

Desenvolvimento Para Dispositivos móveis



Prof. Me. Clênio Silva
E-mail: clenio.silva@uniube.br

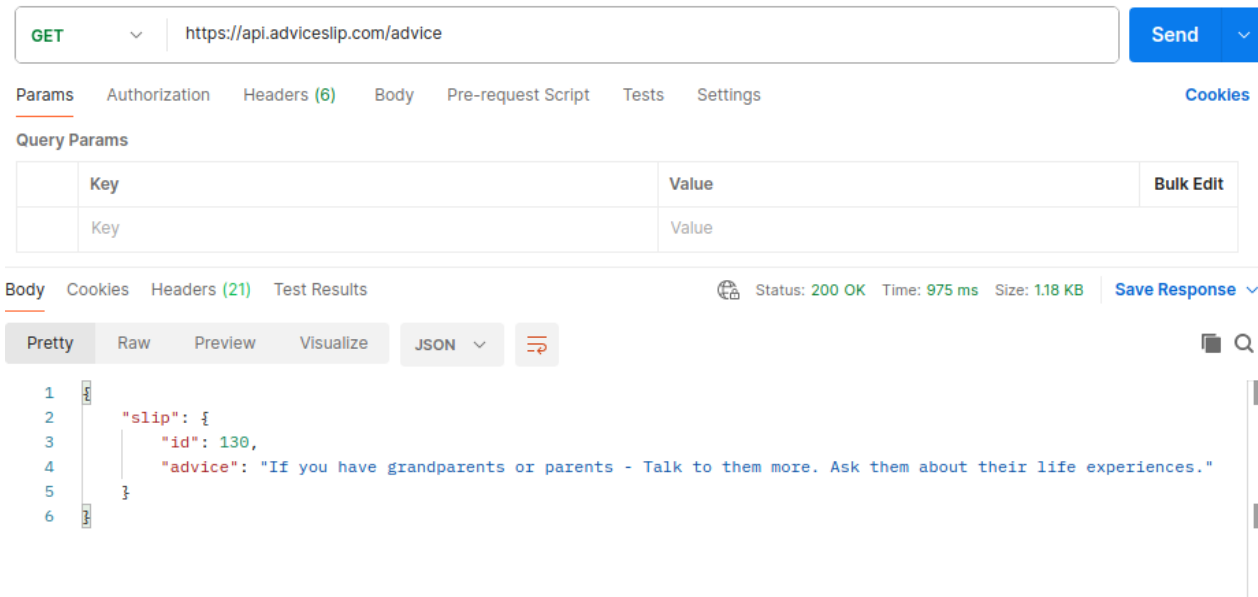
Integração aplicação Android com Api externa

Criar um aplicativo Android que integra com APIs externas para coletar e mostrar dados envolve várias etapas

- Primeiro, você precisa configurar seu projeto no Android Studio.
 - Crie um Novo Projeto:
 - Abra o Android Studio.
 - Dê um nome ao seu aplicativo de **IntegracaoApi**, escolha o Java como linguagem de programação, e finalize a criação do projeto.

Integração aplicação Android com Api externa

Vamos criar um app que integra com uma api aberta <https://api.adviceslip.com/>. A API Advice Slip é fornecida gratuitamente e distribui conselhos em inglês.



The screenshot displays a REST client interface with the following details:

- Method:** GET
- URL:** `https://api.adviceslip.com/advice`
- Buttons:** Send, Cookies
- Tabs:** Params, Authorization, Headers (6), Body, Pre-request Script, Tests, Settings
- Query Params Table:**

Key	Value	Bulk Edit
Key	Value	
- Status Bar:** Status: 200 OK, Time: 975 ms, Size: 1.18 KB, Save Response
- Response Body (JSON):**

```
1 {
2   "slip": {
3     "id": 130,
4     "advice": "If you have grandparents or parents - Talk to them more. Ask them about their life experiences."
5   }
6 }
```

Integração aplicação Android com Api externa

- Para acessar APIs externas, você precisa adicionar a permissão de internet no arquivo **AndroidManifest.xml**.

