20\_07\_21\_학습내용

RTSP Player app개발

1. 로그인 기능 구현 #### 주의사항 : FilterExpression을 사용할 경우
2. Get 요청 후 수신한 JSON 파싱
3. 버튼 동적 할당

1. 로그인 기능

저번에 개발하였던 API Gateway + Lambda+ Dynamodb를 활용

우선 Login 부분에서

HttpUrlConnection을 만든 후

Button btn\_send = (Button) findViewById(R.id.*btn\_login*);

btn\_send.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

// Thread로 웹서버에 접속

new Thread() {

public void run() {

String resultHtml = getResultHtml();

}

}.start();

}

});

private String getResultHtml(){

String resultHtml = "";

URL url =null;

HttpURLConnection http = null;

InputStreamReader isr = null;

BufferedReader br = null;

EditText EID = (EditText)findViewById(R.id.*text\_id*);

EditText EPasswd = (EditText)findViewById(R.id.*text\_passwd*);

String ID = EID.getText().toString();

String Passwd = EPasswd.getText().toString();

String RequestUrl;

System.*out*.println("id is "+ID);

System.*out*.println("password is "+Passwd);

if(ID != "" && Passwd != "")

RequestUrl = "서버 주소 넣으세요"+"?id="+ID+"&passwd="+Passwd;

else

RequestUrl = "https://e5sggq8tjg.execute-api.ap-northeast-2.amazonaws.com/TestDeploy/Test\_Get";

try{

url = new URL(RequestUrl);

http = (HttpURLConnection) url.openConnection();

http.setConnectTimeout(3\*1000);

http.setReadTimeout(3\*1000);

isr = new InputStreamReader(http.getInputStream());

br = new BufferedReader(isr);

String str = null;

while ((str = br.readLine()) != null) {

login\_result = str;

// resultHtml += str + "\n";

}

if(login\_result.compareTo("\""+"Fail"+"\"") == 0 )

{

Toast.*makeText*(this,"아이디 혹은 비밀번호가 틀렸습니다.", Toast.*LENGTH\_LONG*).show();

}

else

{

/\*

로그인 성공시 수행할 기능들이다.

\*/

String[] dronelist = JSONParsing(login\_result); //받아온 응답 JSON을 파싱한다.

Intent intent = new Intent(

getApplicationContext(), // 현재 화면의 제어권자

MainActivity.class); // 다음 넘어갈 클래스 지정

intent.putExtra("DroneList",dronelist);

startActivity(intent); // 다음 화면으로 넘어간다

}

}catch(Exception e){

}finally{

if(http != null){

try{http.disconnect();}catch(Exception e){}

}

if(isr != null){

try{isr.close();}catch(Exception e){}

}

if(br != null){

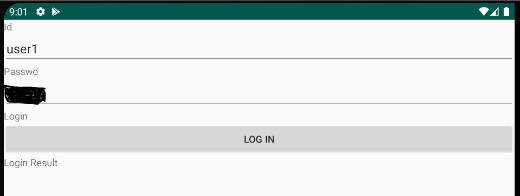
try{br.close();}catch(Exception e){}

}

}

return resultHtml;

}



로그인 버튼을 클릭하여 Get 요청

~/~? /Test\_Get?id=user1&passwd=\*\*\*\*은 매핑 템플릿

{

"TableName": "TestTable",

"FilterExpression":"ID = :x",

"FilterExpression":"Passwd = :y",

"ExpressionAttributeValues": {

":x": "$input.params().querystring.id" },

"ExpressionAttributeValues": {

":y": "$input.params().querystring.passwd" }

}

을 통하여 아래와 같은 JSON을 받아옴.

{

"Items": [

{

"Passwd": "user1",

"ID": "\*\*\*\*",

"DroneList": [

"drone1",

"drone2"

],

"Name": "김민석1"

}

],

"Count": 1,

"ScannedCount": 2

}

#### 주의사항 : 보고서를 작성 하던 중 발견하여 아직 수정은 못했지만.

scan을 위한 js를 위의 매핑 템플릿 처럼 작성하게 될 경우

첫 번째 FilterExpression이 씹혀서 ID는 달라도 Passwd만 같으면 현재

로그인이 되는 것을 발견하였다. 이는 내일 고칠 예정이다.

