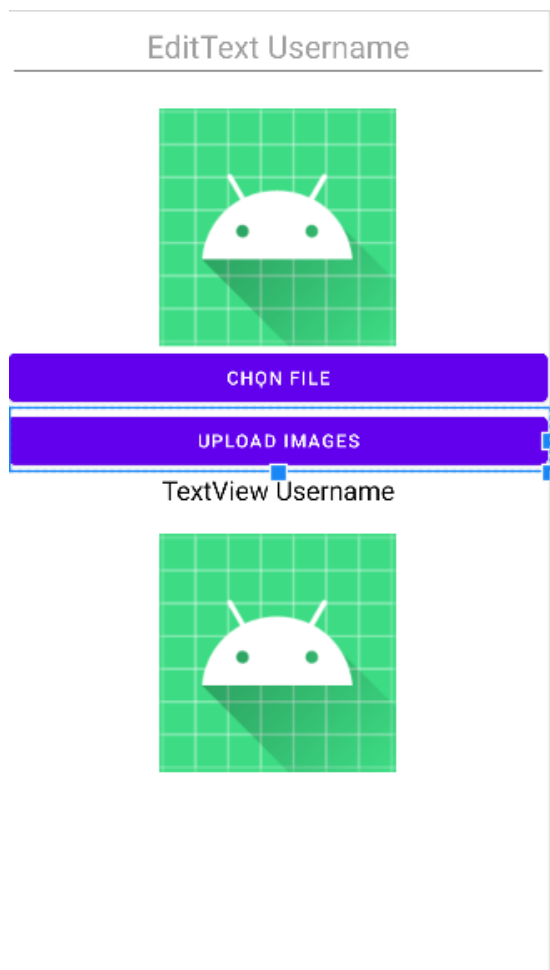




Upload File lên API trong Android



- Bước 1: Thiết kế giao diện
- Bước 2: Thêm thư viện
- Bước 3: Gọi API
- Bước 4: Viết code



- 01 EditText: editUserName
- 02 ImageView: imgChoose, imgMultipart
- 02 Button: btnChoose, btnUpload
- 01 TextView: tvUsername

```

□ // Network & Retrofit
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
//Gson
implementation 'com.google.code.gson:gson:2.10.1'
//load ảnh với Glide
implementation 'com.github.bumptech.glide:glide:4.14.2'
annotationProcessor 'com.github.bumptech.glide:compiler:4.14.2'

```

```

<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_MEDIA_IMAGES" />
<uses-permission android:name="android.permission.READ_MEDIA_AUDIO" />
<uses-permission android:name="android.permission.READ_MEDIA_VIDEO" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"
    tools:ignore="ScopedStorage" />

```

```

android:usesCleartextTraffic="true"

```

□ Tạo các hằng số

```
public class Const {
    public static final String MY_USERNAME = "username";
    public static final String MY_IMAGES = "avatar";
}
```

□ Tải class lấy đường dẫn chính xác của file:

<http://app.iotstar.vn/appfoods/RealPathUtil.java>

□ Tạo Model

```
public class ImageUpload {
    private int id;
    private String username;
    private String avatar;
```

□ Tạo Retrofit Client

```
public interface ServiceAPI {
    public static final String BASE_URL = "http://app.iotstar.vn/appfoods/";
    Gson gson = new GsonBuilder().setDateFormat("yyyy MM dd HH:mm:ss").create();

    ServiceAPI servieapi = new Retrofit.Builder()
        .baseUrl(BASE_URL) Retrofit.Builder
        .addConverterFactory(GsonConverterFactory.create(gson))
        .build() Retrofit
        .create(ServiceAPI.class);

    @Multipart
    @POST("upload.php")
    Call<List<ImageUpload>> upload(@Part(Const.MY_USERNAME)RequestBody username,
        @Part MultipartBody.Part avatar);

    @Multipart
    @POST("upload1.php")
    Call<Message> upload1(@Part(Const.MY_USERNAME)RequestBody username,
        @Part MultipartBody.Part avatar);
}
```

- Tạo hàm cấp quyền truy cập bộ nhớ cho file
- Khai báo biến toàn cục

```

public static String[] storage_permissions = {
    android.Manifest.permission.WRITE_EXTERNAL_STORAGE,
    android.Manifest.permission.READ_EXTERNAL_STORAGE
};

@RequiresApi(api = Build.VERSION_CODES.TIRAMISU)
public static String[] storage_permissions_33 = {
    android.Manifest.permission.READ_MEDIA_IMAGES,
    android.Manifest.permission.READ_MEDIA_AUDIO,
    android.Manifest.permission.READ_MEDIA_VIDEO
};

```

- Tạo hàm cấp quyền truy cập bộ nhớ cho file
- Viết hàm CheckPermission()

```
public static String[] permissions() {
    String[] p;
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
        p = storage_permissions_33;
    } else {
        p = storage_permissions;
    }
    return p;
}

private void CheckPermission() {
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M) {
        openGallery();
        return;
    }
    if (checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) == PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        //ActivityCompat.requestPermissions(UploadImageActivity.this, permissions(), MY_REQUEST_CODE);
        //hoặc
        requestPermissions(permissions(), MY_REQUEST_CODE);
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == MY_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            openGallery();
        }
    }
}
```


