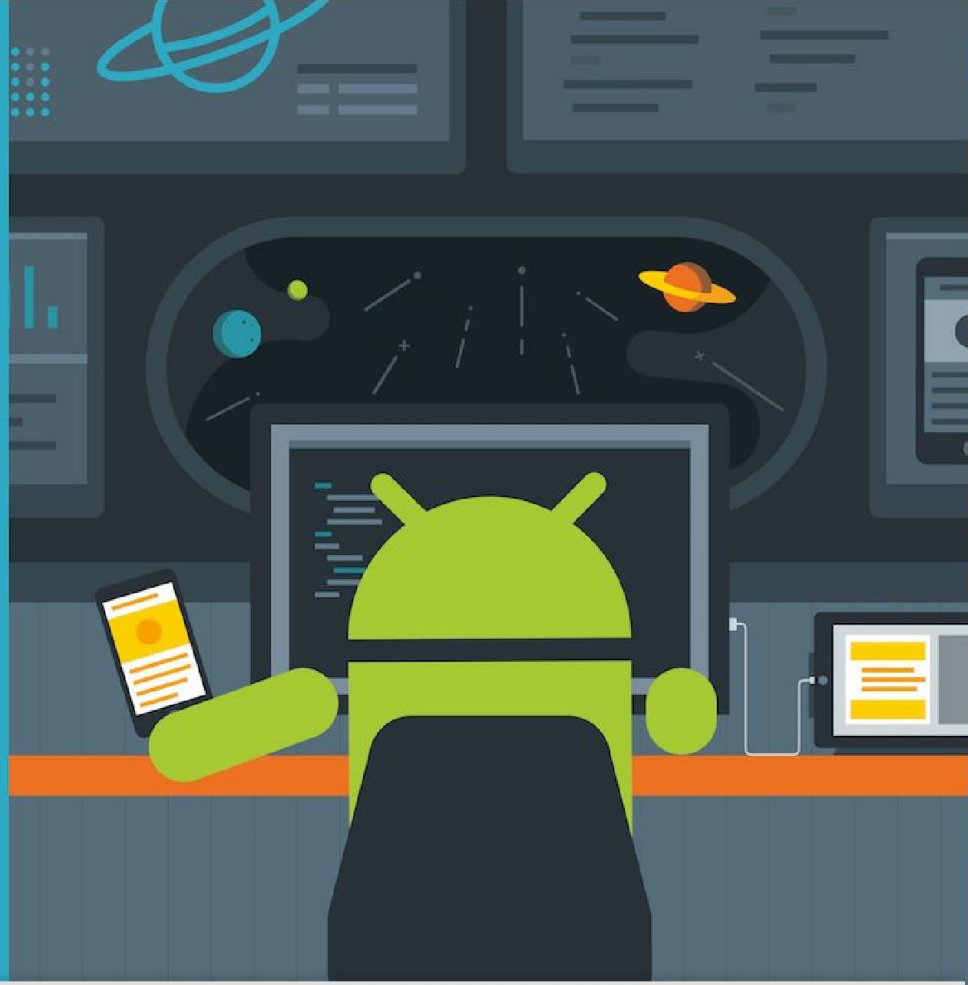
Developpement Android avancé

Consommation d’une API

Retrofit v 2.x

Et volley



La librairie Retrofit v2.x

Conten

- La librairie Retrofit v2.x
- Qu’est ce qu’une API
- Exemple Android avec Retrofit

