

REST sous Android

Atelier T1.A09 @ JDEV 2017

Olivier Le Goaër
olivier.legoer@univ-pau.fr



Prérequis de l'atelier

- ✓ IDE installé, configuré et maîtrisé
 - Eclipse ADT ou Android Studio
 - Android 5.1 (API Level 22) ou supérieur
- ✓ Fondamentaux du SDK Android
 - Activity, ressources (R.java), manifeste, ...
- ✓ Notation JSON (et son parsing)
- ✓ Accès internet



Pointeurs utiles pour cet atelier

- ✓ Cours de programmation Android
 - <http://olegoaer.developpez.com/cours/mobile/>
- ✓ Tutoriel RPC Android
 - <http://olegoaer.developpez.com/tutos/mobile/android/rpc/>
- ✓ Les Web Services en 60 diapos chrono
 - <http://bit.ly/2oXHJ8r>
- ✓ The *genDROID* project
 - <http://gendroid.univ-pau.fr/>



RPC Android, aka "Web services"

✓ Contexte : *Remote Procedure Call (RPC)*

- Le client Android peut appeler*, de façon technologiquement neutre, toute sorte de routines exécutées coté serveur



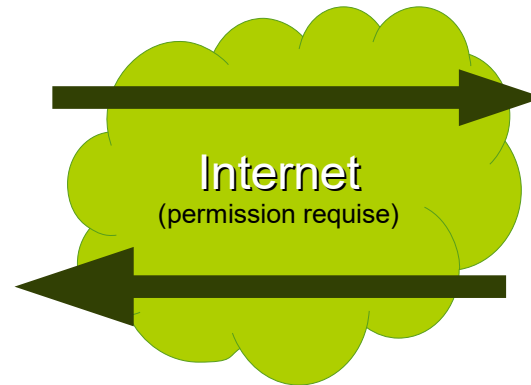
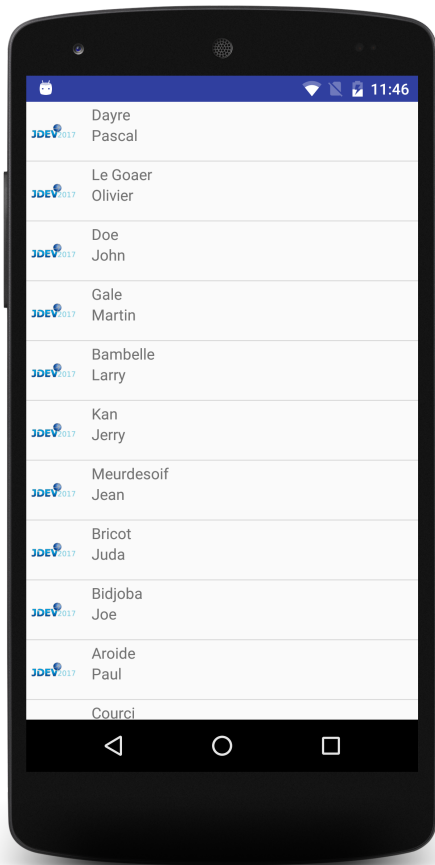


RPC Android de type REST

- ✓ Technologie RPC REST (versus SOAP)
 - Objet client http unique (interface serveur unifiée)
 - URIs + verbes d'action + codes serveur
 - JSON-RPC : les échanges se font en JSON
- ✓ Open Data REST API
 - L'interaction avec une BDD distante (`INSERT`, `SELECT`, ...) doit se faire par ce biais
 - La technologie BDD utilisée coté serveur est neutre (Oracle, MySQL, etc.)
 - Certaines BDD de type NoSQL exposent nativement une telle API REST (ex: [CouchDB](#))



