Android e APIs / Web Service

WebAPI e Web Services

Vocês realmente tem o conceito?

ASync Tasks

Tarefas que não deixam o aplicativo em espera (execuções síncronas)

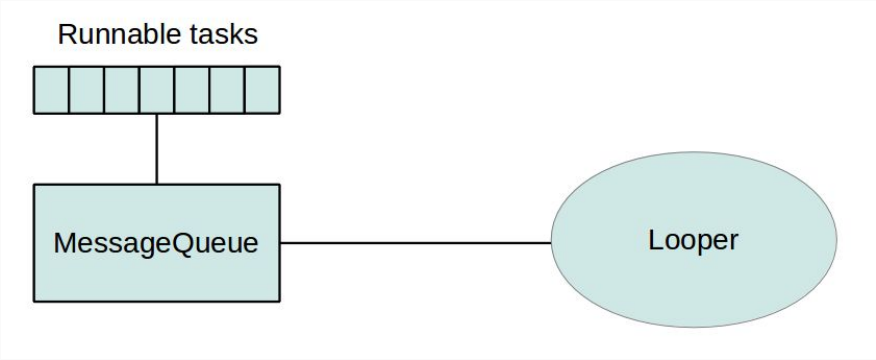
Já ouviram falar de AJAX? XMLHTTPResponse?

E Observables e Promises?

Vantagens

Usabilidade

Offline



Métodos Principais

doInBackground

onProgressUpdate

onPreExecute

onPostExecute

execute

Exercicio 1

Vamos descer uma imagem da internet e somente após o download exibi-la

OBS: Lembrem de dar permissão de internet no manifesto!

Base da activity

```
import ...

public class MainActivity extends AppCompatActivity {

    public ImageView iv;
    public Bitmap imagemEmBitmap;
    /* coloquem qq imagem, desde que em jpg, png e gif */
    public String furacao = "https://abrilexame.files.wordpress.com/2017/09/2017-08-31t171639z_7519
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        this.iv = (ImageView) findViewById(R.id.imageView);

    }

    /* metodo chamado diretamente pelo botao no XML
    Poderiam usar o onClickEventListener etc ...
     */
    public void comecaDownload( View view){
        new DownloadTask().execute();
    }
```

Metodo de download

Funcionará,

porém podem testar com

itens mais pesados que

verão o Android reclamar.

Aguardem Threads...

activity_main.xml × MainActivity.java × AndroidManifest.xml ×

```
// definindo tipo de async como AsyncTask<Params, Progress, Result>
//DO in - String = URL,
//Progress = nao estamos fazendo, necesario se fossemos colocar progress b
//POST - bitmap pelo retorno de imagem

public  class DownloadTask extends AsyncTask<String, Void, Bitmap>{

    @Override
    protected void onPreExecute(){
        super.onPreExecute();
    }

    @Override
    protected Bitmap doInBackground(String... URL) {
        // try pq podemos estar sem internet
        try {
            InputStream is = new java.net.URL(furacao).openStream();
            // conversor de imagens para bitmap, padrao do Resources
            imagemEmBitmap = BitmapFactory.decodeStream(is);
        }catch (Exception e){
            Log.e("Erro", e.toString()); //Coloquei para nao ficar vazio
        }
        return imagemEmBitmap;
    }

    @Override
    protected void onPostExecute(Bitmap o) {
        iv.setImageBitmap(o);
    }

}
}
```

JsonParser

```
import org.json.JSONArray;
import org.json.JSONObject;

String jsonString = readJsonObjectFromSomeWhere();
try {
    JSONObject json = new JSONObject(jsonString);
    } catch (Exception e) {
        e.printStackTrace();
    }
```

Exercicio 2

Consumir o JSON de https://reqres.in/ E exibir numa gridview

Lado Sacal

1-beginObject

2-beginArray

3-beginObject

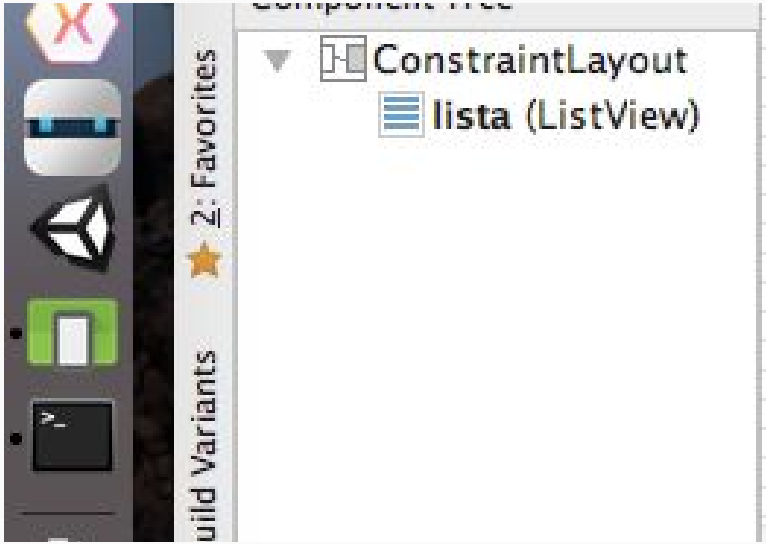
```
{
    "page": 2,
    "per_page": 3,
    "total": 12,
    "total_pages": 4,
    "data": [
        {
            "id": 4,
            "first_name": "Eve",
            "last_name": "Holt",
            "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/marcoramires/128.jpg"
        },
        {
            "id": 5,
            "first_name": "Charles",
            "last_name": "Morris",
            "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/stephenmoon/128.jpg"
        },
        {
            "id": 6,
            "first_name": "Tracey",
            "last_name": "Ramos",
            "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/bigmancho/128.jpg"
        }
    ]
}
```