La librairie Retrofit

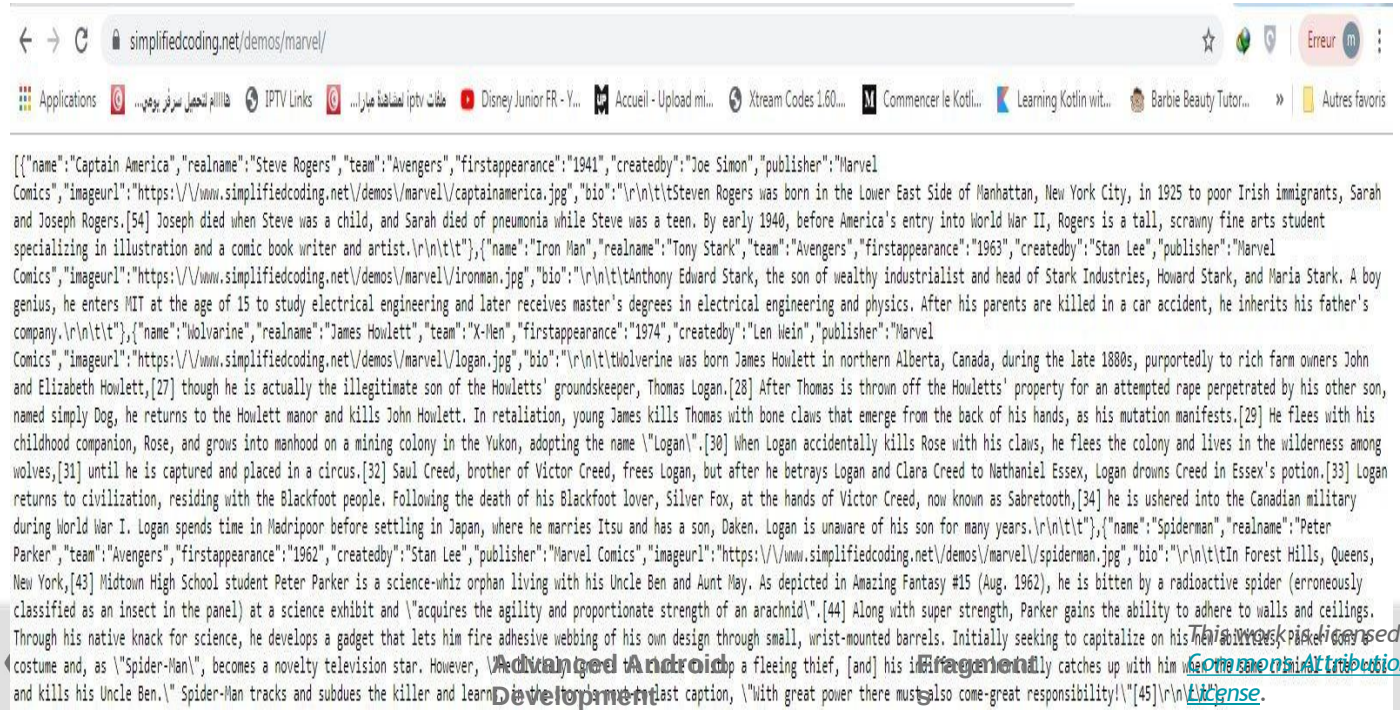
- Une librairie java qui nous permet de consommer des API
- Facilite le travail et permet de s'en passer de gestion de processus asynchrones, mémoires, cache….

AP

1. Un URL qui nous permet d'accéder à des données quelconques en format json ou autre via un http request
2. L'api permet de gérer à une BD quelconque
3. Peut être développé en utilisant différents langages de programmation : J2ee, php, csharp…

Exemple

https://simplifiedcoding.net/demos/marvel/



```
[
  {
    "name": "Captain America",
    "realname": "Steve Rogers",
    "team": "Avengers",
    "firstappearance": "1941",
    "createdby": "Joe Simon",
    "publisher": "Marvel Comics",
    "imageurl": "https://www.simplifiedcoding.net/demos/marvel/captainamerica.jpg",
    "bio": "\r\n\t\tSteven Rogers was born in the Lower East Side of Manhattan, New York City, in 1925 to poor Irish immigrants,
  },
  .
  .
  .
```

Example Android

Ajouter les dépendances

1. Ajouter la librairie Retrofit et Gson

```
implementation 'com.squareup.retrofit2:retrofit:2.3.0'
implementation 'com.squareup.retrofit2:converter-gson:2.2.0'
```

