

# Accéder à une API

Retrofit2 + RxJava

# Retrofit2 ?

Librairie java qui permet de générer un client REST pour consommer une API

- Utilisation d'URL paramétrées
- Header personnalisés
- Téléchargement / Upload de fichier
- ...

# RxJava?

Librairie java basée sur le pattern *Observable* qui permet de faire des appels asynchrones facilement et proprement

- Le résultat d'un appel asynchrone peut être vu comme un évènement
- L'*observer* peut être notifié du résultat

# Installation dans un projet Android

Dans le fichier gradle (app) => Import de la librairie Retrofit + convertisseur de requête + adapter RxJava

```
compile 'com.squareup.retrofit2:retrofit:2.1.0'  
compile 'com.squareup.retrofit2:converter-gson:2.1.0'  
compile 'io.reactivex.rxjava2:rxjava:2.0.1'  
compile 'io.reactivex.rxjava2:rxandroid:2.0.1'  
compile 'com.jakewharton.retrofit:retrofit2-rxjava2-adapter:1.0.0'
```

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

# Interface de communication

Créer une interface contenant les requêtes à utiliser:

```
public interface EventCityApiService {  
  
    @GET("api/v1/cities?populate=events")  
    Observable<EventCityResponse<City>> getCities();  
  
}
```

# Builder

```
EventCityApiService apiService = new Retrofit.Builder()  
    .baseUrl("http://192.168.0.15:3000/")  
    .addCallAdapterFactory(RxJava2CallAdapterFactory.create())  
    .addConverterFactory(GsonConverterFactory.create())  
    .build().create(EventCityApiService.class);
```

# Appel asynchrone à l'API

```
apiService.getCities()
```

```
.observeOn(AndroidSchedulers.mainThread())
```

```
.subscribeOn(Schedulers.io())
```

```
.subscribe(this::handleResponse, this::handleError);
```

# Gestion de la réponse

On crée le RecyclerView et on alimente l'adapter

```
private void handleResponse(EventCityResponse<City> response) {  
    cities = response.getData();  
    RecyclerView rv = (RecyclerView) findViewById(R.id. rvCities);  
    rv.setLayoutManager( new LinearLayoutManager( this));  
    rv.setAdapter( new CityAdapter( this, cities, new CityAdapter.OnCityClickListener() {  
        @Override  
        public void onCityClick(City city) {  
            Intent intent = new Intent(getApplicationContext(), CityActivity. class);  
            [...]  
            startActivityForResult(intent, 1);  
        }  
    }));  
}
```



# Gestion des erreurs

```
private void handleError(Throwable error) {  
  
    Toast.makeText(this, "Error "+error.getLocalizedMessage(), Toast.LENGTH_SHORT).show();  
  
}
```