1. JSON 파싱

위의 내용에서 내가 쓰고 싶은 것은 DroneList의 값들이다.

따라서 이것들을 JSON 파싱 후 DroneList에 해당되는  
[“drone1”,”drone2”]를 알맞게 String 파싱해주면 간단하게 해결된다.

해당 코드는 아래와 같다.

private String[] JSONParsing(String result)

{

String[] dronelist = null;

JSONObject jsonobject = null;

try {

jsonobject = new JSONObject(result);

System.*out*.println("test1 "+jsonobject.toString());

JSONArray ItemArray = jsonobject.getJSONArray("Items");

System.*out*.println("test2 "+ItemArray.getJSONObject(0).toString());

JSONObject DroneObject = ItemArray.getJSONObject(0);

String temp\_result = DroneObject.getString("DroneList");

System.*out*.println("test3" + temp\_result);

//JSON Parsing 끝 --> 결과는 ["drone1","drone2"] 꼴임.

//따라서 추가적으로 StringParsing필요...

String[] temp\_dronelist;

temp\_dronelist = temp\_result.split("\"");

dronelist = new String[temp\_dronelist.length/2];

int list\_idx = 0;

System.*out*.println("length is "+temp\_dronelist.length);

for(int i = 1; i < temp\_dronelist.length-1; i=i+2)

{

dronelist[list\_idx] = temp\_dronelist[i];

System.*out*.println("보낼 list "+dronelist[list\_idx]);

list\_idx++;

}

} catch (JSONException e) {

e.printStackTrace();

}

return dronelist;

}

1. 버튼 동적 할당

위 과정을 통해서 받아온 “Drone x”의 수만큼 버튼을 동적으로 생성하려고 한다.

또한, 젤 바깥의 Layout이 아닌, 어떤 layout의 자식인 특정 layout에 버튼을 생성하려고 한다.

즉,

<FrameLayout

android:id="@+id/capture"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

~~~ 다른 위젯 혹은 레이아웃들 ~~~

<LinearLayout

android:id="@+id/ListSpace"

android:layout\_width="80dp"

android:layout\_height="688dp"

android:layout\_marginTop="52dp"

android:background="#ED4646"

android:visibility="gone"

android:orientation="vertical">

</LinearLayout>

</FrameLayout>

</LinearLayout>

제일 바깥의 LinearLayout 의 자식은 FrameLayout 이고,

이 FrameLayout 의 자식인 또다른 LinearLayout에다가 버튼을 생성하려고 한다.

또한, 처음 상태를 안보이게 해주고( android:visibility="gone"), 메인 액티비티에서 특정 버튼 클릭시 위 레이아웃을 보이게 하려고 한다.

MainActivity 중

DroneList = getIntent().getExtras().getStringArray("DroneList"); //Login에서 받아온 리스트

setContentView(R.layout.*activity\_main*);

for(int i = 0 ; i <DroneList.length; i++)

System.*out*.println("receive list is"+DroneList[i]);

layout = (LinearLayout)findViewById(R.id.*ListSpace*);

for(int i = 0 ; i < DroneList.length; i++) //리스트 항목 수 만큼 버튼 생성

{

Button B = new Button(this);

B.setId(i);

B.setText("Drone "+(i+1));

// String sub\_url = DroneList[i];

B.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

//각 버튼 클릭시 수행될 다른 동작들

}

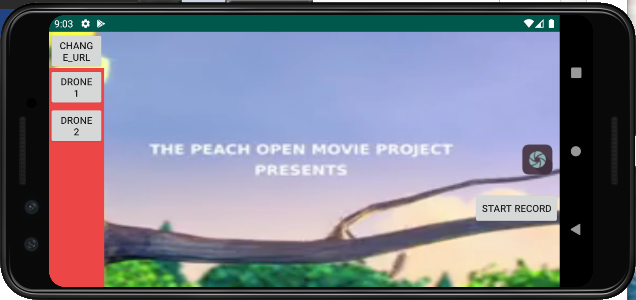
});

layout.addView(B);

}



좌측 상단의 CHANGE\_URL을 클릭시



다음과 같이 숨겨졌던 Layout과 그 안의 버튼들이 나온다.

그 후 다음 버튼들을 누르면 해당 영상이 재생된다.

(지금은 재생할 영상이 없어서 동일한 영상을 다시 처음부터 재생하게 해 놓았음)