Desenvolvimento de Apps para Mobile

Fabrício Tonetto Londero

Aula 11 – API Rest

Projeto 1 - Vamos fazer a clássica tela de cadastro de endereço

- O CEP é informado e com isso os demais campos de endereço são preenchidos automaticamente com os dados retornados do WebService
- Para isso, vamos utilizar a plataforma ViaCEP
  - https://viacep.com.br/

Para esse projeto

- No AndroidManifest, vamos adicionar dentro da tag Application:

```
android:usesCleartextTraffic="true"
```

- Isso possibilita o uso de requisições em texto simples e uso do protocolo http.

Permissões

- No AndroidManifest

```
<uses-permission android:name="android.permission.INTERNET" />
```

Activity

Classe conforme o retorno da API

- https://github.com/Ernakh/AppAndroidStudioJava/blob/main/app/src/main/java/com/ernakh/aplicativoaula/Endereco.java


```
{
  "cep": "97010-000",
  "logradouro": "Rua Venâncio Aires",
  "complemento": "até 1312 - lado par",
  "unidade": "",
  "bairro": "Centro",
  "localidade": "Santa Maria",
  "uf": "RS",
  "estado": "Rio Grande do Sul",
  "regiao": "Sul",
  "ibge": "4316907",
  "gia": "",
  "ddd": "55",
  "siafi": "8841"
}
```


```
3  public class Endereco {
4
5      String logradouro;
6      String bairro;
7      String localidade;
8      String uf;
9      String regiao;
10     String ddd;
11     String siafi;
12     String ibge;
13     String gia;
14
15     public String getLogradouro() {
16         return logradouro;
17     }
18
19     public void setLogradouro(String logradouro) {
20         this.logradouro = logradouro;
21     }
22
23     public String getBairro() {
24         return bairro;
25     }
```

Java – método OnCreate – Parte 1

```
EditText editText = findViewById(R.id.editCEP);
editText.setOnFocusChangeListener(new View.OnFocusChangeListener()
{
    @Override
    public void onFocusChange(View v, boolean hasFocus)
    {
        if (!hasFocus)
        {
```

OnCreate – Parte 2

```
new Thread(new Runnable() {
    @Override
    public void run() {
        try {
            EditText editText = findViewById(R.id.editCEP);
            URL url = new URL("http://viacep.com.br/ws/" + editText.getText() + "/json");
            HttpURLConnection conexao = (HttpURLConnection) url.openConnection();
```

OnCreate – Parte 3

```
if (conexao.getResponseCode() != 200)
    throw new RuntimeException("HTTP error code : " + conexao.getResponseCode());

BufferedReader resposta = new BufferedReader(new
InputStreamReader((conexao.getInputStream())));
String aux, jsonEmString = "";
while ((aux = resposta.readLine()) != null) {
    jsonEmString += aux;
}

String finalJsonEmString = jsonEmString;
```

OnCreate – Parte 4

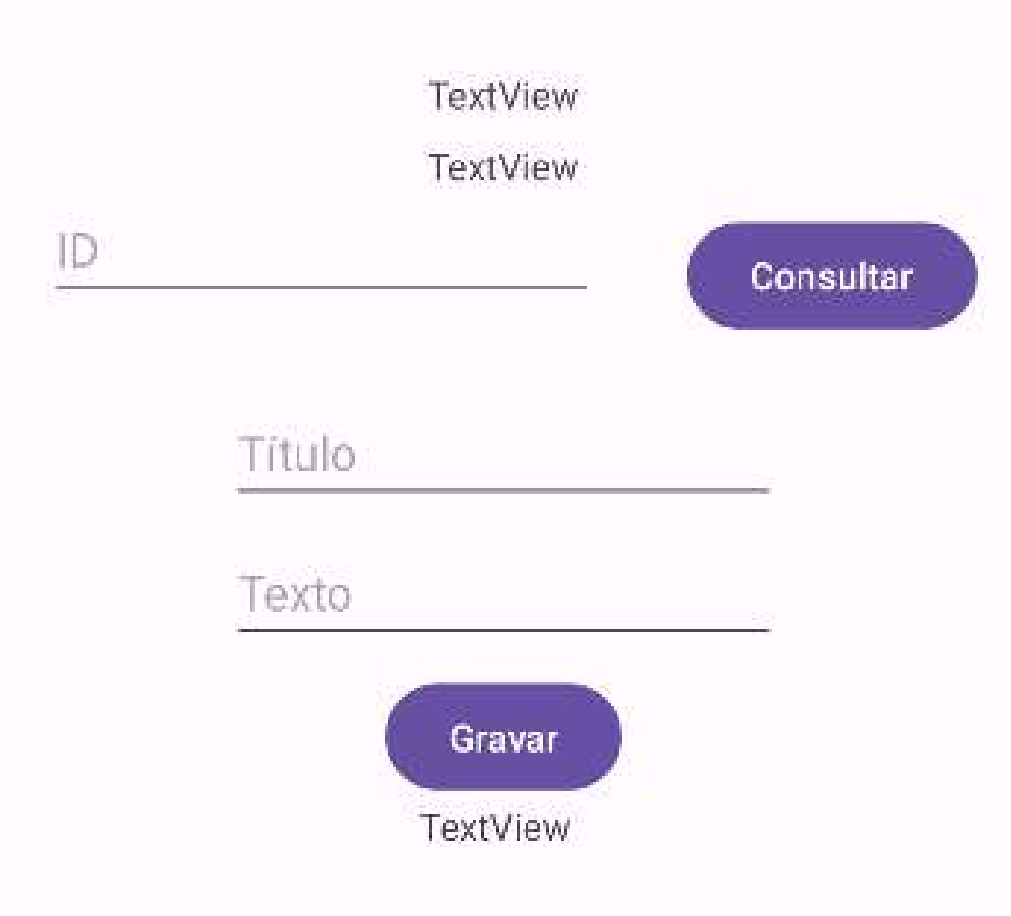
```
runOnUiThread(new Runnable() {
    @Override
    public void run() {
        TextView textView = findViewById(R.id.txvEndereco);
        textView.setText(finalJsonEmString);

        Log.d("JSON", "JSON - antes do GSON");

        Gson gson = new Gson();
        Endereco endereco = gson.fromJson(finalJsonEmString, Endereco.class);

        EditText editTextText4 = findViewById(R.id.editTextText4);
        editTextText4.setText(endereco.getUf());

        Log.d("JSON", "JSON - final");
    }
});
```

