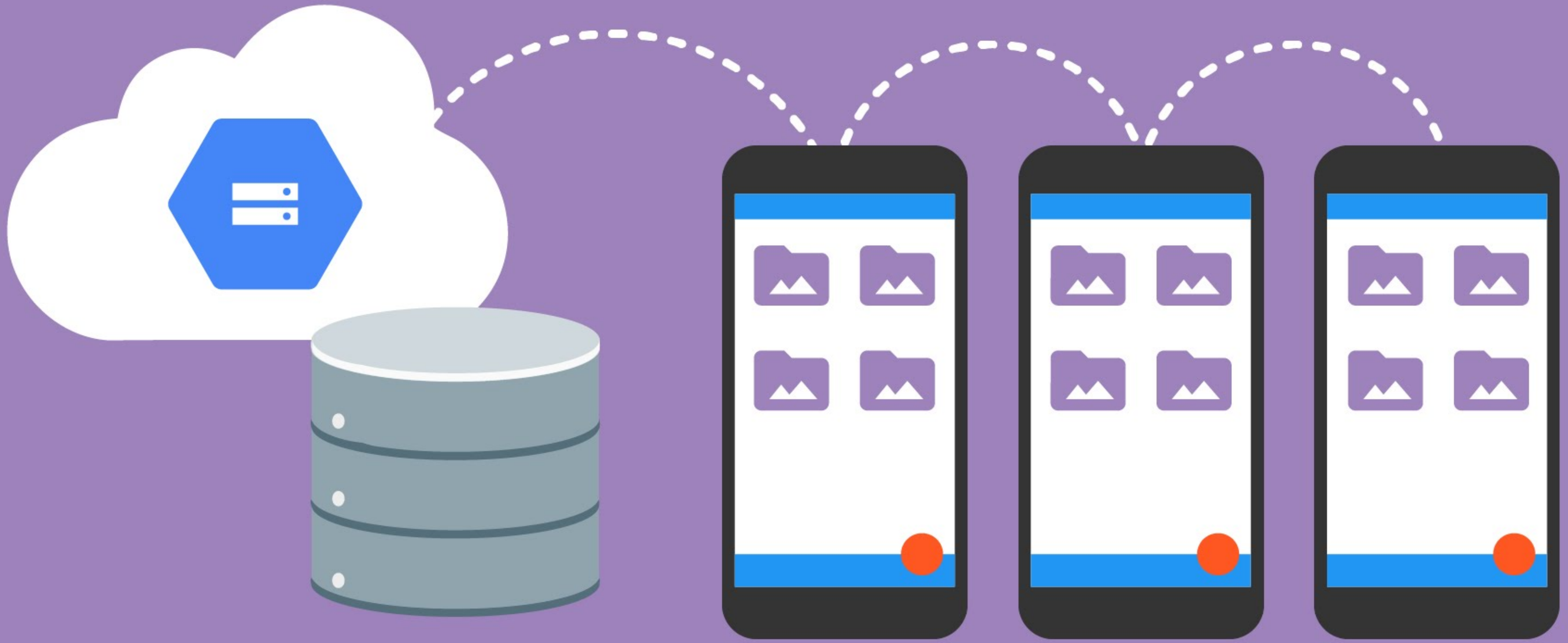
Firebase Integration

Agenda

- ❑ Firebase history
- ❑ The all new Firebase
- ❑ Real-time database
- ❑ Authentication
- ❑ Storage
- ❑ Remote config
- ❑ Hosting
- ❑ Crash reporting
- ❑ Test lab
- ❑ Firebase cloud messaging
- ❑ Dynamic links
- ❑ App indexing
- ❑ Analytics
- ❑ CoffeeMate Highlights & Demos along the way…

Agenda

- ❑ Firebase history
- ❑ The all new Firebase
- ❑ Real-time database
- ❑ Authentication
- ❑ Storage
- ❑ Remote config
- ❑ Hosting
- ❑ Crash reporting
- ❑ Test lab
- ❑ Firebase cloud messaging
- ❑ Dynamic links
- ❑ App indexing
- ❑ Analytics
- ❑ CoffeeMate Highlights & Demos along the way…



Storage

Sync files and folders seamlessly…

Firebase Storage

- Store user’s image, audio, video and other content in the cloud easily without worrying about the network quality.
- Firebase adds Google security to file uploads and downloads.
- Backed by Google Cloud Storage.
- Petabyte scalability available if your app goes viral.

Upload Files To Firebase

- ❑ After adding gradle dependencies. Get Storage reference
  - StorageReference storage = FirebaseStorage.getInstance().getReferenceFromUrl(‘gs:<bucket>’);
  - storage = storage.child(‘myimages/user1234pic.jpg’);
- ❑ Easily change reference.
  - storage.getParent(), getRoot(), getPath(), getName(), getBucket() etc.
- ❑ Upload from memory, stream or from SD card
  - storage.putBytes(imageData); // as byte array.
  - storage.putStream(stream); // as input stream
  - Storage.putFile(fileURI): / Uri of the local file to be uploaded

gs://coffeematefbi.appspot.com

More On File Upload

- Put file metadata
  - StorageMetadata metadata = new StorageMetadata.Builder().setContentType("image/jpg").build()
  - storage.putFile(file, metadata)
- Manage uploads
  - UploadTask task = storage.putFile(file, metadata)
  - task.pause(), resume(), cancel()…
- Monitor upload
  - OnProgressListener, OnPausedListener, OnSuccessListener, OnFailuerListener

