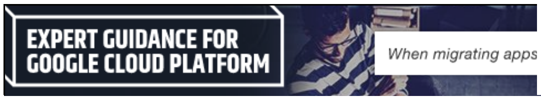


# Firebase Storage – Upload Data from Memory, Local File, Stream | Android

Firebase Cloud Storage helps us upload and share rich content data. Data is stored in a Google Cloud Storage bucket. With Firebase, we can perform robust operations (download/upload) regardless of network quality with strong security (Cloud Storage integrates with Firebase Authentication) and high scalability.

In this tutorial, we're gonna look at ways to upload data from Memory, Local file, Stream in an Android App using Firebase Storage.

More practice:
– Firebase Storage – Download Files to Memory, Local File | Android
– Firebase Storage – Get List of Files example – Image List | Android

Contents [hide]

### I. How to upload file

To use the Firebase Storage to upload data, we need:
– add Firebase to Android App & enable Firebase Auth
– create a reference to the full path of the file, including the file name
– upload data using `putBytes()` for in-memory data, `putStream()` for stream data, `putFile()` for local file.

### 0. Add Firebase to Android App

### 0.1 Create Firebase Project and Add Firebase Config file

Follow this guide to create Firebase Project and generate google-services.json file and move it into your Android App root directory. You don't need to get SHA-1 Key in this example.

### 0.2 Add dependencies

– build.gradle (project-level):

```
buildscript {
    // ...
    dependencies {
        // ...
        classpath 'com.google.gms:google-services:3.1.0'
    }
}
```

– build.gradle (App-level):

```
dependencies {
    // ...
    compile 'com.google.firebase:firebase-auth:11.0.2'
    compile 'com.google.firebase:firebase-storage:11.0.2'
}

apply plugin: 'com.google.gms.google-services'
```

### 1. Enable Firebase Auth

By default, only authenticated users can read or write data, so we need Firebase Authentication for next step.
Go to Your Firebase Project Console -> Authentication -> SIGN-IN METHOD -> Enable Email/Password.

To do without setting up Authentication, we can change the rules in the Firebase Console -> choose Project -> Storage section on the left -> Rules tab:

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

### 2. Create a Reference

We cannot upload data with a reference to the root of Google Cloud Storage bucket. Reference must point to a child URL:

```
// Create a storage reference from our app
StorageReference storageRef = storage.getReference();

// Create a reference to "javasampleapproach.jpg"
StorageReference jsaRef = storageRef.child("mountains.jpg");

// Create a reference to 'images/javasampleapproach.jpg'
StorageReference jsaImagesRef = storageRef.child("images/javasampleapproach.jpg");
```

If the file names are the same:

```
// true
mountainsRef.getName().equals(jsaImagesRef.getName());

// false
mountainsRef.getPath().equals(jsaImagesRef.getPath());
```

### 3. Upload Data using

### 3.1 putBytes()

```
byte[] data = ...;

// StorageReference fileRef = ...;
fileRef.putBytes(data)
```

```
        .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                // Uri: taskSnapshot.getDownloadUrl());
                // Name: taskSnapshot.getMetadata().getName());
                // Path: taskSnapshot.getMetadata().getPath());
                // Size: taskSnapshot.getMetadata().getSizeBytes());
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception exception) {
                // Handle unsuccessful uploads
            }
        })
        .addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onProgress(UploadTask.TaskSnapshot taskSnapshot) {
                // taskSnapshot.getBytesTransferred()
                // taskSnapshot.getTotalByteCount();
            }
        })
        .addOnPausedListener(new OnPausedListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onPaused(UploadTask.TaskSnapshot taskSnapshot) {
                // Upload is paused
            }
        });
```