OnCreate – Parte 5

```
} catch (Exception e) {
                        Log.d("JSON", "JSON - erro: " + e.getMessage());
                        e.printStackTrace();
                    }
                }
            }).start();
        }
    }
  });
  }
}
```

Projeto 2

- Get e Post
- Vamos simular o cadastro e consulta de Posts de um blog através do consumo de API
- Vamos utilizar a API pública JSPNPlaceHolder
- https://jsonplaceholder.typicode.com/posts



Classe Post

- Conforme o JSON retornado


```
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae
ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
  },
```

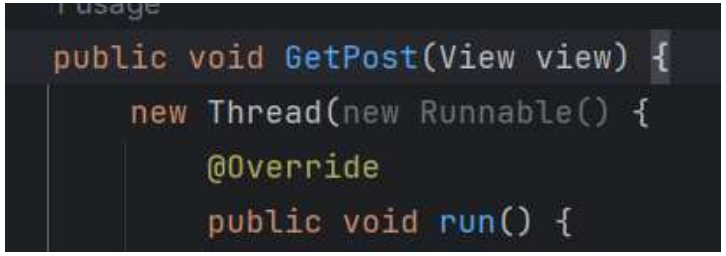
Classe Post

- https://github.com/Ernakh/AppAndroidStudioJava/blob/main/app/src/main/java/com/ernakh/aplicativoaula/Post.java

```java
package com.ernakh.aplicativoaula;

public class Post {
    private int userId;
    private int id;
    private String title;
    private String body;

    public void setUserId(int userId) {
        this.userId = userId;
    }

    public void setId(int id) {
        this.id = id;
    }

    public void setTitle(String title) {
        this.title = title;
    }
```

Activity

- Vamos criar dois novos métodos, um para Get e Outro para Post



Get – Parte 1

```
public void GetPost(View view) {
    new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                EditText edtID = findViewById(R.id.edtID);
                URL url = new URL("https://jsonplaceholder.typicode.com/posts/" + edtID.getText());

                HttpURLConnection conexao = (HttpURLConnection) url.openConnection();
                conexao.setRequestMethod("GET");
```

Get – Parte 2

```
int responseCode = conexao.getResponseCode();
if (responseCode == 200) {
    BufferedReader in = new BufferedReader(new InputStreamReader(conexao.getInputStream()));
    String inputLine;
    StringBuilder response = new StringBuilder();

    while ((inputLine = in.readLine()) != null) {
        response.append(inputLine);
    }
    in.close();

    String resultado = response.toString();
```

Get – Parte 3

```
Gson gson = new Gson();
Type tipoPost = new TypeToken<Post>() {}.getType();
Post post = gson.fromJson(resultado, tipoPost);

runOnUiThread(new Runnable() {
    @Override
    public void run() {
        TextView textView5 = findViewById(R.id.textView5);
        TextView textView6 = findViewById(R.id.textView6);
        textView5.setText(post.getTitle());
        textView6.setText(post.getBody());
        Log.d("JSON", resultado);
    }
});
```

Get – Parte 4 – testar!

```
} else {
                Log.e("API", "Erro de conexão: " + responseCode);
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}).start();
}
```

Método Post – Parte 1

```
public void PostPost(View view) {
    new Thread(() -> {
        try {
            EditText edtTitulo = findViewById(R.id.edtTitulo);
            EditText edtCorpo = findViewById(R.id.edtTexto);

            Post novoPost = new Post();
            novoPost.setUserId(1);
            novoPost.setTitle(edtTitulo.getText().toString());
            novoPost.setBody(edtCorpo.getText().toString());
```