Download Files From Firebase

- Download to a local file
  - `File localFile = File.createTempFile(‘mefile’,’jpg’);`
  - `storage.getFile(localFile).addOnSuccessListener(…);`
- Download to memory
  - `Final long ONE_MEG = 1024*1024;`

  -
    ```
    storage.getBytes(ONE_MEG).addOnSuccessListener(new OnSuccessListener<byte[]>() {
                public void onSuccess(bytes[] bytes){
                            }
                } );
    ```
- Point StorageReference to desired file and call getMetadata() to get metadata.

Firebase Storage

Quick Example…

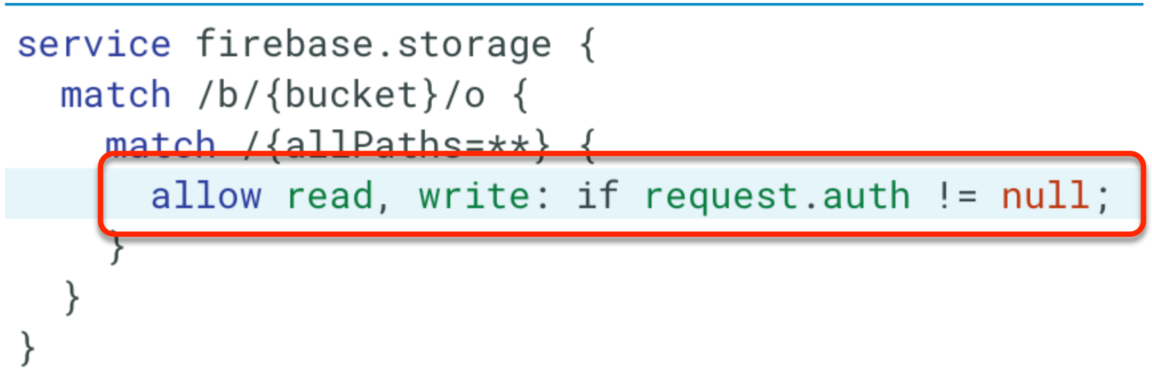
https://github.com/kotlintpoint/Firebase-Storage-Upload-Download

FirebaseStorageEx App *

- On first look, a very basic app to Upload/Download a file
- What it actually does is
  - Using System Intents, allow you to select a file from any source on your device, including your google drive
  - Allow you to name that file and upload it to your Firebase Storage space.
  - Allow you to download a file once the correct filename is supplied from your Firebase Storage and display it using third party apis (glide/picasso).

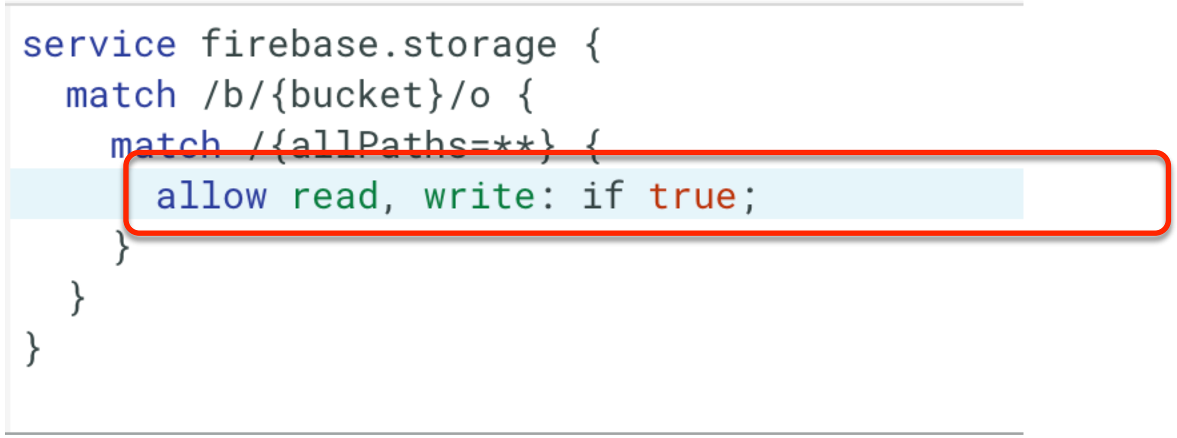
FirebaseStorageEx App - Setup *

Update your Firebase Storage rules to allow reading and writing without authentication (for demo purposes)



```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

BEFORE

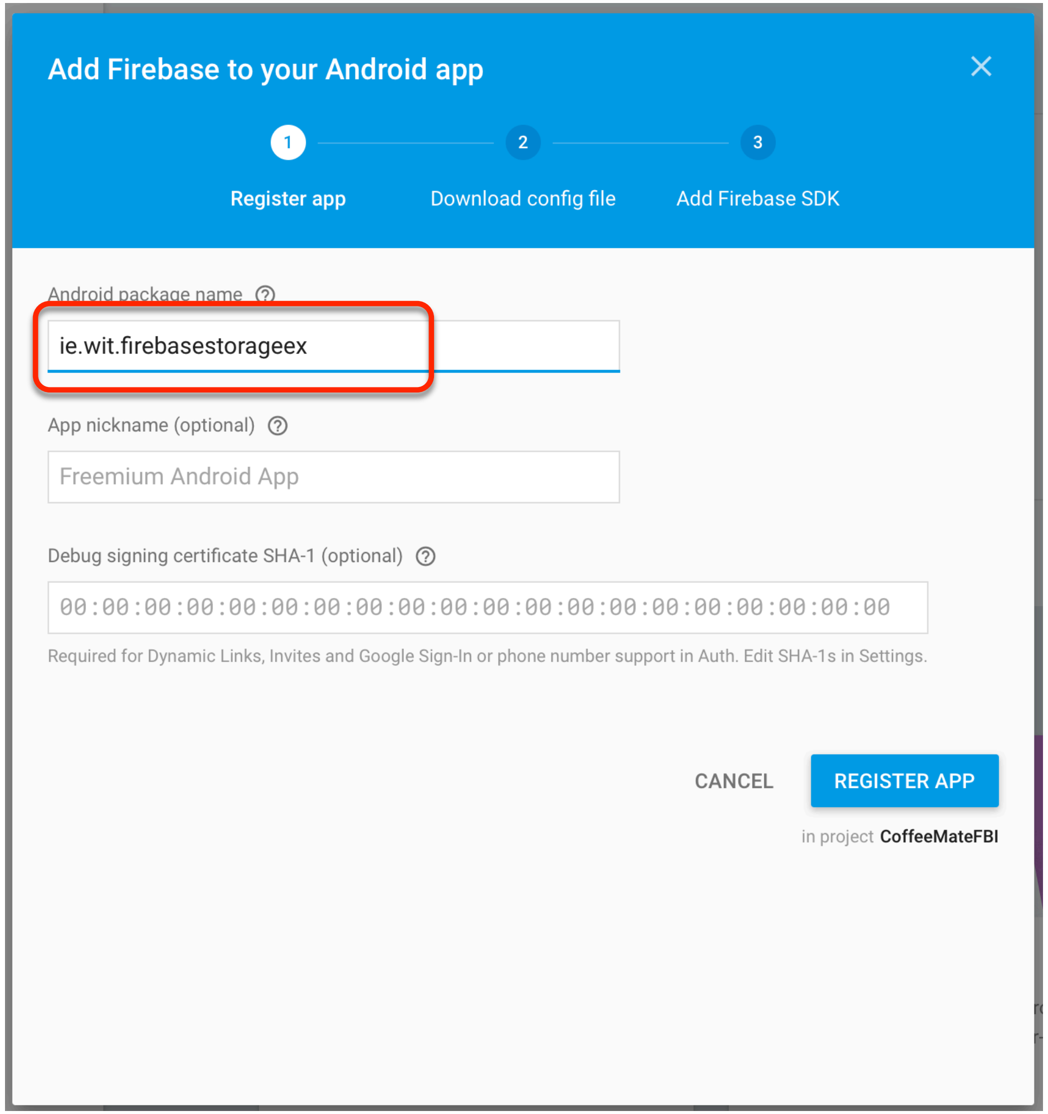


```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if true;
    }
  }
}
```

