

1. firebase image upload download code full-----

@@ java files:

1. Upload---

```
package com.example.anabi.anisulfirebaseimageuploadretriveproject;
```

```
import com.google.firebase.database.Exclude;
```

```
public class Upload {
```

```
    private String imageName;
```

```
    private String imageUrl;
```

```
    private String key;
```

```
    @Exclude
```

```
    public String getKey() {
```

```
        return key;
```

```
    }
```

```
    @Exclude
```

```
    public void setKey(String key) {
```

```
        this.key = key;
```

```
    }
```

```
    public Upload(){
```

```
    }
```

```
    public Upload(String imageName, String imageUrl) {
```

```
        this.imageName = imageName;
```

```
        this.imageUrl = imageUrl;
```

```
    }
```

```
    public String getImageName() {
```

```
        return imageName;
```

```
    }
```

```
    public void setImageName(String imageName) {
```

```
        this.imageName = imageName;
```

```
    }
```

```
    public String getImageUrl() {
```

```
        return imageUrl;
```

```
    }
```

```
    public void setImageUrl(String imageUrl) {
```

```
        this.imageUrl = imageUrl;
```

```
    }
```

```
    }
```

@@@ MyAdapter---

```
package com.example.anabi.anisulfirebaseimageuploadretriveproject;

import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.ContextMenu;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.squareup.picasso.Picasso;

import java.util.List;

public class MyAdapter extends RecyclerView.Adapter<MyAdapter.MyViewHolder> {

    private Context context;
    private List<Upload> uploadList;
    private OnItemClickListener listener;

    public MyAdapter(Context context, List<Upload> uploadList) {
        this.context = context;
        this.uploadList = uploadList;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {

        LayoutInflater layoutInflater = LayoutInflater.from(context);
        View view = layoutInflater.inflate(R.layout.item_layout, viewGroup, false);

        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MyViewHolder myViewHolder, int i) {

        Upload upload = uploadList.get(i);

        myViewHolder.textView.setText(upload.getImageName());

        Picasso.with(context)
                .load(upload.getImageUrl())
                .placeholder(R.mipmap.ic_launcher_round)
                .fit()
                .centerCrop()
                .into(myViewHolder.imageView);

    }
}
```

```

@Override
public int getItemCount() {
return uploadList.size();
}

public class MyViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener,
View.OnCreateContextMenuListener,MenuItem.OnMenuItemClickListener {

TextView textView;
ImageView imageView;

public MyViewHolder(@NonNull View itemView) {
super(itemView);

textView = itemView.findViewById(R.id.cardTextViewId);
imageView = itemView.findViewById(R.id.cardImageViewId);
itemView.setOnClickListener(this);
itemView.setOnCreateContextMenuListener(this);

}

@Override
public void onClick(View v) {

if(listener!=null){

int position = getAdapterPosition();

if(position!=RecyclerView.NO_POSITION){

listener.onClick(position);
}
}

}

@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {

menu.setHeaderTitle("Choose an action");
MenuItem doAnyTask = menu.add(Menu.NONE,1,1,"do any task");
MenuItem delete = menu.add(Menu.NONE,2,2,"delete");

doAnyTask.setOnMenuItemClickListener(this);
delete.setOnMenuItemClickListener(this);

}

@Override
public boolean onOptionsItemSelected(MenuItem item) {

if(listener!=null){

int position = getAdapterPosition();

```

```
if(position!=RecyclerView.NO_POSITION){

switch (item.getItemId()){

case 1:
listener.onDoAnyTask(position);
return true;

case 2:
listener.onDelete(position);
return true;

}

}

return false;
}

}
```

```
//listener
```

```
public interface OnItemClickListener{

void onItemClick(int position);

void onDoAnyTask(int position);

void onDelete(int position);

}

public void setOnItemClickListener(OnItemClickListener listener){

this.listener = listener;

}

}
```

```
@@@ MainActivity----
```

```
package com.example.anabi.anisulfirebaseimageuploadretriveproject;

import android.content.ContentResolver;
import android.content.Intent;
import android.media.Image;
import android.net.Uri;
```

```

import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.webkit.MimeTypeMap;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.StorageTask;
import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Picasso;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    private Button chooseButton,saveButton,displayButton;
    private ImageView imageView;
    private EditText imageNameEditText;
    private ProgressBar progressBar;
    private Uri imageUri;
    StorageTask uploadTask;

    DatabaseReference databaseReference;
    StorageReference storageReference;

    private static final int IMAGE_REQUEST = 1;

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

        databaseReference = FirebaseDatabase.getInstance().getReference("Upload");
        storageReference = FirebaseStorage.getInstance().getReference("Upload");

        chooseButton = findViewById(R.id.chooseImageButtonId);
        displayButton = findViewById(R.id.displayImageButtonId);
        saveButton = findViewById(R.id.saveImageButtonId);
        progressBar =findViewById(R.id.progressBarId);

        imageView = findViewById(R.id.imageViewId);
        imageNameEditText = findViewById(R.id.imageNameEditTextId);

        saveButton.setOnClickListener(this);
        displayButton.setOnClickListener(this);
        chooseButton.setOnClickListener(this);