### 3.2 putStream()

```
InputStream stream = ...;

// StorageReference fileRef = ...;
fileRef.putStream(stream)
        .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                // Uri: taskSnapshot.getDownloadUrl());
                // Name: taskSnapshot.getMetadata().getName());
                // Path: taskSnapshot.getMetadata().getPath());
                // Size: taskSnapshot.getMetadata().getSizeBytes());
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception exception) {
                // Handle unsuccessful uploads
            }
        })
        .addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onProgress(UploadTask.TaskSnapshot taskSnapshot) {
                // taskSnapshot.getBytesTransferred()
                // taskSnapshot.getTotalByteCount() = -1 (always)
            }
        })
        .addOnPausedListener(new OnPausedListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onPaused(UploadTask.TaskSnapshot taskSnapshot) {
                // Upload is paused
            }
        });
```

### 3.3 putFile()

```
Uri fileUri = ...;

// StorageReference fileRef = ...;
fileRef.putFile(fileUri)
        .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                // Uri: taskSnapshot.getDownloadUrl());
                // Name: taskSnapshot.getMetadata().getName());
                // Path: taskSnapshot.getMetadata().getPath());
                // Size: taskSnapshot.getMetadata().getSizeBytes());
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception exception) {
                // Handle unsuccessful uploads
            }
        })
        .addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onProgress(UploadTask.TaskSnapshot taskSnapshot) {
                // taskSnapshot.getBytesTransferred()
                // taskSnapshot.getTotalByteCount()
            }
        })
        .addOnPausedListener(new OnPausedListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onPaused(UploadTask.TaskSnapshot taskSnapshot) {
                // Upload is paused
            }
        });
```

### 3.4 Manage Uploads

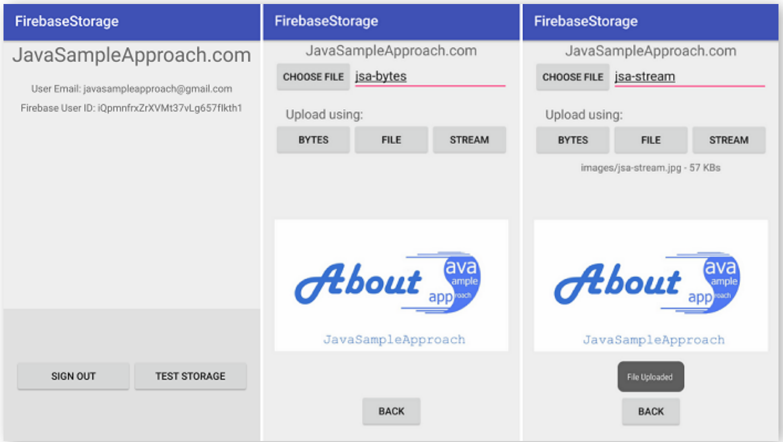
We can pause, resume, or cancel upload process:

```
Upload uploadTask = fileRef.putFile(file); // putBytes() or putStream()

uploadTask.pause();
uploadTask.resume();
uploadTask.cancel();
```

## II. Practice

### 1. Goal

We will build an Android App that can:
– create Account, sign in/sign out for Firebase Authentication.
– choose image from Gallery, then upload it to Firebase Cloud Storage using `putBytes()`, `putStream()` and `putFile()` methods.



User Email: javasampleapproach@gmail.com
Firebase User ID: iQpmnfrxZrXVMt37vLg657flkth1

jsa-bytes

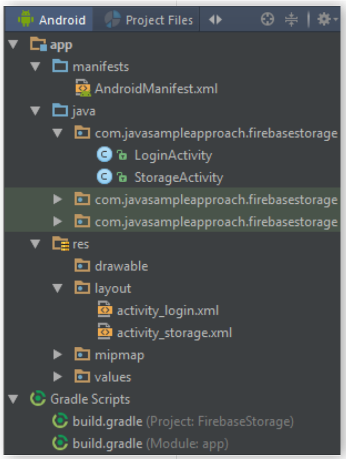
jsa-stream

images/jsa-stream.jpg - 57 KBs

### 2. Technology

– Gradle 2.3.3
– Android Studio 2.x
– Firebase Android SDK 11.x

### 3. Project Structure



```
Android   Project Files
app
    manifests
        AndroidManifest.xml
    java
        com.javasampleapproach.firebasestorage
            LoginActivity
            StorageActivity
        com.javasampleapproach.firebasestorage
        com.javasampleapproach.firebasestorage
    res
        drawable
        layout
            activity_login.xml
            activity_storage.xml
        mipmap
        values
Gradle Scripts
    build.gradle (Project: FirebaseStorage)
    build.gradle (Module: app)
```

LoginActivity is for Authentication, then user can change to StorageActivity to upload image to Firebase Cloud Storage.