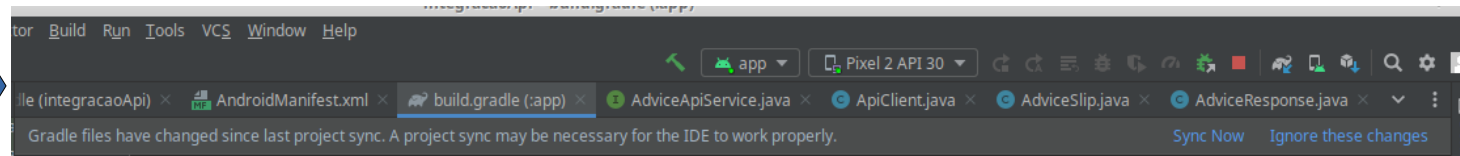
```
</application>  
  
<uses-permission android:name="android.permission.INTERNET" />  
  
</manifest>
```

Integração aplicação Android com Api externa

- Dependendo do tipo de APIs que você vai usar, você provavelmente precisará de algumas bibliotecas adicionais. A biblioteca **Retrofit** é uma escolha popular para fazer requisições HTTP em Android.
 - Adicione as seguintes dependências no arquivo build.gradle (Module: app):

```
implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
```

Assim que adicionado as dependências. Logo abaixo da barra superior aparecerá uma opção para fazer a atualização com o download das dependências, **Sync now**



Integração aplicação Android com Api externa

- Para mapear o JSON de resposta da chamada do endpoint /advice para objetos Java, você precisa de duas classes: **AdviceResponse** e **AdviceSlip**. Crie as classes de modelo que correspondem à estrutura dos dados da API.

Integração aplicação Android com Api externa

- **AdviceSlip**

```
package com.example.integracaoapi;
import com.google.gson.annotations.SerializedName;

3 usages
public class AdviceSlip {
    1 usage
    @SerializedName("id")
    private int id;

    1 usage
    @SerializedName("advice")
    private String advice;

    // Getters
    public int getId() { return id; }

    1 usage
    public String getAdvice() { return advice; }
}
```

Integração aplicação Android com Api externa

- **AdviceResponse**

```
package com.example.integracaoapi;
import com.google.gson.annotations.SerializedName;

6 usages
public class AdviceResponse {
    1 usage
    @SerializedName("slip")
    private AdviceSlip slip;

    // Getter
    1 usage
    public AdviceSlip getSlip() { return slip; }
}
```


Integração aplicação Android com Api externa

- **AdviceSlip** é a classe que contém os campos id e advice. Esses campos são anotados com `@SerializedName` para mapear corretamente os nomes das propriedades JSON para os nomes das variáveis Java.
- **AdviceResponse** é a classe que contém uma instância de `AdviceSlip`. Esta classe mapeia o JSON principal, que possui um único objeto slip.

Integração aplicação Android com Api externa

- Defina uma interface para a API que você está acessando. No nosso caso estamos acessando uma API que retorna um conselho e seu Id, assim usaremos a classe com o modelo de resposta que criamos anteriormente:

```
package com.example.integracaoapi;

import retrofit2.Call;
import retrofit2.http.GET;

2 usages
public interface AdviceApiService {
    1 usage
    @GET("advice")
    Call<AdviceResponse> getAdvice();
}
```

Integração aplicação Android com Api externa

- Agora vamos configurar o **Retrofit** no seu aplicativo. Você pode fazer isso em uma classe singleton para garantir que exista apenas uma instância do Retrofit durante a execução do aplicativo.

Integração aplicação Android com Api externa

- **Configuração do Retrofit:**

```
package com.example.integracaoapi;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

1 usage
public class ApiClient {
    1 usage
    private static final String BASE_URL = "https://api.adviceslip.com/";
    3 usages
    private static Retrofit retrofit;

    1 usage
    public static Retrofit getRetrofitInstance() {
        if (retrofit == null) {
            retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();
        }
        return retrofit;
    }
}
```

Integração aplicação Android com Api externa

- Atualize o layout da atividade principal (activity_main.xml) para incluir um TextView onde o conselho será exibido:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/adviceTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:text="Loading advice..." />

</LinearLayout>
```

Integração aplicação Android com Api externa

- No MainActivity, você pode fazer a requisição para a API e lidar com a resposta.

```
package com.example.integracaoapi;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

2 usages
public class MainActivity extends AppCompatActivity {
    2 usages
    private AdviceApiService apiService;
    3 usages
    private TextView adviceTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        adviceTextView = findViewById(R.id.adviceTextView);
    }
}
```

Integração aplicação Android com Api externa

```
apiService = ApiClient.getRetrofitInstance().create(AdviceApiService.class);

getAdvice();
}

1 usage
private void getAdvice() {
    Call<AdviceResponse> call = apiService.getAdvice();
    call.enqueue(new Callback<AdviceResponse>() {
        @Override
        public void onResponse(Call<AdviceResponse> call, Response<AdviceResponse> response) {
            if (response.isSuccessful() && response.body() != null) {
                AdviceSlip adviceSlip = response.body().getSlip();
                adviceTextView.setText(adviceSlip.getAdvice());
            }
        }

        @Override
        public void onFailure(Call<AdviceResponse> call, Throwable t) {
            adviceTextView.setText("Failed to load advice.");
        }
    });
}
```

Integração aplicação Android com Api externa