AFTER

FirebaseStorageEx App - Setup *

Don’t forget to register your app with your Firebase Project



Add Firebase to your Android app

Android package name: ie.wit.firebasestorageex

App nickname (optional): Freemium Android App

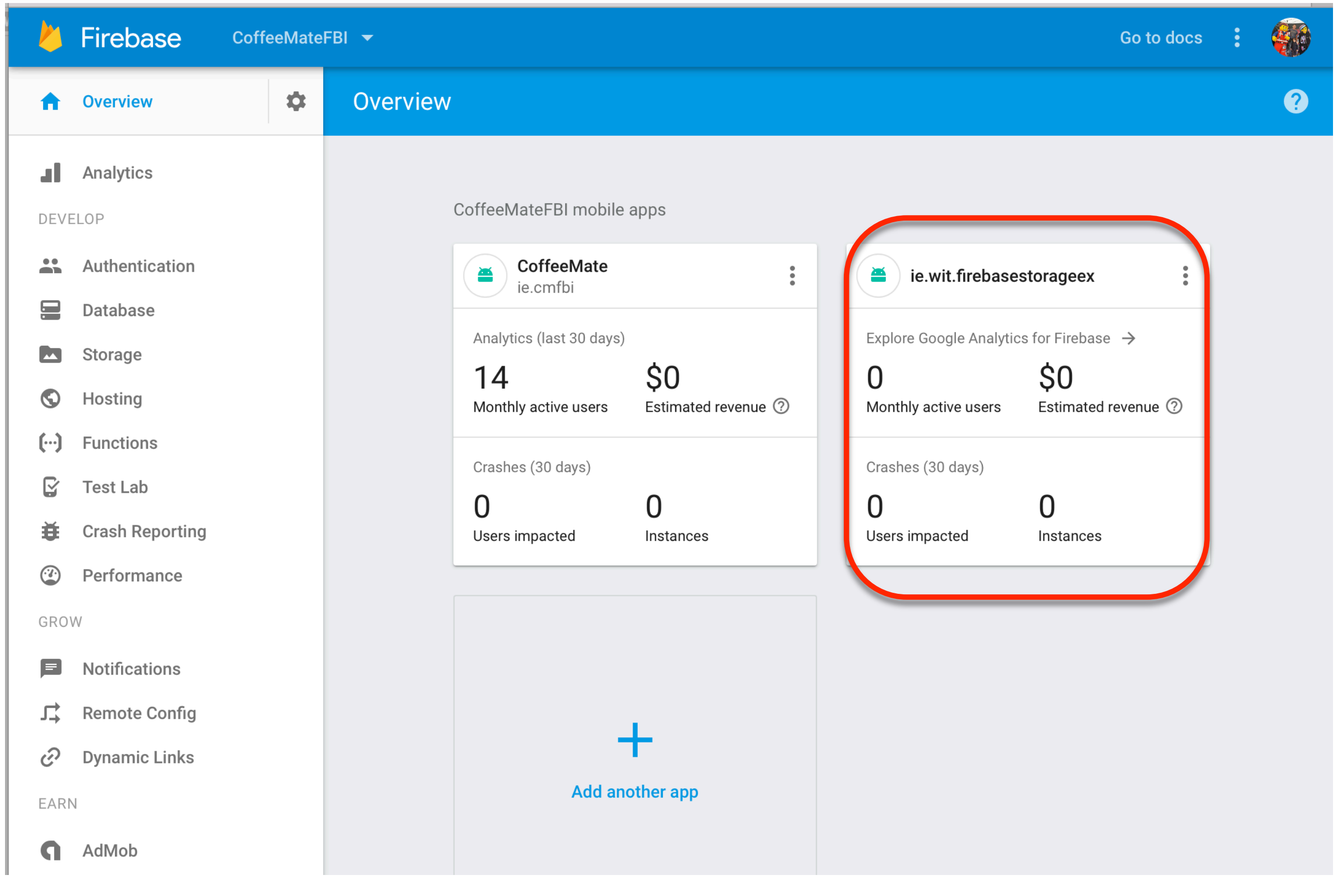
Debug signing certificate SHA-1 (optional): 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Required for Dynamic Links, Invites and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

in project CoffeeMateFBI

FirebaseStorageEx App - Setup *

Don’t forget to register your app with your Firebase Project



CoffeeMateFBI mobile apps

CoffeeMate
ie.cmfbi

Analytics (last 30 days)

14 Monthly active users

$0 Estimated revenue

Crashes (30 days)