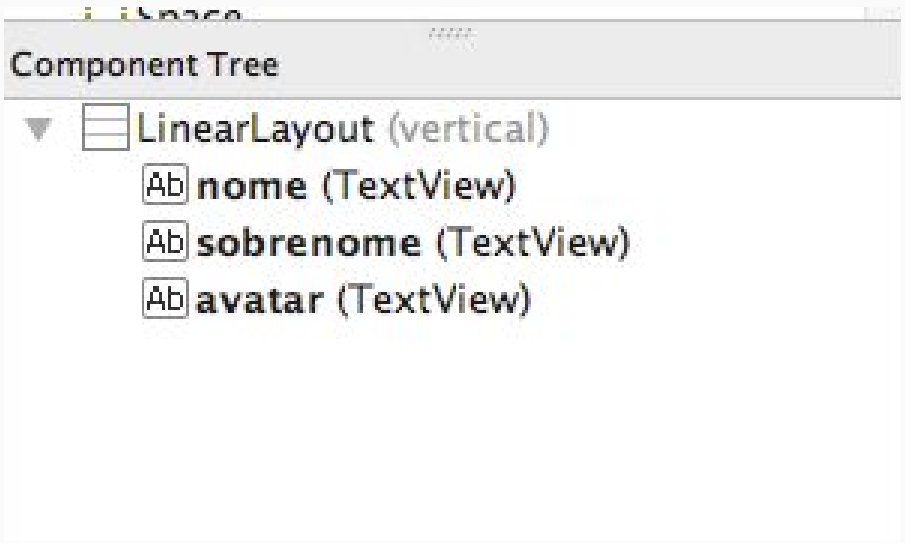
```
JsonReader jsonReader = new JsonReader(responseBodyReader);
jsonReader.beginObject();
while (jsonReader.hasNext()) {
    String key = jsonReader.nextName();
    if (key.equals("data")) {
        jsonReader.beginArray();
        while (jsonReader.hasNext()) {
            jsonReader.beginObject();
            while( jsonReader.hasNext() ) {
                // Aff, chegamos!
                final String innerInnerName = jsonReader.nextName();
                String name = jsonReader.nextName();
                Log.d(name, name);
            }
        }
        break;
    } else {
        jsonReader.skipValue();
    }
}
```

Opção 1 - Biblioteca Propria

Utilizar uma biblioteca que extraia o JSOn como String e permitia utilizar como JsonObject

XML da lista de dos Itens





Codigo base da MainActivity

```
public class MainActivity extends AppCompatActivity {

    /* criando um array de dicionario string string */
    ArrayList<HashMap<String, String>> listaReqRes;
    public String urlApi = "https://reqres.in/api/users?page=2";
    public ListView lv;
    public String jsonStr;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("a","a");
        /* array vazio no inicio para nao bugar o adapter*/
        listaReqRes = new ArrayList<>();
        lv = (ListView) findViewById(R.id.lista);
        /*chamando a execucao em background */
        new pegaJson().execute();
    }
```

Chamada ao Json

Direto pelo nome do attributo

Array de hashes para mapear os campos

```
private class pegaJson extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() { }

    @Override
    protected Void doInBackground(Void... arg0) {
        try {
            HttpHandler sh = new HttpHandler();
            jsonStr = sh.makeServiceCall(urlApi);
            JSONObject jsonObj = new JSONObject(jsonStr);

            // Pegando nó de dados
            JSONArray contacts = jsonObj.getJSONArray("data");

            // Iterando pelos usuarios
            for (int i = 0; i < contacts.length(); i++) {
                HashMap<String, String> item = new HashMap<>();
                JSONObject c = contacts.getJSONObject(i);
                String id = c.getString("id");
                String nome = c.getString("first_name");
                String sobrenome = c.getString("last_name");
                String avatar = c.getString("avatar");
                item.put("id", id);
                item.put("nome", nome);
                item.put("sobrenome", sobrenome);
                item.put("avatar", avatar);
                listaReqRes.add(item);
            }
        } catch (JSONException j) {
            Log.e("erro", j.toString());
        }
        return null;
    }
```

Exibicao no Adapter

```
    @Override
    protected void onPostExecute(Void result) {
        super.onPostExecute(result);
        ListAdapter adapter = new SimpleAdapter(
                MainActivity.this, listaReqRes,
                R.layout.item_lista, new String[]{"nome", "sobrenome",
                "avatar"}, new int[]{
                    R.id.nome,
                    R.id.sobrenome,
                    R.id.avatar
                });
        lv.setAdapter(adapter);
    }

}
```