

# DÉVELOPPEMENT D'APPLICATIONS MOBILES

Alex Morel - iRéalité



@alex\_morel\_



## Programme du cours :

1. Bases Android
2. Intents, HTTP, Asynchronicité, Évènements
3. Retrofit, Google Maps
4. Base de données mobile, Notifications, Bonnes pratiques
5. Encadrement TP

MOB  
ILE  
APPS

PREVIOUSLY ON

*Développement d'applications mobiles*  
*Cours 2*



**Intents** : lier les écrans de mon app (avec params/result), and more



**Git - niveau 2** : les Remotes (git fetch, rebase, clone, push)



**HTTP** : Serveurs REST, OKHttp



**Asynchronicité** : Thread UI gardé, AsyncTasks



**Évènements** : pattern EventBus, Otto sur Android

## **Nos objectifs du jour :**

1. Retrofit : make HTTP great again
2. Git - niveau 3 : travail collaboratif
3. Google Map sur Android
4. Base de données mobile

MOB  
ILE  
APPS



## PlaceSearcher : qu'est-ce que vaut notre code ?

Ca fonctionne mais...

- OkHTTP c'est pratique mais encore beaucoup de chose à gérer à la main
- Traitement de tous les cas d'erreurs (errorCode, body null, réponse au mauvais format...)
- Parser le JSON à la main c'est pénible, error-prone, peu maintenable
- Écrire des AsyncTask pour chaque requête va être fastidieux

## 1 - Retrofit





## 1 - Retrofit

Retrofit c'est :

- Une surcouche basée sur OkHTTP créée spécialement pour les Serveurs REST
- Parsing automatique de la réponse vers des objets Java
- Syntaxe ultra-facilitée pour décrire des APIs REST (GET, POST, DELETE, paramètres...)
- Génère tout le code, vous n'avez plus qu'à traiter les cas d'erreurs et utiliser le résultat





## 1 - Retrofit

```
// Create Retrofit client
Retrofit retrofit = new Retrofit.Builder()
    .client(new OkHttpClient())
    .baseUrl("https://api-adresse.data.gouv.fr")
    .addConverterFactory(GsonConverterFactory.create(gsonConverter))
    .build();
```

Pour initialiser Retrofit il nous faut :

- Un client HTTP (e.g. OkHTTP)
- Une URL de serveur de base (e.g. notre service de geoloc)
- Un convertisseur JSON -> Java (e.g. GSON)



# 1 - Retrofit

- Un convertisseur JSON -> Java (e.g. GSON)

```
// Create GSON Converter that will be used to convert from JSON to Java
Gson gsonConverter = new GsonBuilder()
    .excludeFieldsWithModifiers(Modifier.FINAL, Modifier.TRANSIENT, Modifier.STATIC)
    .serializeNulls()
    .excludeFieldsWithoutExposeAnnotation().create();
```

- A partir d'une classe Java, conversion/parsing JSON
- Possibilité d'ajouter des règles (uniquement les @Expose, conversion custom de Date...)

```
public class PlaceSearchResult {
    @Expose
    public List<PlaceAddress> features;
}
```



```
{
  licence: "ODbL 1.0",
  - features: [
    - {
      type: "Feature",
      - geometry: {
        type: "Point",
        - coordinates: [
          2.292859,
          48.845316
        ]
      },
      - properties: {
        y: "GB60818",
        id: "75115 XXXX a00a13",
      }
    }
  ]
}
```



## 1 - Retrofit

A partir d'une Interface décrivant les routes du serveur (GET, POST, DELETE...)

```
public interface PlaceSearchRESTService {  
    @GET("search/")  
    Call<PlaceSearchResult> searchForPlaces(@Query("q") String search);  
}
```

Et des modèles JAVA correspondants

```
public class PlaceSearchResult {  
    @Expose  
    public List<PlaceAddress> features;  
}
```

Retrofit génère tout le code pour nous !

```
// Use retrofit to generate our service  
mPlaceSearchRESTService = retrofit.create(PlaceSearchRESTService.class);
```



# 1 - Retrofit

```
public interface PlaceSearchRESTService {  
    @GET("search/")  
    Call<PlaceSearchResult> searchForPlaces(@Query("q") String search);  
}
```

- Appels réseau (timeouts, problèmes de connexion...)
- Parsing du JSON, conversion en Java, type-checking
- Asynchronicité (plus besoin de créer d'AsyncTask)

```
mPlaceSearchRESTService.searchForPlaces(search).enqueue(new Callback<PlaceSearchResult>() {  
    @Override  
    public void onResponse(Call<PlaceSearchResult> call, retrofit2.Response<PlaceSearchResult> response) {  
        // Request has succeeded and we directly have the result as java  
        PlaceSearchResult result = response.body();  
    }  
  
    @Override  
    public void onFailure(Call<PlaceSearchResult> call, Throwable t) {  
        // Request has failed or is not at expected format  
    }  
});
```