0 Users impacted

0 Instances

ie.wit.firebasestorageex

Explore Google Analytics for Firebase →

0 Monthly active users

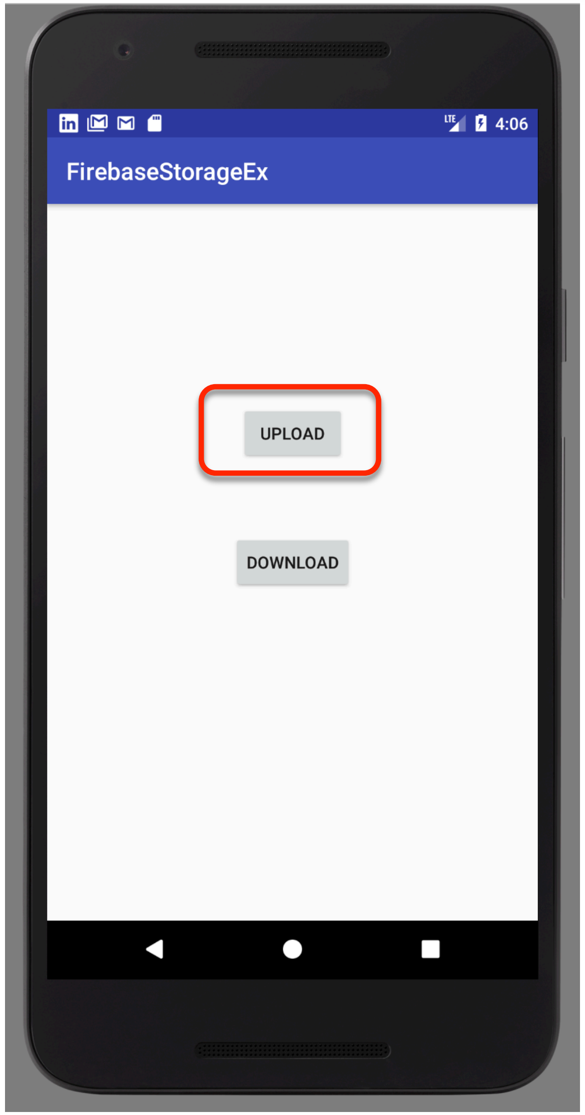
$0 Estimated revenue

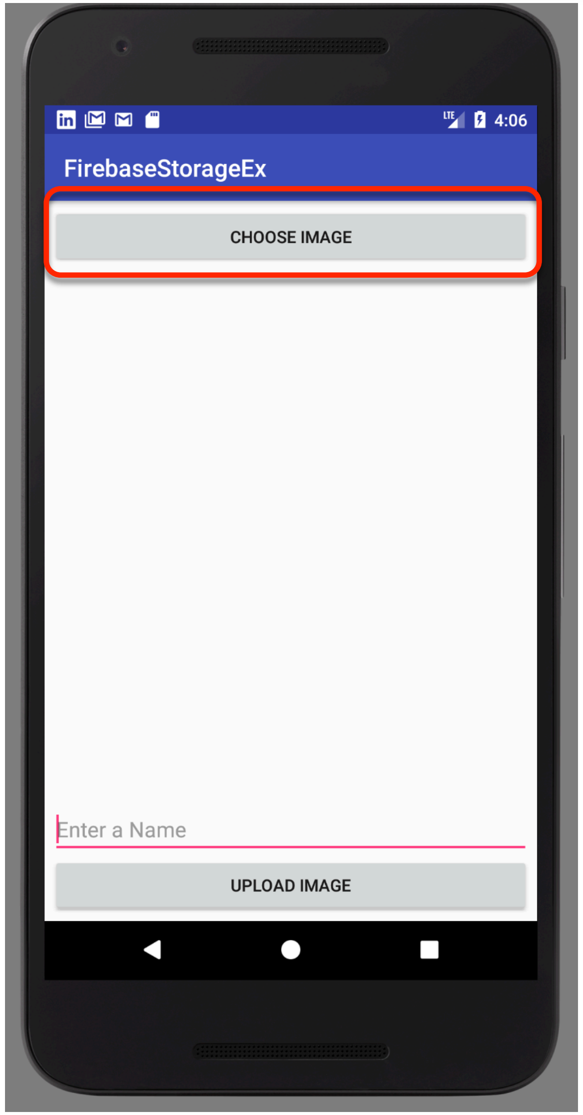
Crashes (30 days)