**4. Step by step**

### 4.1 Create Android Project

– Generate new Android Project with package `com.javasampleapproach.firebasestorage`.
– build.gradle (project-level):

```
buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.3.3'
        classpath 'com.google.gms:google-services:3.1.0'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}
```

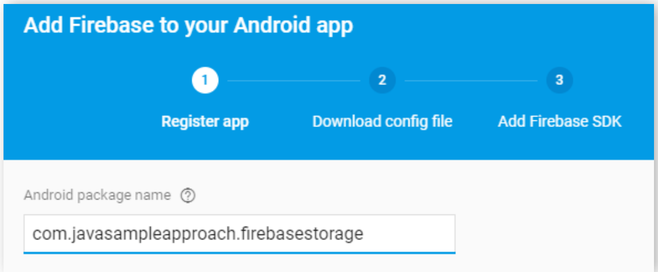
– build.gradle (App-level):

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.0"
    defaultConfig {
        applicationId "com.javasampleapproach.firebasestorage"
        minSdkVersion 15
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.google.firebase:firebase-auth:11.0.2'
    compile 'com.google.firebase:firebase-storage:11.0.2'
    testCompile 'junit:junit:4.12'
}

apply plugin: 'com.google.gms.google-services'
```

### 4.2 Create Firebase Project & Add Firebase Config file

– Follow this guide to generate google-services.json file and move it into your Android App root directory. You don't need to have SHA-1 Key in this example, just leave blank.

– Make sure that *package_name* in google-services.json has a correct value according to:
+ *applicationId* in build.gradle (App-level).
+ *package* in AndroidManifest.xml.
In this case, it is `com.javasampleapproach.firebasestorage`.



Add Firebase to your Android app

1 Register app

2 Download config file

3 Add Firebase SDK

Android package name: com.javasampleapproach.firebasestorage

### 4.3 Enable Firebase Auth

Go to Your Firebase Project Console -> Authentication -> SIGN-IN METHOD -> Enable Email/Password.

### 4.4 LoginActivity

To know how to implement Firebase Authentication App Client, please visit:
Firebase Authentication – How to Sign Up, Sign In, Sign Out, Verify Email | Android

In this tutorial, we don't explain way to authenticate an user again.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:weightSum="3">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="2"
        android:gravity="center_horizontal"
        android:orientation="vertical">

        <TextView
            android:id="@+id/title_text"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"
            android:gravity="center"
            android:text="JavaSampleApproach.com"
            android:textSize="28sp" />

        <TextView
            android:id="@+id/status"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:padding="4dp"
            android:text="Signed Out"
            android:textSize="14sp" />

        <TextView
            android:id="@+id/detail"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:padding="4dp"
            android:textSize="14sp"
            tools:text="Firebase User ID: 123456789abc" />

    </LinearLayout>

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#E0E0E0"
        android:gravity="center_vertical">

        <LinearLayout
            android:id="@+id/email_password_fields"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:paddingLeft="16dp"
            android:paddingRight="16dp">

            <EditText
                android:id="@+id/edt_email"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:hint="Email"
                android:inputType="textEmailAddress" />

            <EditText
                android:id="@+id/edt_password"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:hint="Password"
                android:inputType="textPassword" />
        </LinearLayout>

        <LinearLayout
            android:id="@+id/email_password_buttons"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_below="@+id/email_password_fields"
            android:orientation="horizontal"
            android:paddingLeft="16dp"
            android:paddingRight="16dp">

            <Button
                android:id="@+id/btn_email_sign_in"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_marginLeft="4dp"
                android:layout_marginRight="4dp"
                android:layout_weight="1"
                android:text="Sign In" />

            <Button
                android:id="@+id/btn_email_create_account"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_marginLeft="4dp"
                android:layout_marginRight="4dp"
                android:layout_weight="1"
                android:text="Create Account" />
        </LinearLayout>

        <LinearLayout
            android:id="@+id/layout_signed_in_control"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:orientation="horizontal"
            android:paddingLeft="16dp"
            android:paddingRight="16dp"
            android:visibility="gone">

            <Button
                android:id="@+id/btn_sign_out"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Sign Out" />

            <Button
                android:id="@+id/btn_test_message"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Test Storage" />

        </LinearLayout>

    </RelativeLayout>

</LinearLayout>
```

```
package com.javasampleapproach.firebasestorage;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class LoginActivity extends AppCompatActivity implements
        View.OnClickListener {