Método Post – Parte 2

```
Gson gson = new Gson();
String jsonPost = gson.toJson(novoPost);

URL url = new URL("https://jsonplaceholder.typicode.com/posts");
HttpURLConnection conexao = (HttpURLConnection) url.openConnection();
conexao.setRequestMethod("POST");
conexao.setRequestProperty("Content-Type", "application/json; charset=UTF-8");
conexao.setDoOutput(true);
```

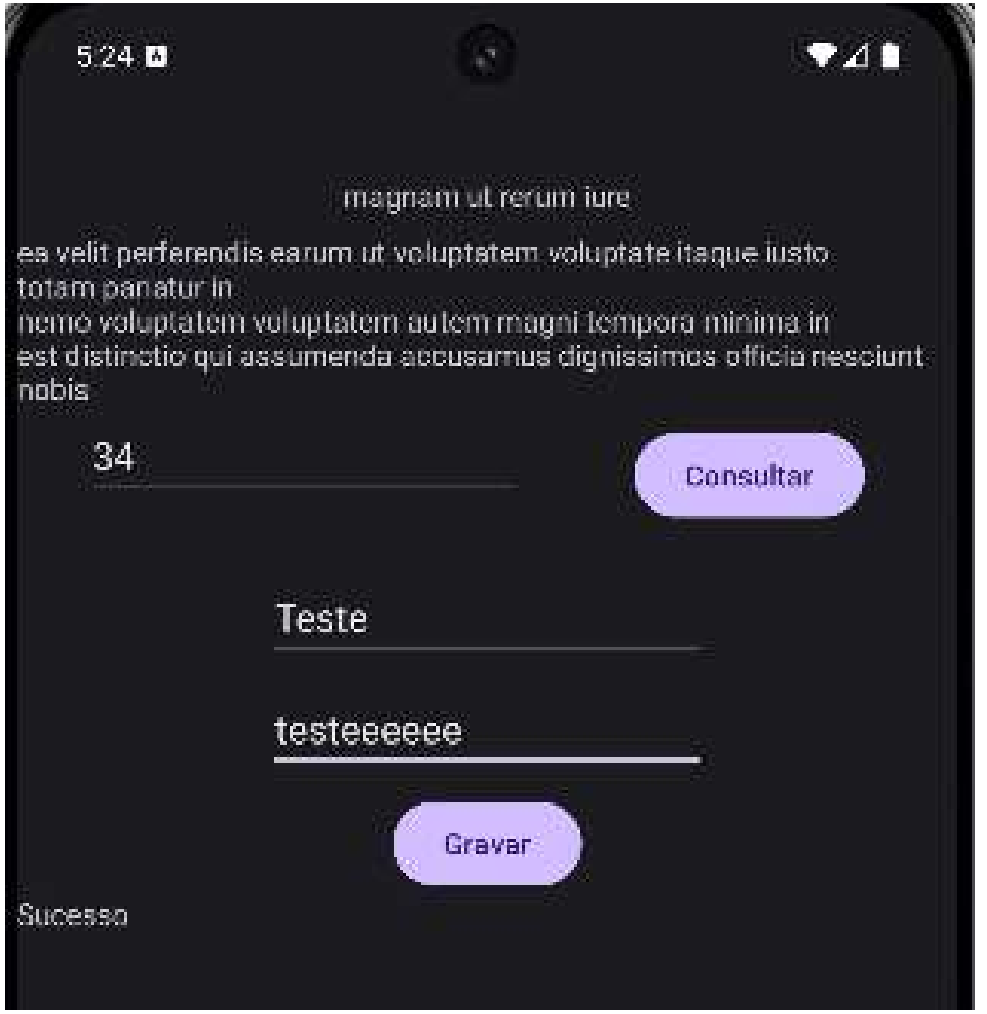
Método Post – Parte 3

```
OutputStream os = conexao.getOutputStream();
os.write(jsonPost.getBytes("UTF-8"));
os.close();

int responseCode = conexao.getResponseCode();
TextView txtResultado = findViewById(R.id.textView8);

if (responseCode == 201) {
    txtResultado.setText("Sucesso ao gravar o POST!");
} else {
    txtResultado.setText("Erro ao gravar o POST!");
}
```

Método Post – Parte 4

```
} catch (Exception e) {
            Log.d("Erro", e.getMessage());
            TextView txtResultado = findViewById(R.id.textView8);
            txtResultado.setText("Erro: " + e.getMessage());
        }

    }).start();
  }
}
```

Testar!


5:24
magnam ut rerum iure
ea velit perferendis earum ut voluptatem voluptate itaque iusto
totam pariatur in
nemo voluptatem voluptatem autem magni tempora minima in
est distinctio qui assumenda accusamus dignissimos officia nesciunt
nobis
34
Consultar
Teste
testeeeeee
Gravar
Sucesso

Atividade 1

- Acesse o https://jsonplaceholder.typicode.com/ e implemente o PUT e o DELETE

Atividade 2

- Implemente o CRUD em uma das seguintes APIs gratuitas:
  - https://reqres.in/
  - https://dummyjson.com/