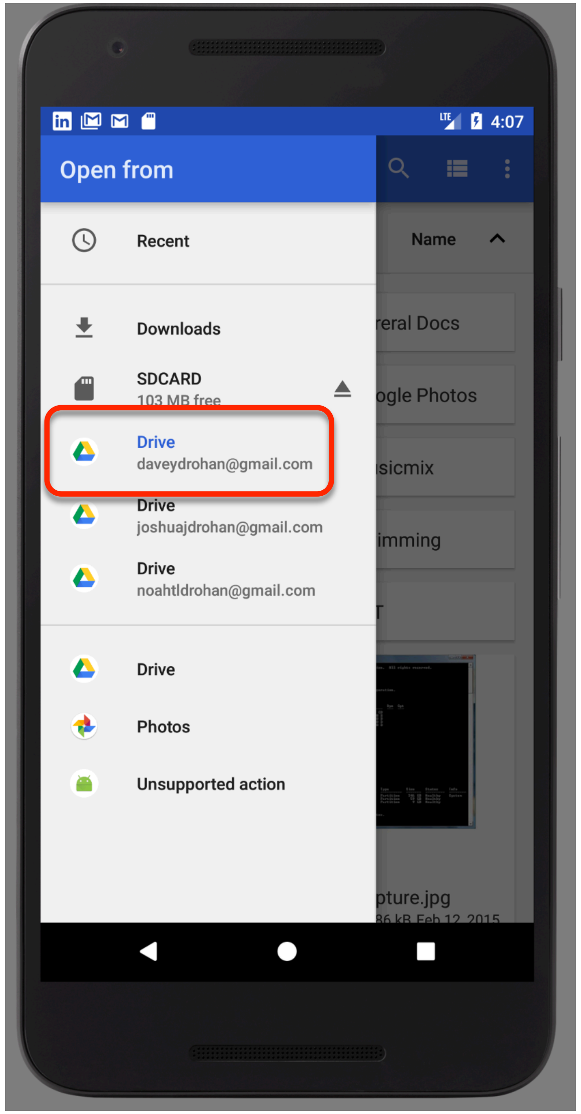
0 Users impacted

0 Instances

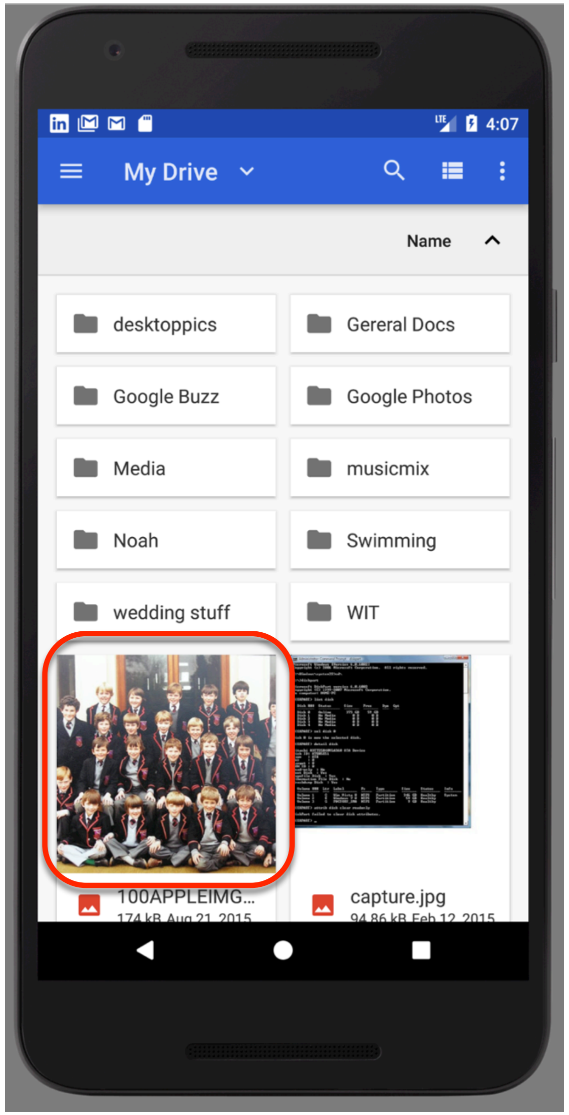
FirebaseStorageEx App - Upload *

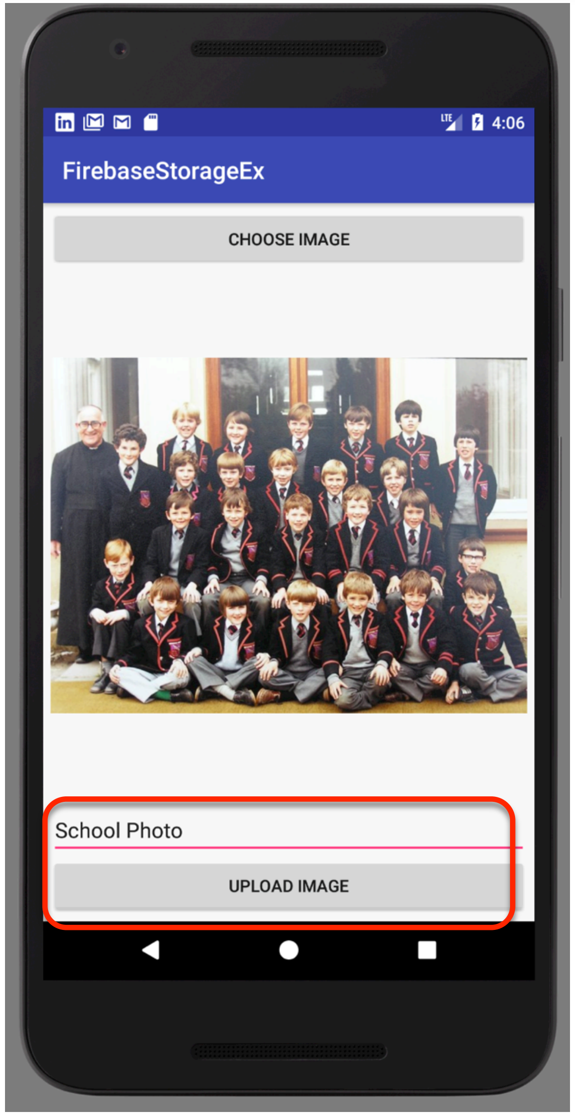




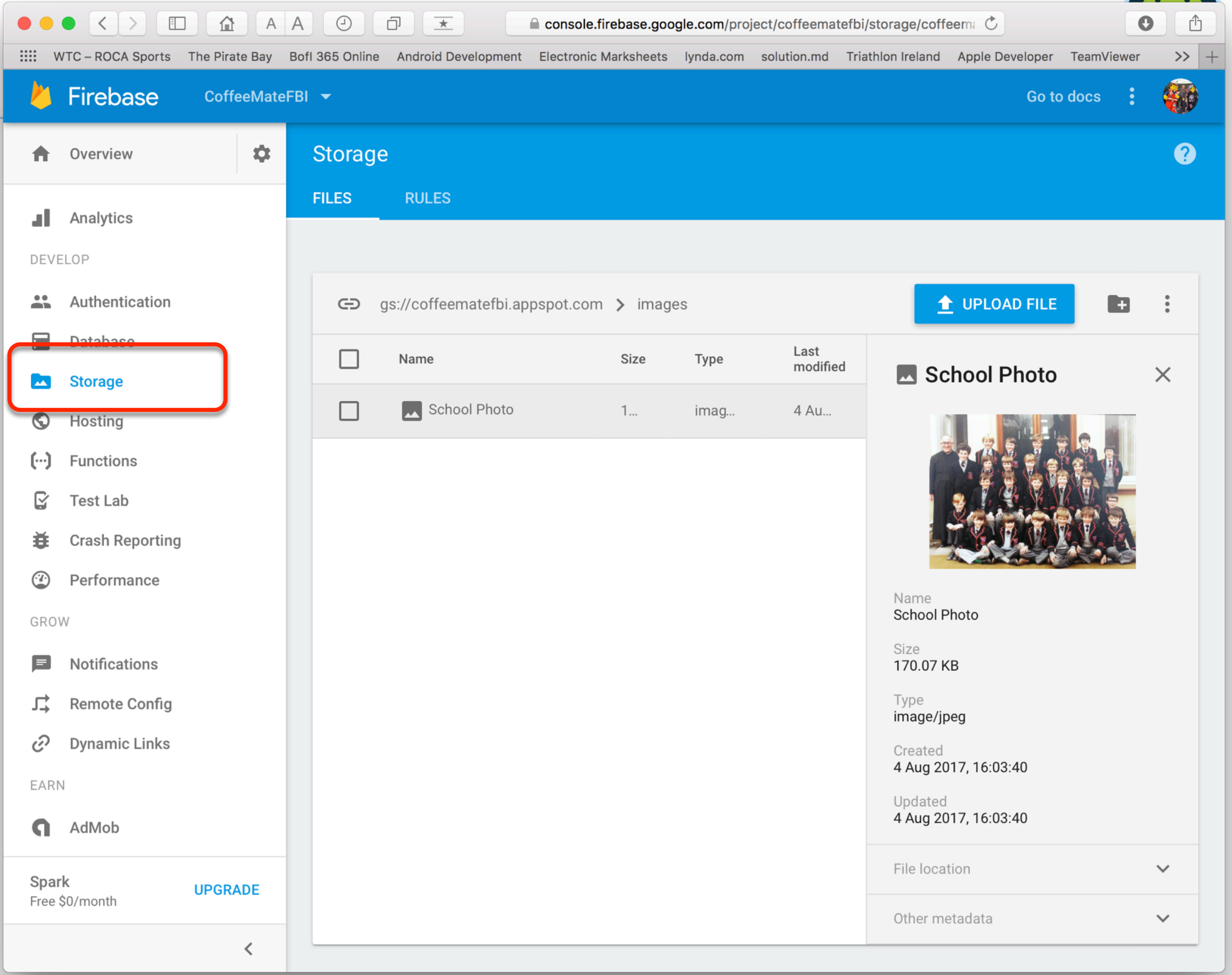


FirebaseStorageEx App - Upload *





FirebaseStorageEx App – Upload



□	Name	Size	Type	Last modified
□	School Photo	1...	imag...	4 Au...