Internet Permission

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

Créer l'interface Api

2. Créer l'interface de l'API Api.java

L’URL root et le call

2. Créer L’URL Root pour l’API

3. On a http Get Request

4.On définit une méthode : le type de retour est Call<List<Hero>>

```
public interface Api {

    String BASE_URL = "https://simplifiedcoding.net/demos/";

    @GET("marvel")
    Call<List<Hero>> getHeroes();
}
```

Créer la classe Model

La classe Hero avec des attributs de même nom que dans le fichier json

```java
package com.example.retrofitexemple;

public class Hero {


    private String name;
    private String realname;
    private String team;
    private String firstappearance;
    private String createdby;
    private String publisher;
    private String imageurl;
    private String bio;

    public Hero(String name, String realname, String team, String firstappearance, String createdby, String publisher, String imageurl, 
        this.name = name;
        this.realname = realname;
        this.team = team;
        this.firstappearance = firstappearance;
        this.createdby = createdby;
        this.publisher = publisher;
        this.imageurl = imageurl;
        this.bio = bio;
    }
```

```java
@SerializedName("name")
private String name;
```

En cas ou les deux noms ne correspondent pas ajouter @Serialized

Appel de l'API dans le MainActivity

5. Appel de l'Api dans le MainActivity

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


    //calling the method to display the heroes
    getHeroes();
}

private void getHeroes() {
```

Appel de l'API dans le MainActivity

5. Appel de l'Api dans le MainActivity

```
private void getHeroes() {


    Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(Api.BASE_URL)
            .addConverterFactory(GsonConverterFactory.create()) //Here we are using the GsonConverterFac
            .build();
```

Créer une instance de retrofit avec l'URL de base

Appel de l'API dans le MainActivity

5. Créer une instance de retrofit avec l'URL de base

```
private void getHeroes() {


    Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(Api.BASE_URL)
            .addConverterFactory(GsonConverterFactory.create()) //Here we are using the GsonConverterFac
            .build();
```

Appel de l'API dans le MainActivity

5. Créer le Api interface

```
private void getHeroes() {


    Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(Api.BASE_URL)
            .addConverterFactory(GsonConverterFactory.create()) //Here we are using the Gson
            .build();

    Api api = retrofit.create(Api.class);

    Call<List<Hero>> call = api.getHeroes();
```

Appel de l'API dans le MainActivity

5. Créer l'objet call avec la méthode de l'interface déjà crée

```
Api api = retrofit.create(Api.class);

Call<List<Hero>> call = api.getHeroes();
```

Appel de l'API dans le MainActivity

6. Exécuter le call

```
Call<List<Hero>> call = api.getHeroes();

call.enqueue(new Callback<List<Hero>>() {
    @Override
    public void onResponse(Call<List<Hero>> call, Response<List<Hero>> response) {

        List<Hero> herolist =response.body();

    }

    @Override
    public void onFailure(Call<List<Hero>> call, Throwable t) {
        Toast.makeText(getApplicationContext(), t.getMessage(), Toast.LENGTH_SHORT).show();