    private static final String TAG = "LoginActivity";

    private TextView txtStatus;
    private TextView txtDetail;
    private EditText edtEmail;
    private EditText edtPassword;

    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        txtStatus = (TextView) findViewById(R.id.status);
        txtDetail = (TextView) findViewById(R.id.detail);
        edtEmail = (EditText) findViewById(R.id.edt_email);
        edtPassword = (EditText) findViewById(R.id.edt_password);

        findViewById(R.id.btn_email_sign_in).setOnClickListener(this);
        findViewById(R.id.btn_email_create_account).setOnClickListener(this);
        findViewById(R.id.btn_sign_out).setOnClickListener(this);
        findViewById(R.id.btn_test_message).setOnClickListener(this);

        mAuth = FirebaseAuth.getInstance();
    }

    @Override
    protected void onStart() {
        super.onStart();

        FirebaseUser currentUser = mAuth.getCurrentUser();
        updateUI(currentUser);
    }

    @Override
    public void onClick(View view) {
        int i = view.getId();

        if (i == R.id.btn_email_create_account) {
            createAccount(edtEmail.getText().toString(), edtPassword.getText().toString());
        } else if (i == R.id.btn_email_sign_in) {
            signIn(edtEmail.getText().toString(), edtPassword.getText().toString());
        } else if (i == R.id.btn_sign_out) {
            signOut();
        } else if (i == R.id.btn_test_message) {
            testStorage();
        }
    }

    private void createAccount(String email, String password) {
        Log.e(TAG, "createAccount:" + email);
        if (!validateForm(email, password)) {
            return;
        }

        mAuth.createUserWithEmailAndPassword(email, password)
                .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            Log.e(TAG, "createAccount: Success!");

                            // update UI with the signed-in user's information
                            FirebaseUser user = mAuth.getCurrentUser();
                            updateUI(user);
                        } else {
                            Log.e(TAG, "createAccount: Fail!", task.getException());
                            Toast.makeText(LoginActivity.this, "Authentication failed!", Toast.LENGTH_SHORT).show();
                            updateUI(null);
                        }
                    }
                });
    }

    private void signIn(String email, String password) {
        Log.e(TAG, "signIn:" + email);
        if (!validateForm(email, password)) {
            return;
        }

        mAuth.signInWithEmailAndPassword(email, password)
                .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            Log.e(TAG, "signIn: Success!");

                            // update UI with the signed-in user's information
                            FirebaseUser user = mAuth.getCurrentUser();
                            updateUI(user);
                        } else {
                            Log.e(TAG, "signIn: Fail!", task.getException());
                            Toast.makeText(LoginActivity.this, "Authentication failed!", Toast.LENGTH_SHORT).show();
                            updateUI(null);
                        }

                        if (!task.isSuccessful()) {
                            txtStatus.setText("Authentication failed!");
                        }
                    }
                });
    }

    private void signOut() {
        mAuth.signOut();
        updateUI(null);
    }

    private boolean validateForm(String email, String password) {

        if (TextUtils.isEmpty(email)) {
            Toast.makeText(LoginActivity.this, "Enter email address!", Toast.LENGTH_SHORT).show();
            return false;
        }

        if (TextUtils.isEmpty(password)) {
            Toast.makeText(LoginActivity.this, "Enter password!", Toast.LENGTH_SHORT).show();
            return false;
        }

        if (password.length() < 6) {
            Toast.makeText(LoginActivity.this, "Password too short, enter minimum 6 characters!", Toast.LENGTH_SHORT).show();
            return false;
        }

        return true;
    }

    private void updateUI(FirebaseUser user) {
        if (user != null) {
            txtStatus.setText("User Email: " + user.getEmail());
            txtDetail.setText("Firebase User ID: " + user.getUid());

            findViewById(R.id.email_password_buttons).setVisibility(View.GONE);
            findViewById(R.id.email_password_fields).setVisibility(View.GONE);
            findViewById(R.id.layout_signed_in_control).setVisibility(View.VISIBLE);

        } else {
            txtStatus.setText("Signed Out");
            txtDetail.setText(null);

            findViewById(R.id.email_password_buttons).setVisibility(View.VISIBLE);
            findViewById(R.id.email_password_fields).setVisibility(View.VISIBLE);
            findViewById(R.id.layout_signed_in_control).setVisibility(View.GONE);
        }
    }

    private void testStorage() {
        startActivity(new Intent(this, StorageActivity.class));
    }
}
```