School Photo

Name
School Photo

Size
170.07 KB

Type
image/jpeg

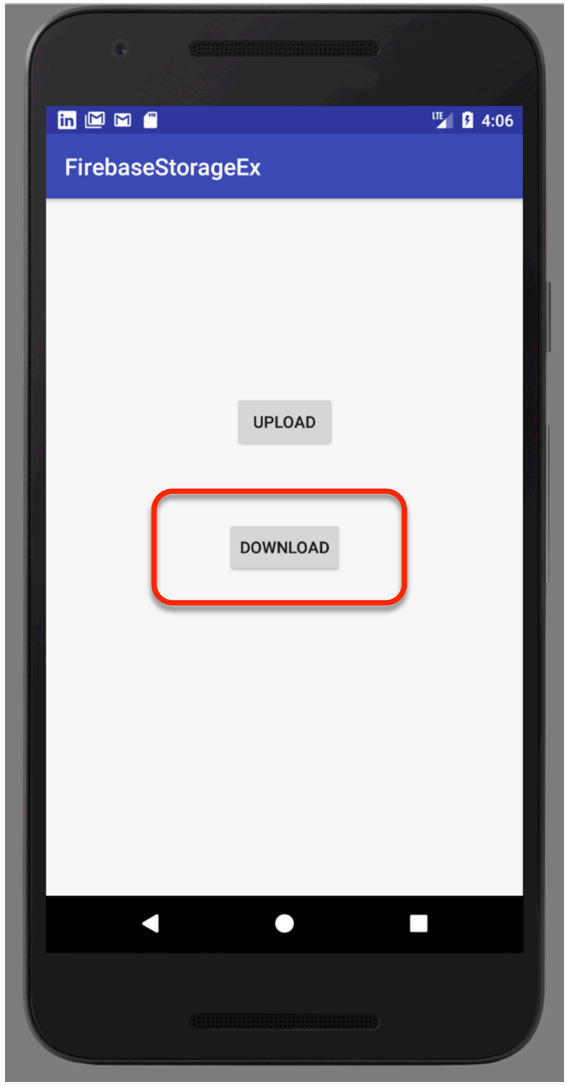
Created
4 Aug 2017, 16:03:40

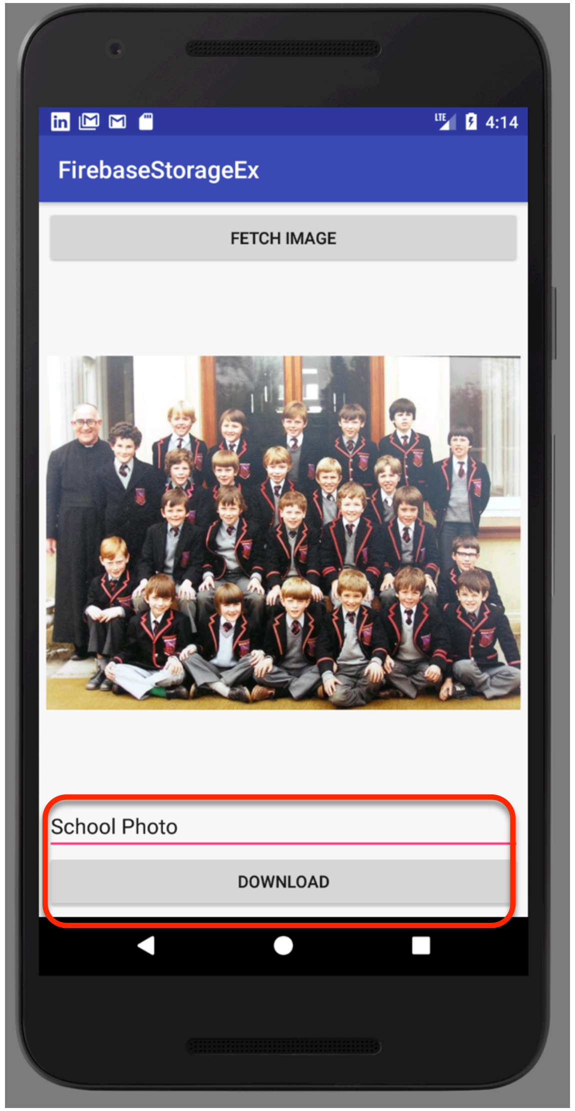
Updated
4 Aug 2017, 16:03:40

File location

Other metadata

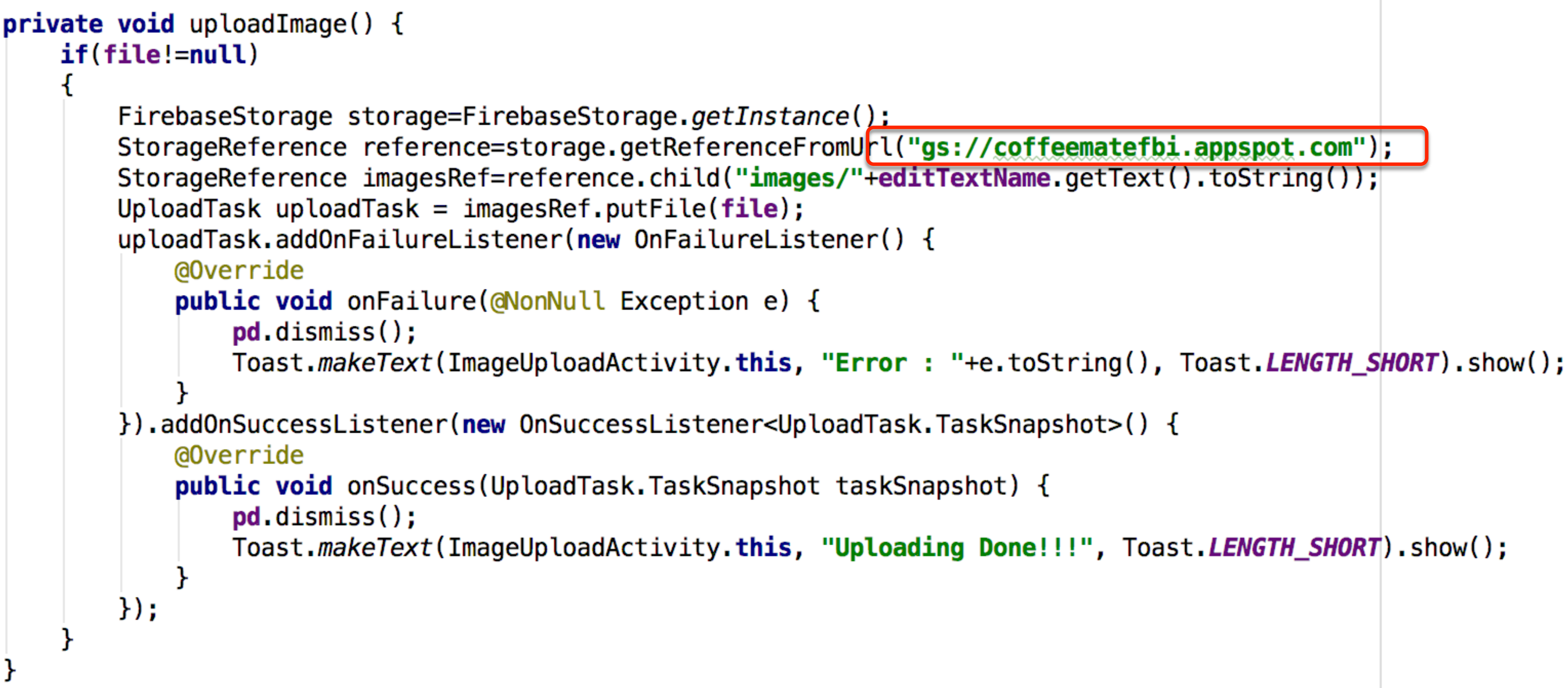
FirebaseStorageEx App - Download *





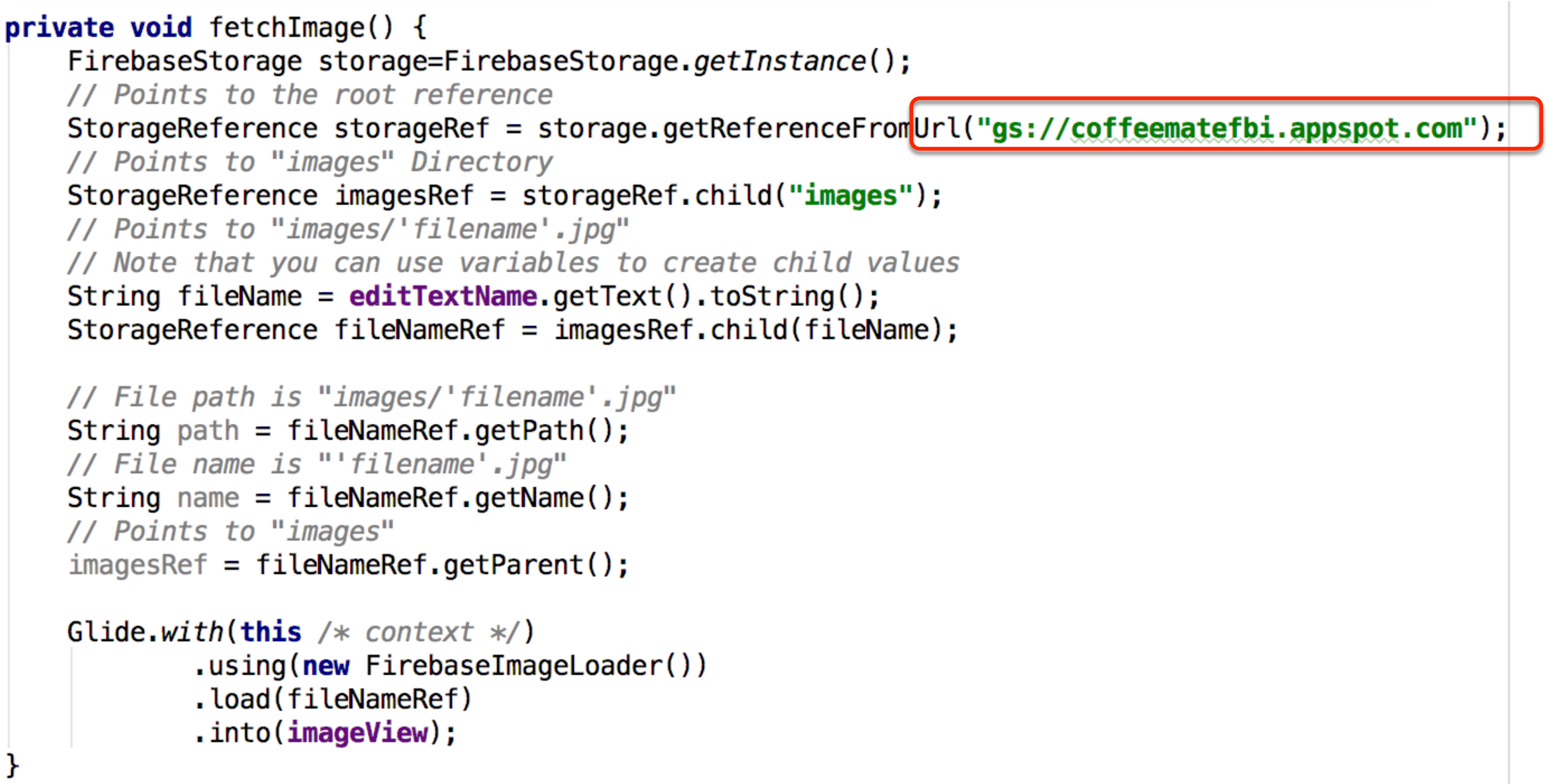
School Photo

FirebaseStorageEx App – uploadImage( )*



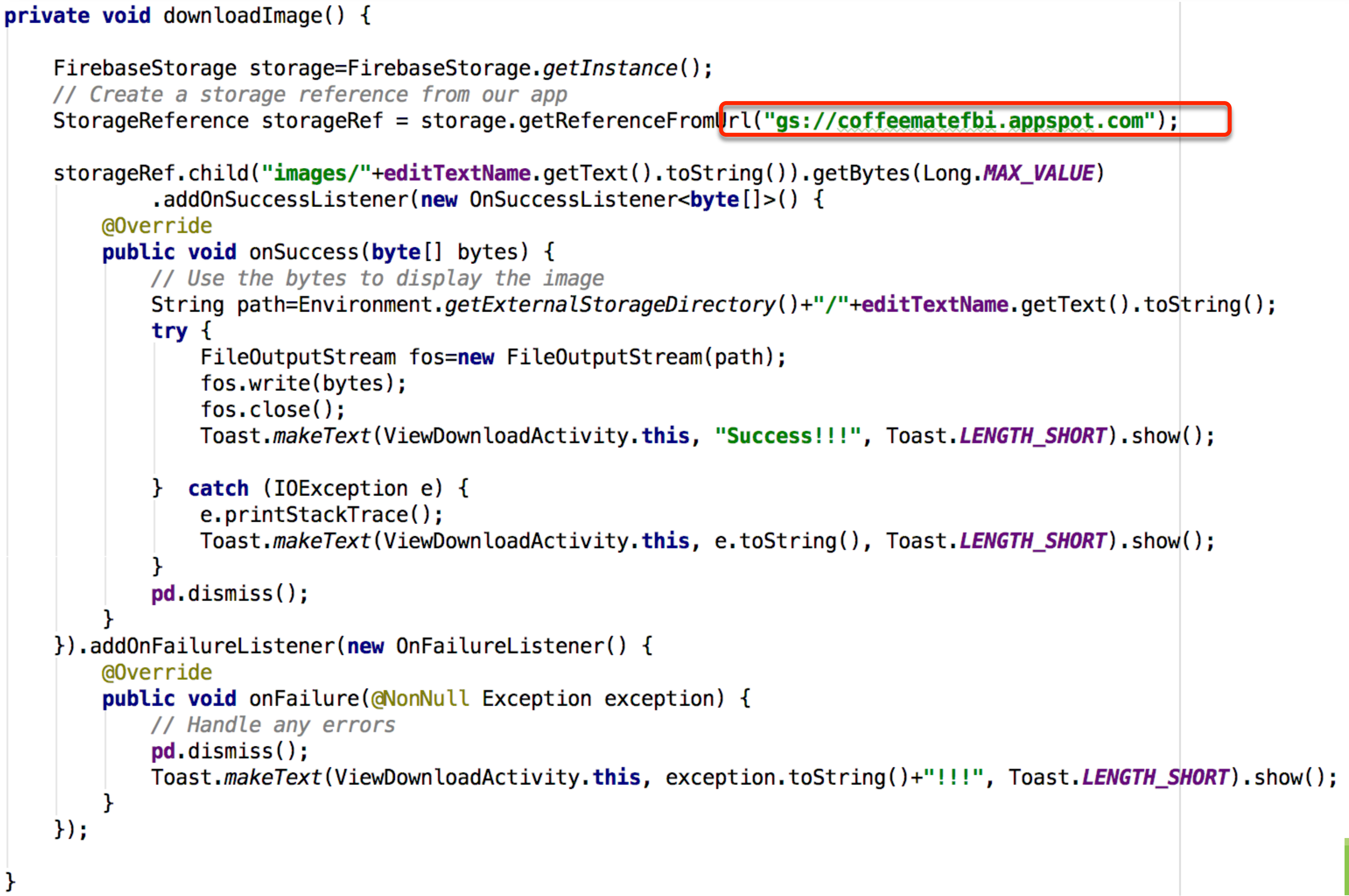
```java
private void uploadImage() {
    if(file!=null)
    {
        FirebaseStorage storage=FirebaseStorage.getInstance();
        StorageReference reference=storage.getReferenceFromUrl("gs://coffeematefbi.appspot.com");
        StorageReference imagesRef=reference.child("images/"+editTextName.getText().toString());
        UploadTask uploadTask = imagesRef.putFile(file);
        uploadTask.addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                pd.dismiss();
