FlyHttp

Library for http client requests and server with Android.

Repositório:
https://github.com/mariodearaujocarvalho/FlyHttp

With <3 : Mário Carvalho

FlyHttp

Oque é:

É uma biblioteca simples, para fazer requisições web no Android de forma simples e prática.

FlyHttp

Onde está:

https://github.com/mariodearaujocarvalho/FlyHttp

FlyHttp

Licença de Uso:

FlyHttp

Vantagens:

- **Requisição em menos de 10 linhas**
- **Simples**
- **Poderosa**
- **Maravilhosa**
- **Orientada a Objetos**
- **JSON String to ArrayList em 1 linha**
- **OpenSource**
- **Interfaces de Callback**
- **Tratamento de erros**
- **Ferramentas utilitárias pra internet em Android**

FlyHttp

Desvantagens:

- **Ainda não está no Gradle :’(**
- **1 aninho de idade**

FlyHttp

Configurando no projeto:

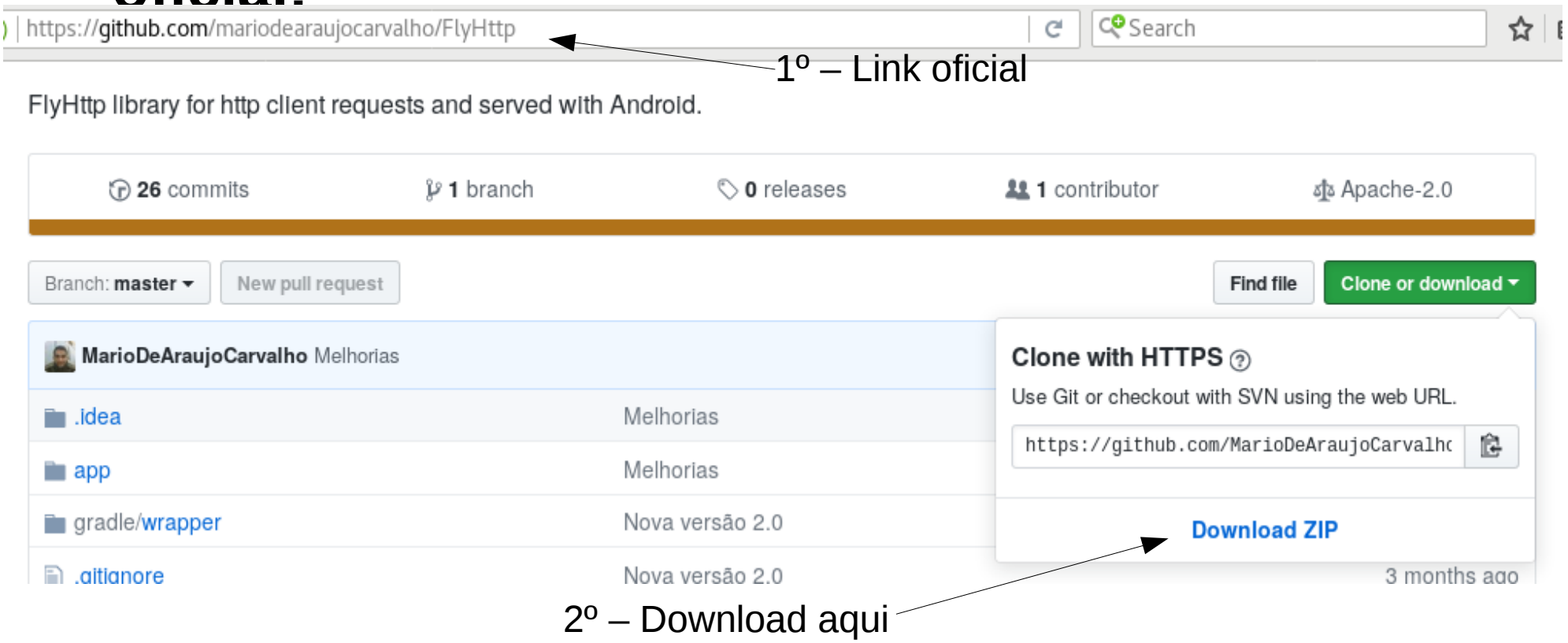
Tecnicamente são apenas 6 passos:

- **1 – Baixar a biblioteca**
- **2 – Adicionar a biblioteca ao seu projeto**
- **3 – Adicionar dependências GSON e Volley**
- **4 – Usar**
- **5 – Compartilhar com os amigos e ser feliz**

FlyHttp

1 – Baixar a biblioteca:

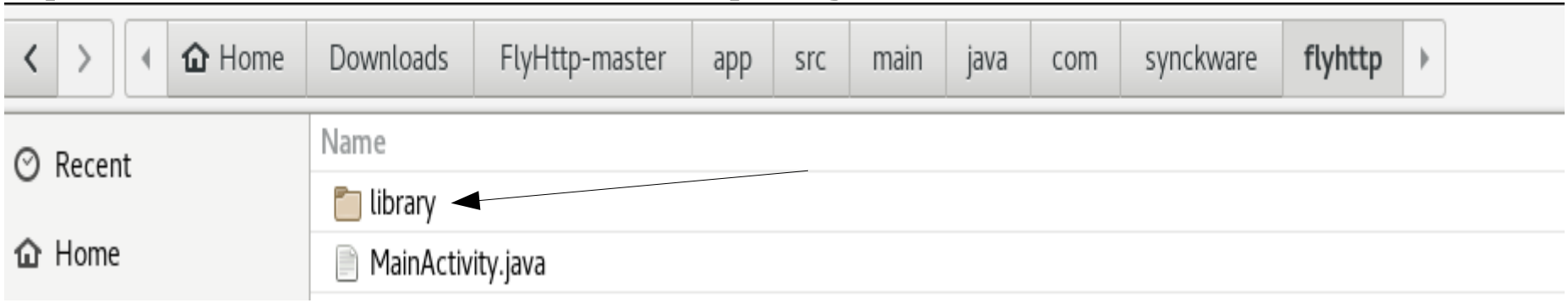
A biblioteca pode ser baixada no repositório oficial:



FlyHttp

2 - Adicionar a biblioteca ao seu projeto:

1) Descompactar o zip
2) Abrir pasta: ../FlyHttpmaster/app/src/main/java/com/synckware
3) Copiar diretório library
4) Colar na raiz do seu projeto



FlyHttp

3 – Adicionar dependências GSON e Volley:

Ela utiliza o Volley para facilitar o envio de dados para web. Logo, as únicas dependências que você terá de adicionar será a da biblioteca Volley e GSON do Google.

- **1 – Abrar o build.gradle**

- **2 – Adionar as dependências:**
  - **compile 'com.android.volley:volley:1.0.0'**
  - **compile 'com.google.code.gson:gson:2.6.2'**

FlyHttp

4 – Usar - Instâncias:

```
//Exemplo de uma requisição simples com método POST e retorno em String
String URL_WEB_SERVICE = "https://api.ipify.org/?format=json";
FlyHttp http = new FlyHttp(Metode.POST, URL_WEB_SERVICE, context: MainActivity.this);
//ProgressDialog (Opicional):  poder ser ativado via método set ou pelo construtor
http.setWithProgress(true);

//Criando um formulário para passar dados pra web - opicional
FormKeyValue<String, String> params = new FormKeyValue<String, String>();
//Setando os valores no formulário do tipo chave e valor
params.put("Key", "Value");
```

FlyHttp

4 – Usar – Retornos: String:

```
1  //Retorno em String ou JSON, ambos com CallBack
2  http.build(new OnCallbackResponseString() { // String
3      @Override
4      public void onSuccessString(String result) throws Exception {
5          String resultdo = result;
6      }
7      @Override
8      public void onError(String result) throws Exception {
9          String erro = result;
10     }
11 });
```

FlyHttp

4 – Usar – Retornos: JSON:

```
1  http.build(new OnCallbackResponseJson() {// JSON
2      @Override
3      public void onSuccessJSONObject(JSONObject result) throws JSONException {
4          String ip = String.format("Result is: %s", result.getString("ip");
5        }
6      @Override
7      public void onError(String result) throws Exception {
8          String erro = result;
9      }
10 });
```

FlyHttp

4 – Usar – JSON String to ArrayList:

1 - Domain Class

```
1  package br.embrapa.exampleflyhttp;
2  import com.google.gson.annotations.SerializedName;
3  import java.io.Serializable;
4  public class Item implements Serializable{
5      @SerializedName("id")
6      public int id;
7      @SerializedName("name")
8      public int name;
9      public Item() {}
10 }
```

2 – Use

```
1  //Tratamento avançado dos retornos em String JSON: JSONString to ArrayList of Class's
2  final ArrayList<Item> mDataArrayList =
3              new ConverterJSONToArray<Item>().toArrayList(result,Item.class);
4  //...
5  Item item = new Item();
6  item.id = mDataArrayList.get(0).id;
7  item.name = mDataArrayList.get(0).name;
```

FlyHttp

4 – Usar – Classe Utilitária:

```
10  InternetUtil.openURL(Context context, String url)
11  InternetUtil.checkConnection(Context context) : return boolean;
12  /*.ConnectivityManager
13  *.TYPE_MOBILE 0
14  *.TYPE_WIFI 1
15  *.TYPE_WIMAX 6
16  *.TYPE_ETHERNET 9
17  */
```

FlyHttp

5 - Compartilhar com os coleguinhas...

FlyHttp: Uma forma inteliguente de tratar web services em Android

#javaneiros
#FlyHttp

E agora?

Macho, vá lá...

Tenha isso no coração...

“O mundo não é um grande arco-íris...
...É o reconhecimento FaceID do iPhone X.”
– Adaptado de Rudson Lima, Liveo-O.