□ Viết hàm mở ảnh trên thiết bị

//Hàm xử lý lấy ảnh

```
private void openGallery() {  
    Intent intent = new Intent();  
    intent.setType("image/*");  
    intent.setAction(Intent.ACTION_GET_CONTENT);  
    startActivityForResult(Intent.createChooser(intent, title: "Select Picture"));  
}
```

- Viết hàm xử lý kết quả trả về từ hàm openGallery(), hàm này viết trên hàm onCreate()

```
private ActivityResultLauncher<Intent> mActivityResultLauncher = registerForActivityResult(
    new ActivityResultContracts.StartActivityForResult(),
    new ActivityResultCallback<ActivityResult>() {
        @Override
        public void onActivityResult(ActivityResult result) {
            Log.e(TAG, msg: "onActivityResult");
            if(result.getResultCode()== Activity.RESULT_OK){
                //request code
                Intent data = result.getData();
                if(data==null){
                    return;
                }
                Uri uri = data.getData();
                mUri = uri;
                try {
                    Bitmap bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(),uri);
                    imageViewChoose.setImageBitmap(bitmap);
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
);
```

- Viết khai báo biến toàn cục và viết hàm ánh xạ trong class MainActivity

```
public class MainActivity extends AppCompatActivity {
    Button btnChoose, btnUpload;
    ImageView imageViewChoose, imageViewUpload;
    EditText editTextUserName;
    TextView textViewUsername;
    private Uri mUri;
    private ProgressDialog mProgressDialog;
    public static final int MY_REQUEST_CODE=100;
    public static final String TAG = MainActivity.class.getName();
```

```
private void AnhXa() {
    btnChoose = findViewById(R.id.btnChoose);
    btnUpload = findViewById(R.id.btnUpload);
    imageViewUpload = findViewById(R.id.imgMultipart);
    editTextUserName = findViewById(R.id.editUserName);
    textViewUsername = findViewById(R.id.tvUsername);
    imageViewChoose = findViewById(R.id.imgChoose);
}
```

□ Gọi các hàm và bắt sự kiện trong onCreate()

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //gọi hàm ánh xạ
    AnhXa();
    //khởi tạo Progressbar
    mProgressDialog = new ProgressDialog( context: MainActivity.this);
    mProgressDialog.setMessage("Please wait upload....");
    //bắt sự kiện nút chọn ảnh
    btnChoose.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            CheckPermission();//checks quyền
            //chooseImage();
        }
    });
    //bắt sự kiện upload ảnh
    btnUpload.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(mUri != null){
                UploadImage1();
            }
        }
    });
}
```

□ Viết hàm Upload file lên Server và nhận phản hồi

```
public void UploadImage1(){
    mProgressDialog.show();
    //khai báo biến và setText nếu có
    String username = editTextUserName.getText().toString().trim();
    RequestBody requestUsername = RequestBody.create(MediaType.parse("multipart/form-data"), username);
    // create RequestBody instance from file
    String IMAGE_PATH = RealPathUtil.getRealPath(context: this, mUri);
    Log.e(tag: "ffff", IMAGE_PATH);
    File file = new File(IMAGE_PATH);
    RequestBody requestFile = RequestBody.create(MediaType.parse("multipart/form-data"), file);
    // MultipartBody.Part is used to send also the actual file name
    MultipartBody.Part partbodyavatar =
        MultipartBody.Part.createFormData(Const.MY_IMAGES, file.getName(), requestFile);
    //gọi Retrofit
    ServiceAPI.servieapi.upload(requestUsername, partbodyavatar).enqueue(new Callback<List<ImageUpload>>() {
        @Override
        public void onResponse(Call<List<ImageUpload>> call, Response<List<ImageUpload>> response) {
            mProgressDialog.dismiss();
            List<ImageUpload> imageUpload = response.body();
            if (imageUpload.size() > 0) {
                for (int i = 0; i < imageUpload.size(); i++) {
                    textViewUsername.setText(imageUpload.get(i).getUsername());
                    Glide.with(activity: MainActivity.this)
                        .load(imageUpload.get(i).getAvatar())
                        .into(imageViewUpload);
                    Toast.makeText(context: MainActivity.this, text: "Thành công", Toast.LENGTH_LONG).show();
                }
            } else {
                Toast.makeText(context: MainActivity.this, text: "Thất bại", Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```

- Viết hàm Upload file lên Server và nhận phản hồi

```

    }
    @Override
    public void onFailure(Call<List<ImageUpload>> call, Throwable t) {
        mProgressDialog.dismiss();
        Log.e(tag: "TAG", t.toString());
        Toast.makeText(context: MainActivity.this, text: "Gọi API thất bại", Toast.LENGTH_LONG).show();
    }
});
}

```

- Nguyễn Hữu Trung
- 0908617108
- trungnh@hcmute.edu.vn
- utex.hcmute.edu.vn