### 4.5 StorageActivity

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingLeft="16dp"
    android:paddingRight="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="JavaSampleApproach.com"
        android:textSize="20sp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="3">

        <Button
            android:id="@+id/btn_choose_file"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Choose File" />

        <EditText
            android:id="@+id/edt_file_name"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="File Name" />

    </LinearLayout>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="16dp"
        android:text="Upload using:"
        android:textSize="18sp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="3">

        <Button
            android:id="@+id/btn_upload_byte"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Bytes" />

        <Button
            android:id="@+id/btn_upload_file"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="File" />

        <Button
            android:id="@+id/btn_upload_stream"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Stream" />

    </LinearLayout>

    <TextView
        android:id="@+id/tv_file_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="5dp"
        android:text="File Name" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:weightSum="5">

        <ImageView
            android:id="@+id/img_file"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="4.7" />

        <Button
            android:id="@+id/btn_back"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:text="Back" />

    </LinearLayout>

</LinearLayout>
```

```
package com.javasampleapproach.firebasestorage;

import android.app.ProgressDialog;
```

```java
import android.content.ContentResolver;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.provider.MediaStore;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.webkit.MimeTypeMap;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.OnPausedListener;
import com.google.firebase.storage.OnProgressListener;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.io.ByteArrayOutputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;

public class StorageActivity extends AppCompatActivity implements View.OnClickListener {

    private static final String TAG = "StorageActivity";
    //track Choosing Image Intent
    private static final int CHOOSING_IMAGE_REQUEST = 1234;

    private TextView tvFileName;
    private ImageView imageView;
    private EditText edtFileName;

    private Uri fileUri;
    private Bitmap bitmap;
    private StorageReference imageReference;

    ProgressDialog progressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_storage);

        imageView = (ImageView) findViewById(R.id.img_file);
        edtFileName = (EditText) findViewById(R.id.edt_file_name);
        tvFileName = (TextView) findViewById(R.id.tv_file_name);
        tvFileName.setText("");

        imageReference = FirebaseStorage.getInstance().getReference().child("images");
        progressDialog = new ProgressDialog(this);

        findViewById(R.id.btn_choose_file).setOnClickListener(this);
        findViewById(R.id.btn_upload_byte).setOnClickListener(this);
        findViewById(R.id.btn_upload_file).setOnClickListener(this);
        findViewById(R.id.btn_upload_stream).setOnClickListener(this);
        findViewById(R.id.btn_back).setOnClickListener(this);
    }

    private void uploadBytes() {

        if (fileUri != null) {
            String fileName = edtFileName.getText().toString();

            if (!validateInputFileName(fileName)) {
                return;
            }

            progressDialog.setTitle("Uploading...");
            progressDialog.show();

            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            bitmap.compress(Bitmap.CompressFormat.JPEG, 50, baos);
            byte[] data = baos.toByteArray();

