

Museum Tour

```
public class StoreAppDetails extends Activity implements View.OnClickListener{

    LinearLayout matchStatsLayout;

    RelativeLayout backButton;

    String title, imageUrl, videoUrl;

    ImageView trophyPhoto;

    TextView trophyTitle, titleTextView;

    LinearLayout playVideoLayout;

    Typeface customFont;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        //Remove title bar

        this.requestWindowFeature(Window.FEATURE_NO_TITLE);

        //Remove notification bar

        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);

        setContentView(R.layout.activity_store_app_details);
```

```

customFont = Typeface.createFromAsset(getAssets(), "fonts/avenir.otf");

titleTextView = (TextView) findViewById(R.id.titleTextView);

Intent intent = getIntent();

title = intent.getStringExtra("Title");

ImageUrl = intent.getStringExtra("ImageUrl");

VideoUrl = intent.getStringExtra("VideoUrl");

Log.d("URL", ImageUrl+"ffff");

titleTextView.setText(title);

trohpyPhoto = (ImageView) findViewById(R.id.imageView3);

trophyTitle = (TextView) findViewById(R.id.textView2);


Log.d("Lamhe", ImageUrl);


Picasso.with(getApplicationContext())

    .load(ImageUrl)

    .into(trohpyPhoto);

trophyTitle.setText(title);


playVideoLayout = (LinearLayout) findViewById(R.id.playVideoLayout);

playVideoLayout.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v)

    {

        Intent goNext = new Intent(StoreAppDetails.this, VideoPlayerActivity.class);

```

```
        goNext.putExtra("Title", title);

        goNext.putExtra("ImageUrl", imageUrl);

        goNext.putExtra("VideoUrl", videoUrl);

        startActivity(goNext);
    }
});
```

```
backButton= (RelativeLayout) findViewById(R.id.backButton);

backButton.setOnClickListener(this);
```

```
matchStatsLayout= (LinearLayout) findViewById(R.id.matchStatsLayout);

matchStatsLayout.setOnClickListener(this);
}
```

```
@Override
```

```
public void onClick(View v) {
```

```
    switch (v.getId())
```

```
    {
```

```
        case R.id.backButton:
```

```
            Intent goBack=new Intent(getApplicationContext(),RecognitionFinderActivity.class);
```

```
            goBack.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TOP);
```

```
            startActivity(goBack);
```

```
            this.finish();
```

```
            break;
```

```
case R.id.matchStatsLayout:

    Intent openmatchStatsIntent=new Intent(getApplicationContext(),MatchStatsDetails.class);

    openmatchStatsIntent.putExtra("Title", title);

    openmatchStatsIntent.putExtra("ImageUrl", imageUrl);

    openmatchStatsIntent.putExtra("VideoUrl", videoUrl);

    startActivity(openmatchStatsIntent);

    break;

}
```

```
@Override
```

```
public void onBackPressed() {

    super.onBackPressed();

    Intent goBack=new Intent(getApplicationContext(),MainActivity.class);

    goBack.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TOP);

    startActivity(goBack);

    this.finish();

}
```

```
}
```

Museum Tour App

```
public class StoreAppActivity extends Activity implements View.OnClickListener {
```

```
    Button storeAppDetailsButton;
```

```
    RelativeLayout backButton;
```

```
    String uid, title, imageUrl, videoUrl;
```

```
    ProgressDialog progressDialog;
```

```
    ImageView trophyPhoto;
```

```
    Typeface customFont;
```

```
    TextView titleTextView;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        //Remove title bar
```

```
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
```

```
        //Remove notification bar
```

```
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

```
        setContentView(R.layout.activity_store_app);
```

```
        customFont = Typeface.createFromAsset(getAssets(), "fonts/avenir.otf");
```

```
Intent intent = getIntent();

uid = intent.getStringExtra("UID");

// Toast.makeText(this, "UID : "+uid, Toast.LENGTH_SHORT).show();

titleTextView = (TextView) findViewById(R.id.titleTextView);


        storeAppDetailsButton= (Button) findViewById(R.id.storeAppDetailsButton);

storeAppDetailsButton.setOnClickListener(this);

storeAppDetailsButton.setTypeface(customFont);


trophyPhoto = (ImageView) findViewById(R.id.storeAppImageView);


backButton= (RelativeLayout) findViewById(R.id.backButton);

backButton.setOnClickListener(this);

progressDialog = new ProgressDialog(StoreAppActivity.this);

progressDialog.show();

new SentUIdTask().execute("http://192.168.0.7/image-recognition/getData.php");
}
```

@Override

```
public void onBackPressed() {

    super.onBackPressed();


    Intent goBack=new Intent(getApplicationContext(),MainActivity.class);

    goBack.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TOP);
```

```
startActivity(goBack);  
  
this.finish();  
}
```

@Override

```
public void onClick(View v) {  
  
    switch (v.getId())  
    {  
  
        case R.id.storeAppDetailsButton:  
  