## 1 - Retrofit

### Exercice n° 14

- Réécrire le PlaceSearchService à l'aide de Retrofit

*Le sujet donne volontairement peu d'indices, vous êtes officiellement des développeurs Android. Il est normal que plusieurs parties de votre code doivent être modifiées car vous devriez être amenés à changer le modèle de données. N'hésitez surtout pas à demander l'aide du professeur.*

## 2 - Git - niveau 3

MOB  
ILE  
APPS





## 2 - Git - niveau 3

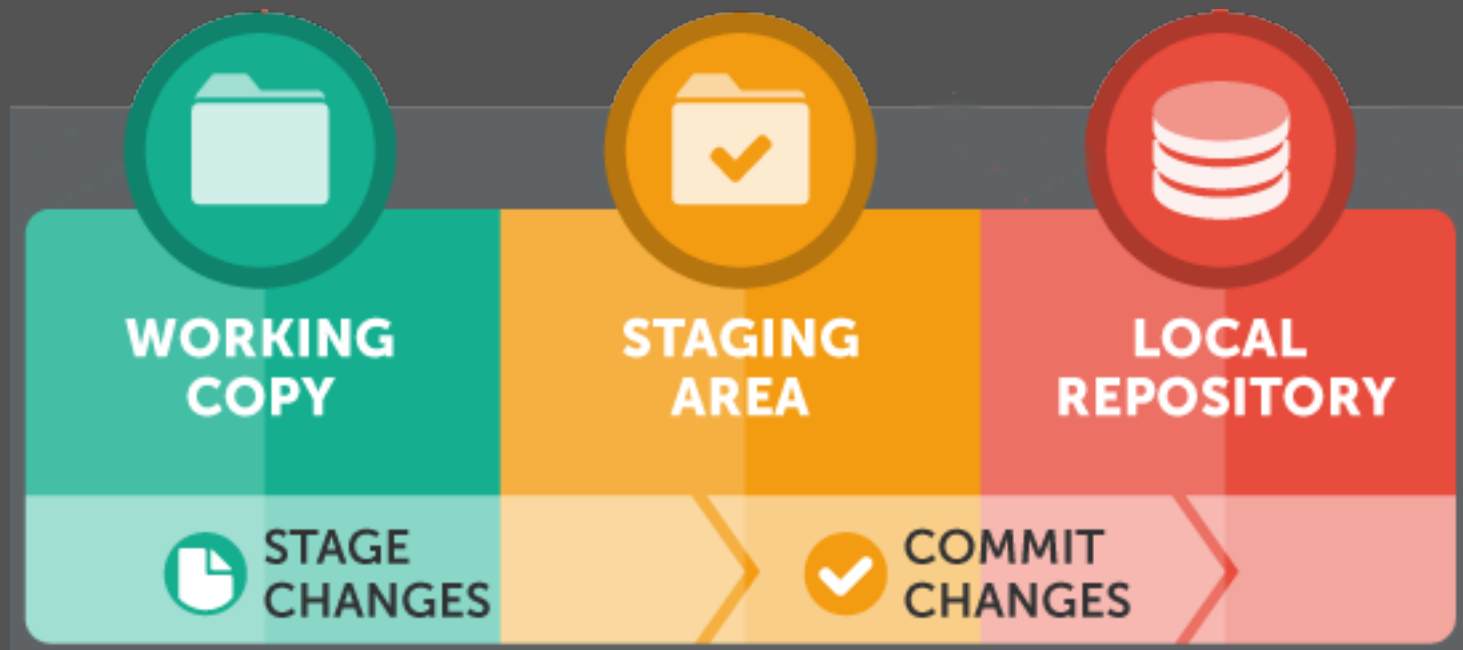
NE TRAVAILLEZ **JAMAIS** SANS REPO GIT