            StorageReference fileRef = imageReference.child(fileName + "." + getFileExtension(fileUri));
            fileRef.putBytes(data)
                    .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                            progressDialog.dismiss();

                            Log.e(TAG, "Uri: " + taskSnapshot.getDownloadUrl());
                            Log.e(TAG, "Name: " + taskSnapshot.getMetadata().getName());

                            tvFileName.setText(taskSnapshot.getMetadata().getPath() + " - "
                                    + taskSnapshot.getMetadata().getSizeBytes() / 1024 + " KBs");
                            Toast.makeText(StorageActivity.this, "File Uploaded ", Toast.LENGTH_LONG).show();
                        }
                    })
                    .addOnFailureListener(new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception exception) {
                            progressDialog.dismiss();

                            Toast.makeText(StorageActivity.this, exception.getMessage(), Toast.LENGTH_LONG).show();
                        }
                    })
                    .addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onProgress(UploadTask.TaskSnapshot taskSnapshot) {
                            // progress percentage
                            double progress = (100.0 * taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();

                            // percentage in progress dialog
                            progressDialog.setMessage("Uploaded " + ((int) progress) + "%...");
                        }
                    })
                    .addOnPausedListener(new OnPausedListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onPaused(UploadTask.TaskSnapshot taskSnapshot) {
                            System.out.println("Upload is paused!");
                        }
                    });
        } else {
            Toast.makeText(StorageActivity.this, "No File!", Toast.LENGTH_LONG).show();
        }
    }

    private void uploadFile() {
        if (fileUri != null) {
            String fileName = edtFileName.getText().toString();

            if (!validateInputFileName(fileName)) {
                return;
            }

            progressDialog.setTitle("Uploading...");
            progressDialog.show();

            StorageReference fileRef = imageReference.child(fileName + "." + getFileExtension(fileUri));
            fileRef.putFile(fileUri)
                    .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                            progressDialog.dismiss();

                            Log.e(TAG, "Uri: " + taskSnapshot.getDownloadUrl());
                            Log.e(TAG, "Name: " + taskSnapshot.getMetadata().getName());

                            tvFileName.setText(taskSnapshot.getMetadata().getPath() + " - "
                                    + taskSnapshot.getMetadata().getSizeBytes() / 1024 + " KBs");
                            Toast.makeText(StorageActivity.this, "File Uploaded ", Toast.LENGTH_LONG).show();
                        }
                    })
                    .addOnFailureListener(new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception exception) {
                            progressDialog.dismiss();

                            Toast.makeText(StorageActivity.this, exception.getMessage(), Toast.LENGTH_LONG).show();
                        }
                    })
                    .addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onProgress(UploadTask.TaskSnapshot taskSnapshot) {
                            // progress percentage
                            double progress = (100.0 * taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();

                            // percentage in progress dialog
                            progressDialog.setMessage("Uploaded " + ((int) progress) + "%...");
                        }
                    })
                    .addOnPausedListener(new OnPausedListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onPaused(UploadTask.TaskSnapshot taskSnapshot) {
                            System.out.println("Upload is paused!");
                        }
                    });
        } else {
            Toast.makeText(StorageActivity.this, "No File!", Toast.LENGTH_LONG).show();
        }
    }

    private void uploadStream() {
        if (fileUri != null) {
            String fileName = edtFileName.getText().toString();

            if (!validateInputFileName(fileName)) {
                return;
            }

            progressDialog.setTitle("Uploading...");
            progressDialog.show();

            try {
                InputStream stream = getContentResolver().openInputStream(fileUri);

                StorageReference fileRef = imageReference.child(fileName + "." + getFileExtension(fileUri));
                fileRef.putStream(stream)
                        .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                            @Override
                            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                                progressDialog.dismiss();
                                Log.e(TAG, "Uri: " + taskSnapshot.getDownloadUrl());
                                Log.e(TAG, "Name: " + taskSnapshot.getMetadata().getName());