```

```
}

@Override
public void onClick(View v) {

    switch (v.getId()){

        case R.id.chooseImageButtonId:

            openFileChooser();

            break;


        case R.id.saveImageButtonId:

            if(uploadTask!=null && uploadTask.isInProgress()){

                Toast.makeText(getApplicationContext(),"Uploading is in Progress",Toast.LENGTH_SHORT).show();

            }
            else{

                saveData();

            }

            break;


        case R.id.displayImageButtonId:

            Intent intent = new Intent(getApplicationContext(),ImageActivity.class);
            startActivity(intent);
            break;

    }

}

void openFileChooser(){

    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(intent,IMAGE_REQUEST);

}
```

```
}
```

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

    if(requestCode == IMAGE_REQUEST && resultCode == RESULT_OK && data!=null && data.getData()!=null){

        imageUri = data.getData();
        Picasso.with(this).load(imageUri).into(imageView);

    }
}
```

```
// getFile Extension
```

```
public String getFileExtension(Uri imageUri){

    ContentResolver contentResolver = getContentResolver();
    MimeTypeMap mimeTypeMap = MimeTypeMap.getSingleton();
    return mimeTypeMap.getExtensionFromMimeType(contentResolver.getType(imageUri));

}
```

```
//Save Data
```

```
private void saveData(){

    final String imageName = imageNameEditText.getText().toString().trim();

    if(imageName.isEmpty()){

        imageNameEditText.setError("Enter the image name");
        imageNameEditText.requestFocus();
        return;
    }

}
```

```
StorageReference ref = storageReference.child(System.currentTimeMillis()+"."+getFileExtension(imageUri));
```

```
ref.putFile(imageUri)
    .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
```

```
// Get a URL to the uploaded content
```

```
Toast.makeText(getApplicationContext(),"Image is stored successfully",Toast.LENGTH_SHORT).show();
```

```
Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();
```

```
while(!uriTask.isSuccessful());
```

```
Uri downloadUrl = uriTask.getResult();
```

```
Upload upload = new Upload(imageName, downloadUrl.toString());
```

```
String uploadId = databaseReference.push().getKey();
```

```
databaseReference.child(uploadId).setValue(upload);
```

```
}
```

```
}}
```

```
.addOnFailureListener(new OnFailureListener() {
```

```
@Override
```

```
public void onFailure(@NonNull Exception exception) {
```

```
// Handle unsuccessful uploads
```

```
Toast.makeText(getApplicationContext(),"Image is not stored! failed",Toast.LENGTH_SHORT).show();
```

```
// ...
```

```
}
```

```
});
```

```
}
```

```
}
```

```
@@@ ImageActivity-----
```

```
package com.example.anabi.anisulfirebaseimageuploadretriveproject;
```

```
import android.support.annotation.NonNull;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
import android.support.v7.widget.LinearLayoutManager;
```

```
import android.support.v7.widget.RecyclerView;
```

```
import android.view.View;
```

```
import android.widget.ProgressBar;
```

```
import android.widget.Toast;
```

```
import com.google.android.gms.tasks.OnSuccessListener;
```

```
import com.google.firebase.database.DataSnapshot;
```

```
import com.google.firebase.database.DatabaseError;
```

```
import com.google.firebase.database.DatabaseReference;
```

```
import com.google.firebase.database.FirebaseDatabase;
```

```
import com.google.firebase.database.ValueEventListener;
```

```
import com.google.firebase.storage.FirebaseStorage;
```

```
import com.google.firebase.storage.StorageReference;
```

```
import java.util.ArrayList;
```



```

import java.util.List;

public class ImageActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private MyAdapter myAdapter;
    private List<Upload> uploadList;
    DatabaseReference databaseReference;
    private ProgressBar newProgressbar;
    private FirebaseStorage firebaseStorage;