Case Study : participants JDEV



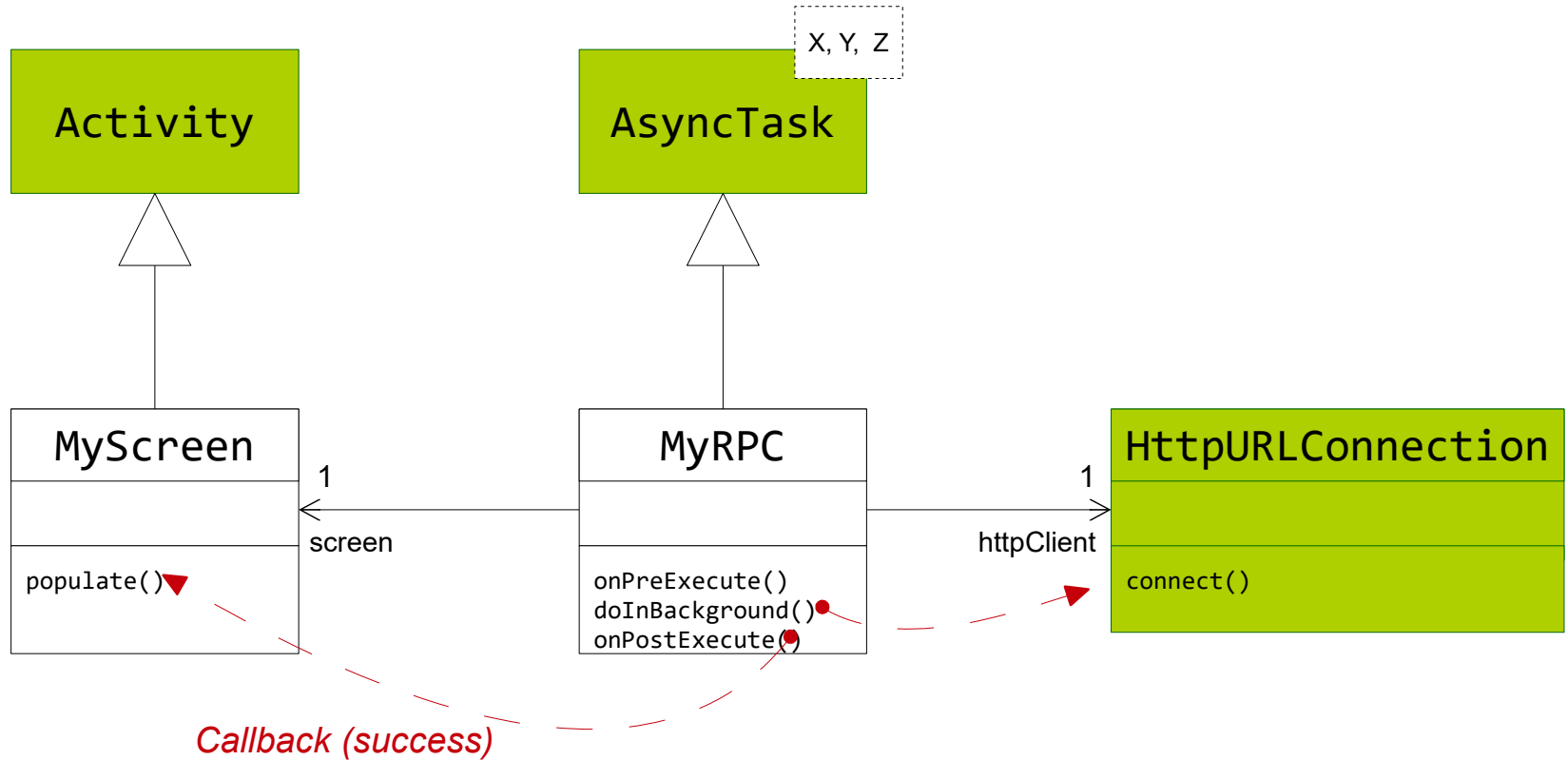
```
{ "hits": 19, "people":  
  [ { "nom": "DAYRE", "prenom": "Pascal" },  
    { "nom": "LE GOAER", "prenom": "Olivier" },  
    { "nom": "DOE", "prenom": "John" },  
    { "nom": "GALE", "prenom": "Martin" },  
    { "nom": "BAMBELLE", "prenom": "Larry" },  
    { "nom": "KAN", "prenom": "Jerry" },  
    { "nom": "MEURDESOSIF", "prenom": "Jean" },  
    { "nom": "BRICOT", "prenom": "Juda" },  
    { "nom": "BIDJOBA", "prenom": "Joe" },  
    { "nom": "AROIDE", "prenom": "Paul" },  
    { "nom": "COURCI", "prenom": "Sarah" },  
    { "nom": "PROVISTE", "prenom": "Alain" },  
    { "nom": "STRUEUX", "prenom": "Simon" },  
    { "nom": "COVERT", "prenom": "Harry" },  
    { "nom": "TERGEIST", "prenom": "Paul" },  
    { "nom": "PEUPLUS", "prenom": "Jean" },  
    { "nom": "BONNO", "prenom": "Jean" },  
    { "nom": "GONZOLA", "prenom": "Igor" },  
    { "nom": "REMORD", "prenom": "Yves" } ] }
```

<http://www.iut-adouretud.univ-pau.fr/~olegoaer/webservices/jdev2017.php?transform=maj>