                                tvFileName.setText(taskSnapshot.getMetadata().getPath() + " - "
                                        + taskSnapshot.getMetadata().getSizeBytes() / 1024 + " KBs");
                                Toast.makeText(StorageActivity.this, "File Uploaded ", Toast.LENGTH_LONG).show();
                            }
                        })
                        .addOnFailureListener(new OnFailureListener() {
                            @Override
                            public void onFailure(@NonNull Exception exception) {
                                progressDialog.dismiss();

                                Toast.makeText(StorageActivity.this, exception.getMessage(), Toast.LENGTH_LONG).show();
                            }
                        })
                        .addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {
                            @Override
                            public void onProgress(UploadTask.TaskSnapshot taskSnapshot) {
                                // because this is a stream so:
                                // taskSnapshot.getTotalByteCount() = -1 (always)
                                progressDialog.setMessage("Uploaded " + taskSnapshot.getBytesTransferred() + " Bytes...");
                            }
                        })
                        .addOnPausedListener(new OnPausedListener<UploadTask.TaskSnapshot>() {
                            @Override
                            public void onPaused(UploadTask.TaskSnapshot taskSnapshot) {
                                System.out.println("Upload is paused!");
                            }
                        });

            } catch (FileNotFoundException e) {
                e.printStackTrace();
            }
        } else {
            Toast.makeText(StorageActivity.this, "No File!", Toast.LENGTH_LONG).show();
        }
    }

    private void showChoosingFile() {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Select Image"), CHOOSING_IMAGE_REQUEST);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (bitmap != null) {
            bitmap.recycle();
        }

        if (requestCode == CHOOSING_IMAGE_REQUEST && resultCode == RESULT_OK && data != null && data.getData() != null) {
            fileUri = data.getData();
            try {
                bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), fileUri);
                imageView.setImageBitmap(bitmap);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    @Override
    public void onClick(View view) {
        int i = view.getId();

        if (i == R.id.btn_choose_file) {
            showChoosingFile();
        } else if (i == R.id.btn_upload_byte) {
            uploadBytes();
        } else if (i == R.id.btn_upload_file) {
            uploadFile();
        } else if (i == R.id.btn_upload_stream) {
            uploadStream();
        } else if (i == R.id.btn_back) {
            finish();
        }
    }

    private String getFileExtension(Uri uri) {
        ContentResolver contentResolver = getContentResolver();
        MimeTypeMap mime = MimeTypeMap.getSingleton();

        return mime.getExtensionFromMimeType(contentResolver.getType(uri));
    }

    private boolean validateInputFileName(String fileName) {

        if (TextUtils.isEmpty(fileName)) {
            Toast.makeText(StorageActivity.this, "Enter file name!", Toast.LENGTH_SHORT).show();
            return false;
        }

        return true;
    }
}
```

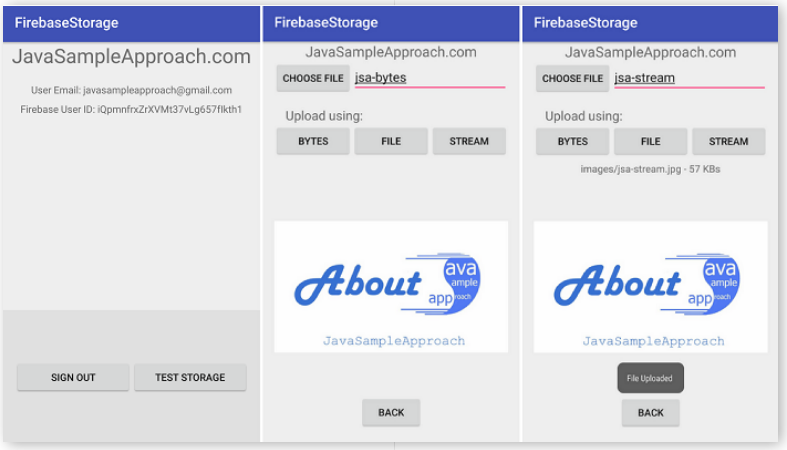
### 4.6 Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.javasampleapproach.firebasestorage">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".LoginActivity">
```

```
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".StorageActivity" />
    </application>

</manifest>
```