    }
});
```

La librairie Volley

Conten

- La librairie Volley
- Exemple Android avec Volley

La librairie Volley

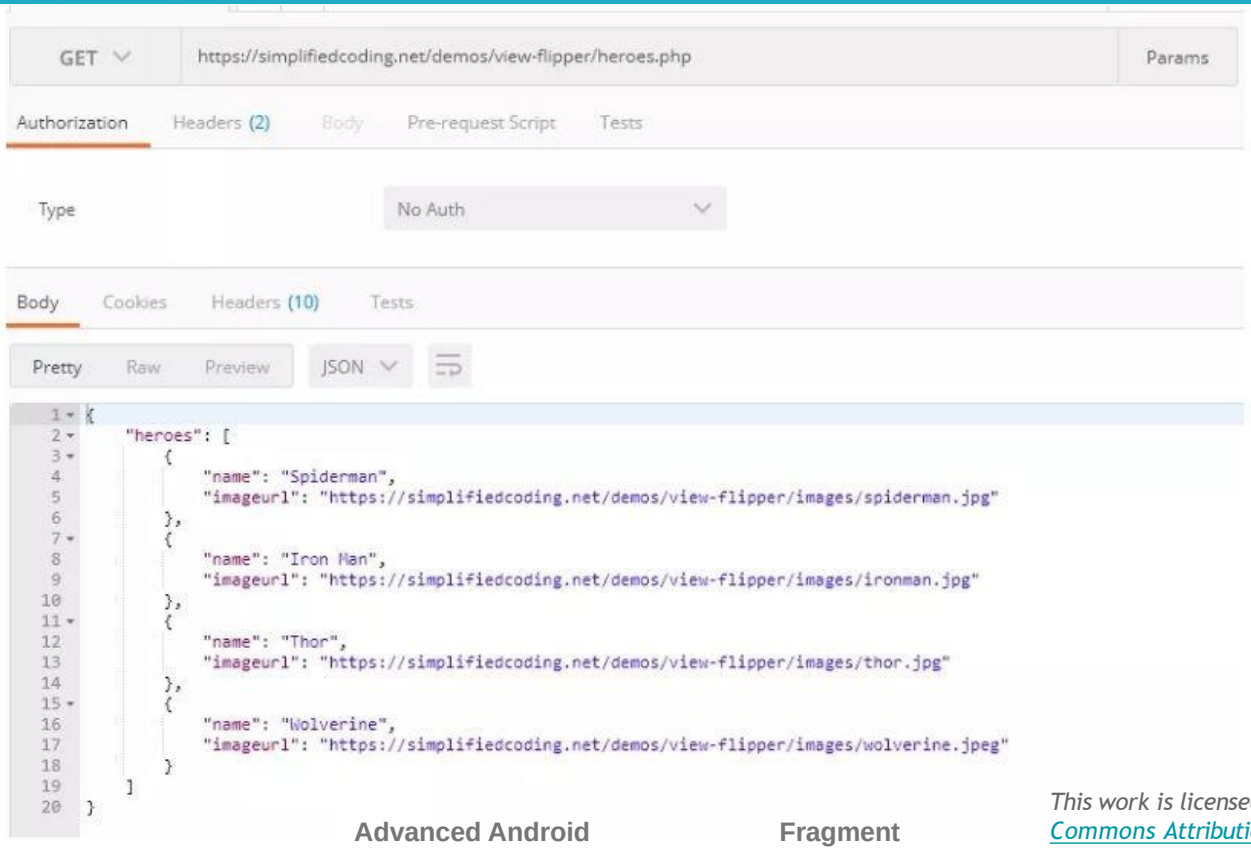
- La librairie Volley
- Exemple Android avec Volley

Qu’est ce que Volley

- Librairie http qui rend l’accès au réseau simple et facile:
- Utilisation Facile
- Simple de l’adapter à nos besoins

Exemple android avec Volley

Accéder à n'importe quel web



Ajouter la librairie au gradle et permission

gradle

```
implementation 'com.android.volley:volley:1.0.0'
```

manifest

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

Accès en utilisant volley

```
private lateinit var textView: TextView
private var requestQueue: RequestQueue? = null
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    //title = "KotlinVolley"
    textView = findViewById(R.id.textViewResult)
    val button: Button = findViewById(R.id.btnParse)
    requestQueue = Volley.newRequestQueue(this)
    button.setOnClickListener {
        jsonParse()
    }
}
```

Accès en utilisant volley

```
private fun jsonParse() {
    val url = "https://jsonplaceholder.typicode.com/users"
    val request = JsonArrayRequest(Request.Method.GET, url, null, {        //définir la reqûete http

            response ->try {
      //Si array avec nom dans le fichier json   val jsonArray = response.getJSONArray(« monTableau")

        for (i in 0 until response.length()) {          //Parcourir le JsonArray réponse (response)
            val user = response.getJSONObject(i)      //récupérer le JsonObject de position i du jsonArray response
            val firstName = user.getString("name")
            val username = user.getString("username")
            val mail = user.getString("email")
            textView.append("$firstName, $username, $mail")
        }
    } catch (e: JSONException) {
        e.printStackTrace()
    }
    }, { error -> error.printStackTrace() })
    requestQueue?.add(request)              //ajouter notre reqête à la file des requête (requestQueue)
}
```

END