                Toast.makeText(ImageUploadActivity.this, "Error : "+e.toString(), Toast.LENGTH_SHORT).show();
            }
        }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                pd.dismiss();
                Toast.makeText(ImageUploadActivity.this, "Uploading Done!!!", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

FirebaseStorageEx App - fetchImage( ) *



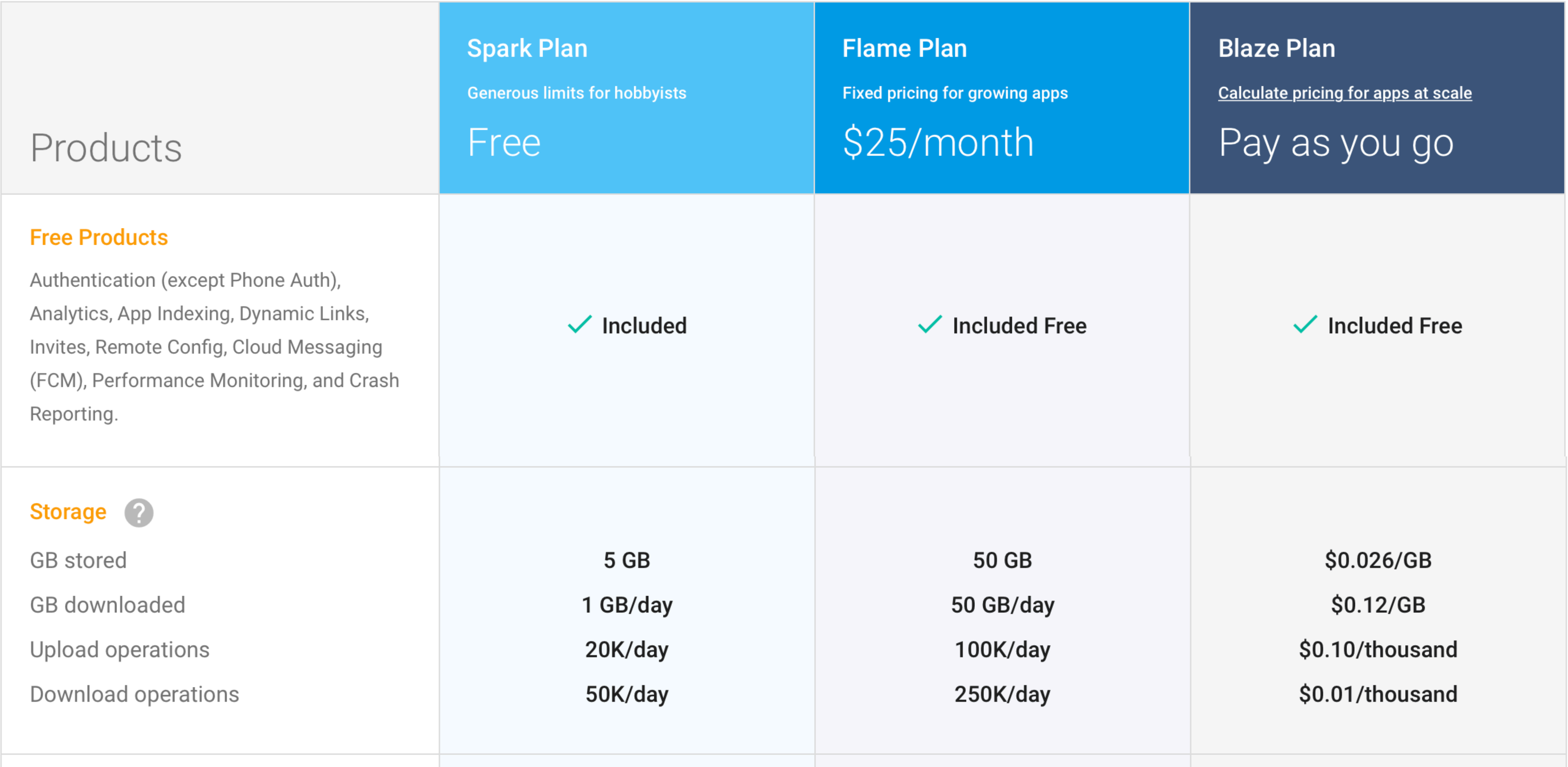
```
private void fetchImage() {
    FirebaseStorage storage=FirebaseStorage.getInstance();
    // Points to the root reference
    StorageReference storageRef = storage.getReferenceFromUrl("gs://coffeematefbi.appspot.com");
    // Points to "images" Directory
    StorageReference imagesRef = storageRef.child("images");
    // Points to "images/'filename'.jpg"
    // Note that you can use variables to create child values
    String fileName = editTextName.getText().toString();
    StorageReference fileNameRef = imagesRef.child(fileName);