Paramètre optionnel



Design pattern RPC REST Android

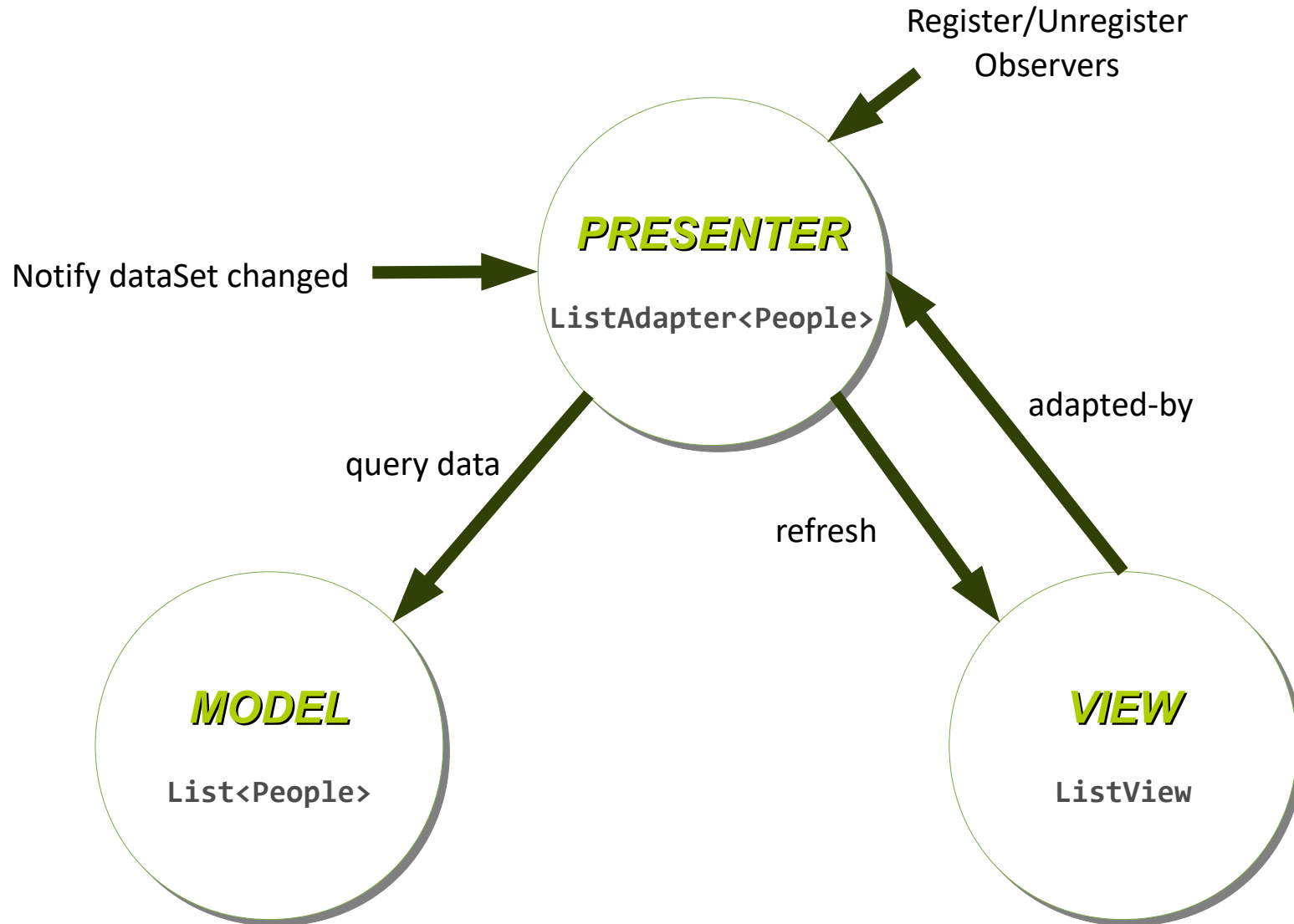


Solution 100% native SDK Android

Librairies tierces : Volley, OkHttp, Retrofit, LoopJ, Ion...



Model-View-Presenter





The *gen*DROID Project

- ✓ On recommence l'atelier, mais cette fois-ci avec les plaisanteries sur Chuck Norris
 - <http://api.icndb.com/jokes/random/60>
 - Retourne 60 plaisanteries aléatoirement
- ✓ On ne se fatigue plus, et on utilise le générateur automatique de code
 - <http://gendroid.univ-pau.fr/remoteDatabase.html>
 - 3 classes Java + un fichier xml à intégrer dans votre projet Android Studio
 - Ajustez votre manifest. Compilez le tout. Enjoy !