            Intent openStoreAppDetailsIntent=new Intent(getApplicationContext(),StoreAppDetails.class);  
  
            openStoreAppDetailsIntent.putExtra("Title", title);  
  
            openStoreAppDetailsIntent.putExtra("ImageUrl", imageUrl);  
  
            openStoreAppDetailsIntent.putExtra("VideoUrl", videoUrl);  
  
  
            startActivity(openStoreAppDetailsIntent);  
  
            break;  
  
        case R.id.backButton:  
  
            Intent goBack=new Intent(getApplicationContext(),RecognitionFinderActivity.class);  
  
            goBack.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TOP);  
  
            startActivity(goBack);  
  
            this.finish();  
  
            break;  
  
    }  
}
```

```

public class SentUIdTask extends AsyncTask<String, Void, String> {

    @Override

    protected void onPreExecute() {

        super.onPreExecute();

    }


    protected String doInBackground(String... params)

    {

        // TODO Auto-generated method stub

        Log.i("TAG", "URL : " + params[0]);


        ServiceHandler sh = new ServiceHandler();

        List<NameValuePair> nameValuePairsList = new ArrayList<NameValuePair>();

        nameValuePairsList.add(new BasicNameValuePair("uid", uid));


        String res = sh.makeServiceCall(params[0], ServiceHandler.GET, nameValuePairsList);

        return res;

    }


    @Override

```



```
protected void onPostExecute(String result) {

    // TODO Auto-generated method stub

    super.onPostExecute(result);

    Log.d("ResultJaanu", result);

    // Toast.makeText(StoreAppActivity.this, "Hi : "+result, Toast.LENGTH_SHORT).show();

    JSONObject jsonObject;

    try
    {

        jsonObject = new JSONObject(result);

        jsonObject = jsonObject.getJSONObject("data");

        title = jsonObject.getString("title");

        imageUrl = jsonObject.getString("image");

        videoUrl = jsonObject.getString("video");

        imageUrl = "http://192.168.0.7/image-recognition/"+imageUrl;

        videoUrl = "http://192.168.0.7/image-recognition/"+videoUrl;

        Log.d("ImageURL", imageUrl);

        titleTextView.setText(title);

    } catch (JSONException e) {

        e.printStackTrace();

    }

    progressDialog.cancel();
```

```
Picasso.with(getApplicationContext())  
    .load(ImageUrl)  
    .into(trophyPhoto);  
storeAppDetailsButton.setText(title);  
}  
}  
  
}
```

```
public class ServiceHandler  
{  
  
    static String response = null;  
    public static final int GET = 1;  
    public static final int POST = 2;  
    String newUrl, paramString="";  
  
    public ServiceHandler()  
    {  
    }  
  
    public String makeServiceCall(String url, int method,  
        List<NameValuePair> params)  
    {
```

```

try {

    // http client

    newUrl=url.toString();

    //DefaultHttpClient httpClient = new DefaultHttpClient();

    HttpEntity httpEntity = null;

    HttpResponse httpResponse = null;


    HttpClient httpClient = new DefaultHttpClient();
    SSLSocketFactory sf = (SSLSocketFactory)httpClient.getConnectionManager()
        .getSchemeRegistry().getScheme("https").getSocketFactory();
    sf.setHostnameVerifier(new AllowAllHostnameVerifier());


    // Checking http request method type
    if (method == POST)
    {

        HttpPost httpPost = new HttpPost(url);

        // adding post params
        if (params != null) {
            httpPost.setEntity(new UrlEncodedFormEntity(params));
        }


        httpResponse = httpClient.execute(httpPost);
    }
}

```

```

    } else {

        if (method == GET)

        {

            // appending params to url

            int totalCount = params.size();

            for(int count=0; count<totalCount; count++)

            {

                if(count==totalCount-1)

                {

                    paramString =
paramString+params.get(count).getName()+"="+params.get(count).getValue();

                }

                else

                {

                    paramString =
paramString+params.get(count).getName()+"="+params.get(count).getValue()+"&";

                }

            }

            if (params != null)

            {

                //Log.v("Theka : ", params.get(0).getValue());

```

```

String paramString = URLEncodedUtils.format(params,
"UTF-8");

//                                     //Below line only for Swedish Application, If you are
developing app in english please delete below line.

//                                     paramString=paramString.replace("%25", "%");
//                                     paramString=paramString.replace("%40", "@");
//                                     //End

url += "?" + paramString;

System.out.println("parameters : " + paramString);

} else {

    url = newUrl;

}

System.out.println("url ----- " + url);

HttpGet httpGet = new HttpGet(url);

httpResponse = httpClient.execute(httpGet);

}

}

httpEntity = httpResponse.getEntity();

response = EntityUtils.toString(httpEntity);

```

```
    }  
    catch (UnsupportedEncodingException e)  
    {  
        e.printStackTrace();  
    }  
    catch (ClientProtocolException e)  
    {  
        e.printStackTrace();  
    }  
    catch (IOException e)  
    {  
        e.printStackTrace();  
    }  
  
    return response;  
  
}  
  
}
```