    // File path is "images/'filename'.jpg"
    String path = fileNameRef.getPath();
    // File name is "'filename'.jpg"
    String name = fileNameRef.getName();
    // Points to "images"
    imagesRef = fileNameRef.getParent();

    Glide.with(this /* context */)
            .using(new FirebaseImageLoader())
            .load(fileNameRef)
            .into(imageView);
}
```

FirebaseStorageEx App - downloadImage( ) *



```
private void downloadImage() {

    FirebaseStorage storage=FirebaseStorage.getInstance();
    // Create a storage reference from our app
    StorageReference storageRef = storage.getReferenceFromUrl("gs://coffeematefbi.appspot.com");

    storageRef.child("images/"+editTextName.getText().toString()).getBytes(Long.MAX_VALUE)
            .addOnSuccessListener(new OnSuccessListener<byte[]>() {
        @Override
        public void onSuccess(byte[] bytes) {
            // Use the bytes to display the image
            String path=Environment.getExternalStorageDirectory()+"/"+editTextName.getText().toString();
            try {
                FileOutputStream fos=new FileOutputStream(path);
                fos.write(bytes);
                fos.close();
                Toast.makeText(ViewDownloadActivity.this, "Success!!!", Toast.LENGTH_SHORT).show();

            }  catch (IOException e) {
                e.printStackTrace();
                Toast.makeText(ViewDownloadActivity.this, e.toString(), Toast.LENGTH_SHORT).show();
            }
            pd.dismiss();
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception exception) {
            // Handle any errors
            pd.dismiss();
            Toast.makeText(ViewDownloadActivity.this, exception.toString()+"!!!", Toast.LENGTH_SHORT).show();
        }
    });

}
```

Pricing



Products	Spark Plan Generous limits for hobbyists Free	Flame Plan Fixed pricing for growing apps $25/month	Blaze Plan Calculate pricing for apps at scale Pay as you go
Free Products Authentication (except Phone Auth), Analytics, App Indexing, Dynamic Links, Invites, Remote Config, Cloud Messaging (FCM), Performance Monitoring, and Crash Reporting.	✓ Included	✓ Included Free	✓ Included Free
Storage ?			
GB stored	5 GB	50 GB	$0.026/GB
GB downloaded	1 GB/day	50 GB/day	$0.12/GB
Upload operations	20K/day	100K/day	$0.10/thousand
Download operations	50K/day	250K/day	$0.01/thousand

Thanks!

Some important points though…

- ❑ Do not think RDBMS, think JSON. How data should be structured is very important.
- ❑ Firebase has a recycler view, that integrates with real time database smoothly without any listeners. (FirebaseUI)
- ❑ Test lab which is available in paid plan (Blaze), is an amazing feature for testing your app on different real and virtual devices (next section)
- ❑ Set developer mode to true when testing Remote Config (next section).

References & Links

- ❑ Presentation by Kaushal Dhruw & Shakti Moyal 2016
- ❑ https://firebase.google.com
- ❑ Demo app available at https://goo.gl/WBP5fR

Questions?