### 4.7 Run & Check result

– Use Android Studio, build and Run your Android App:



User Email: javasampleapproach@gmail.com

Firebase User ID: iQpmnfrxZrXVMt37vLg657flkth1

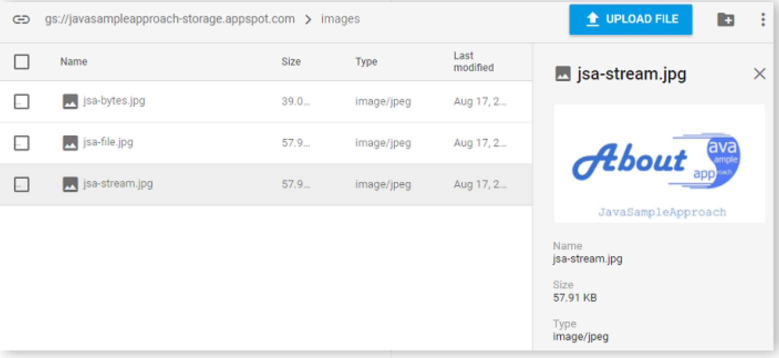
jsa-bytes

jsa-stream

images/jsa-stream.jpg - 57 KBs

File Uploaded

– Open Firebase Project Console -> Storage:



gs://javasampleapproach-storage.appspot.com > images

| ☐ | Name | Size | Type | Last modified |
|---|---|---|---|---|
| ☐ | jsa-bytes.jpg | 39.0... | image/jpeg | Aug 17, 2... |
| ☐ | jsa-file.jpg | 57.9... | image/jpeg | Aug 17, 2... |
| ☐ | jsa-stream.jpg | 57.9... | image/jpeg | Aug 17, 2... |

jsa-stream.jpg

Name
jsa-stream.jpg

Size
57.91 KB

Type
image/jpeg

## III. Source code

FirebaseStorage

By JavaSampleApproach | August 17, 2017.
Last updated on December 22, 2017.

### Related Posts

- Kotlin Firebase Storage – Get List of Files example – Image List with FirebaseRecyclerAdapter | Android
- Kotlin Firebase Storage – Download Files to Memory, Local File | Android
- Kotlin Firebase Storage – Upload Data from Memory, Local File, Stream | Android
- Kotlin Firebase Realtime Database – Display List of Data with FirebaseRecyclerAdapter | Android
- Kotlin Firebase Realtime Database – Get List of Data example | Android
- Kotlin Firebase Realtime Database – Read/Write Data example | Android
- Kotlin Firebase Authentication – Google Sign In | Android
- Kotlin Firebase Authentication – Send Reset Password Email / Forgot Password | Android
- Kotlin Firebase Authentication – How to Sign Up, Sign In, Sign Out, Verify Email | Android
- Kotlin Firebase Cloud Messaging – How to Send Upstream Messages | Android

### Post Tags

android | android development | cloud database | cloud storage | firebase | firebase storage | google cloud | upload file

## 4 thoughts on "Firebase Storage – Upload Data from Memory, Local File, Stream | Android"

Joy

September 18, 2017 at 6:00 pm

it helped a lot thanks
now i have to find something that doens't need Login or anything the tutorials i've found are for sdk 24.0.0 version and i don't know why don't matter what I do it doesn't works out

JavaSampleApproach

September 20, 2017 at 1:42 am

Hi Joy,

If you want to work with Firebase Storage without Authentication (Login), you can go to your Project in Firebase Console, chose Storage tab -> RULES, change the rules:

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

Scott

November 2, 2017 at 10:32 am

Which of the methods would be fastest? Have you tried to compare speed?

JavaSampleApproach

November 3, 2017 at 6:20 am

Hi Scott,

`putBytes()` would be the fastest method because it holds the entire contents of a file in memory at once.

Regards,
JSA.

JavaSampleApproach

Home | Privacy Policy | Contact Us | Our Team

## FOLLOW US

## ABOUT US

We are passionate engineers in software development by Java Technology & Spring Framework. We believe that creating little good thing with specific orientation everyday can make great influence on the world someday.