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

        recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        uploadList = new ArrayList<>();

        firebaseStorage = FirebaseStorage.getInstance();

        newProgressbar = findViewById(R.id.progressBarNew);

        databaseReference = FirebaseDatabase.getInstance().getReference("Upload");

        databaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

                uploadList.clear();

                for(DataSnapshot dataSnapshot1: dataSnapshot.getChildren()){

                    Upload upload = dataSnapshot1.getValue(Upload.class);
                    upload.setKey(dataSnapshot1.getKey());

                    uploadList.add(upload);
                }

                myAdapter = new MyAdapter(ImageActivity.this,uploadList);
                recyclerView.setAdapter(myAdapter);

                myAdapter.setOnItemClickListener(new MyAdapter.OnItemClickListener() {
                    @Override
                    public void onItemClick(int position) {

                        String text = uploadList.get(position).getImageName();
                        Toast.makeText(getApplicationContext(),text+" is selected "+position,Toast.LENGTH_SHORT).show();
                    }

                })

            @Override

```

```

public void onDoAnyTask(int position) {

    Toast.makeText(getApplicationContext()," On Do Any Task Selected ",Toast.LENGTH_SHORT).show();

}

@Override
public void onDelete(int position) {

    Upload selectedItem = uploadList.get(position);
    final String key = selectedItem.getKey();

    StorageReference storageReference = firebaseStorage.getReferenceFromUrl(selectedItem.getImageUrl());
    storageReference.delete().addOnSuccessListener(new OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {

            databaseReference.child(key).removeValue();
            Toast.makeText(getApplicationContext(),"Item is deleted",Toast.LENGTH_SHORT).show();

        }
    });

}

}

newProgressbar.setVisibility(View.INVISIBLE);

}

@Override
public void onCancelled(@NonNull DatabaseError databaseError) {

    Toast.makeText(getApplicationContext(),"Error: "+databaseError.getMessage(),Toast.LENGTH_LONG).show();

    newProgressbar.setVisibility(View.VISIBLE);

}

}

}

```

activity_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"

```

```
android:weightSum="3"
tools:context=".MainActivity">
```

```
<Button
android:backgroundTint="@color/colorAccent"
android:id="@+id/chooseImageButtonId"
style="@style/Base.TextAppearance.AppCompat.Widget.Button.Colored"
android:layout_marginTop="10dp"
android:layout_weight="0.25"
android:text="Choose Image"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
```

```
<EditText
android:id="@+id/imageNameEditTextId"
android:textAppearance="@style/Base.TextAppearance.AppCompat.Large"
android:layout_weight="0.25"
android:layout_marginTop="10dp"
android:hint="Enter a name for Image"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
```

```
<ImageView
android:id="@+id/imageViewId"
android:layout_marginTop="10dp"
android:layout_weight="2"
android:layout_width="match_parent"
android:layout_height="300dp" />
```

```
<ProgressBar
android:id="@+id/progressBarId"
android:layout_weight="0.16"
android:layout_gravity="center"
android:visibility="gone"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
```

```
<Button
android:backgroundTint="@color/colorAccent"
android:id="@+id/saveImageButtonId"
android:text="Save Image"
style="@style/Base.TextAppearance.AppCompat.Widget.Button.Colored"
android:layout_weight="0.17"
android:layout_marginTop="5dp"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
```

```
<Button
android:backgroundTint="@color/colorAccent"
android:id="@+id/displayImageButtonId"
android:text="Display Image"
style="@style/Base.TextAppearance.AppCompat.Widget.Button.Colored"
android:layout_weight="0.17"
android:layout_marginTop="5dp"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
```

</LinearLayout>

item_layout.xml-----

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:foreground="?android:attr/selectableItemBackground"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardElevation="6dp"
    app:cardCornerRadius="5dp"
    android:elevation="10dp"
    android:padding="10dp"

    android:layout_margin="5dp">
```

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

```
<ImageView
    android:id="@+id/cardImageViewId"
    android:scaleType="fitXY"
    android:layout_marginTop="10dp"
    android:layout_width="match_parent"
    android:layout_height="250dp" />
```

```
<TextView
    android:id="@+id/cardTextViewId"
    android:textColor="#464646"
    android:textAlignment="center"
    android:text="Image name"
    android:layout_marginTop="5dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="@style/Base.TextAppearance.AppCompat.Large"/>
```

</LinearLayout>

</android.support.v7.widget.CardView>

activity_image.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#efefef"
    tools:context=".ImageActivity">
```

```
<android.support.v7.widget.RecyclerView
android:id="@+id/recyclerView"
android:layout_width="match_parent"
android:layout_height="match_parent">

</android.support.v7.widget.RecyclerView>
```

```
<ProgressBar
android:id="@+id/progressbarNew"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />

</LinearLayout>
```