COMMITEZ **PETIT** POUR LES CONFLITS

COMMITEZ **TOUT** LE TEMPS



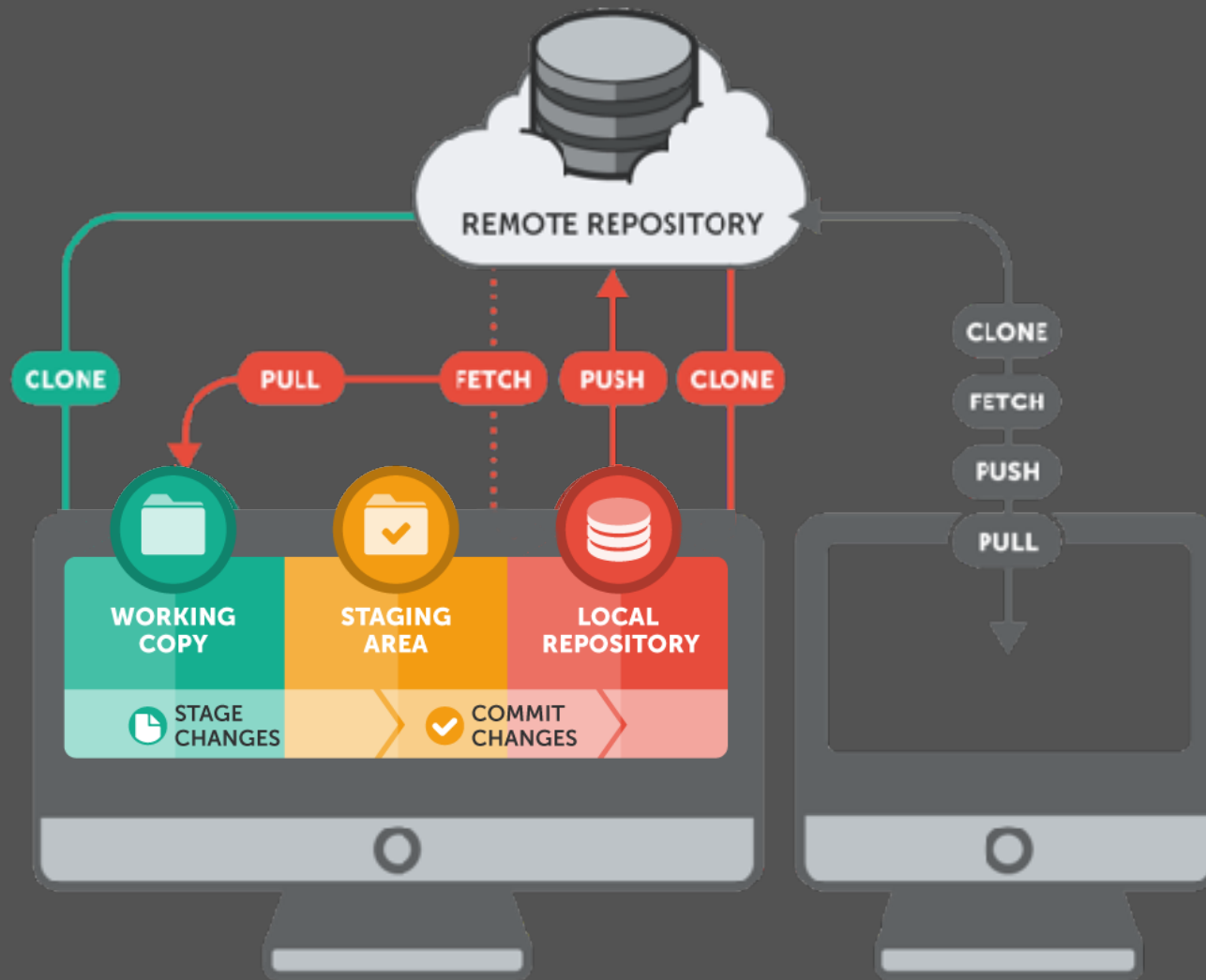
## 2 - Git - niveau 3







## 2 - Git - niveau 3





## 2 - Git - niveau 3

`git fetch team`

- Mets à jour le pointeur local team/master
- Ne change pas votre historique local (master)

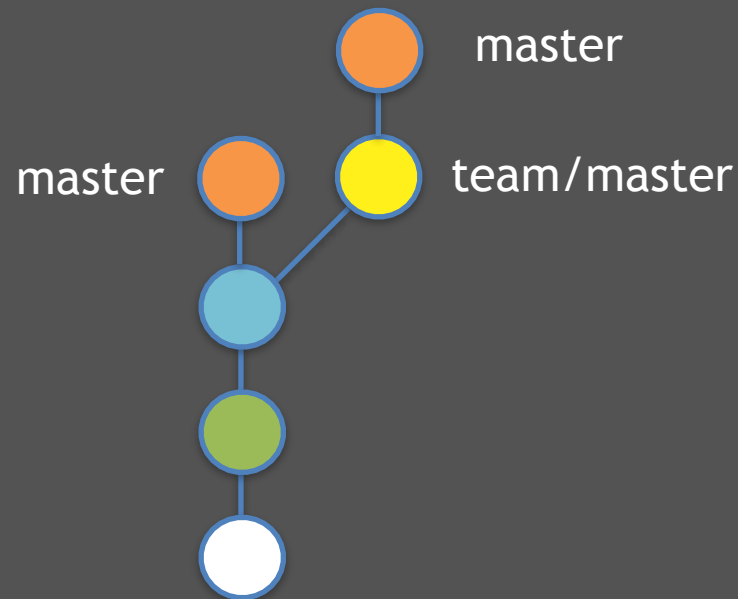




## 2 - Git - niveau 3

```
git rebase team/master master
```

- Applique tous les comits de master en partant de team/master
- Peut générer des conflits



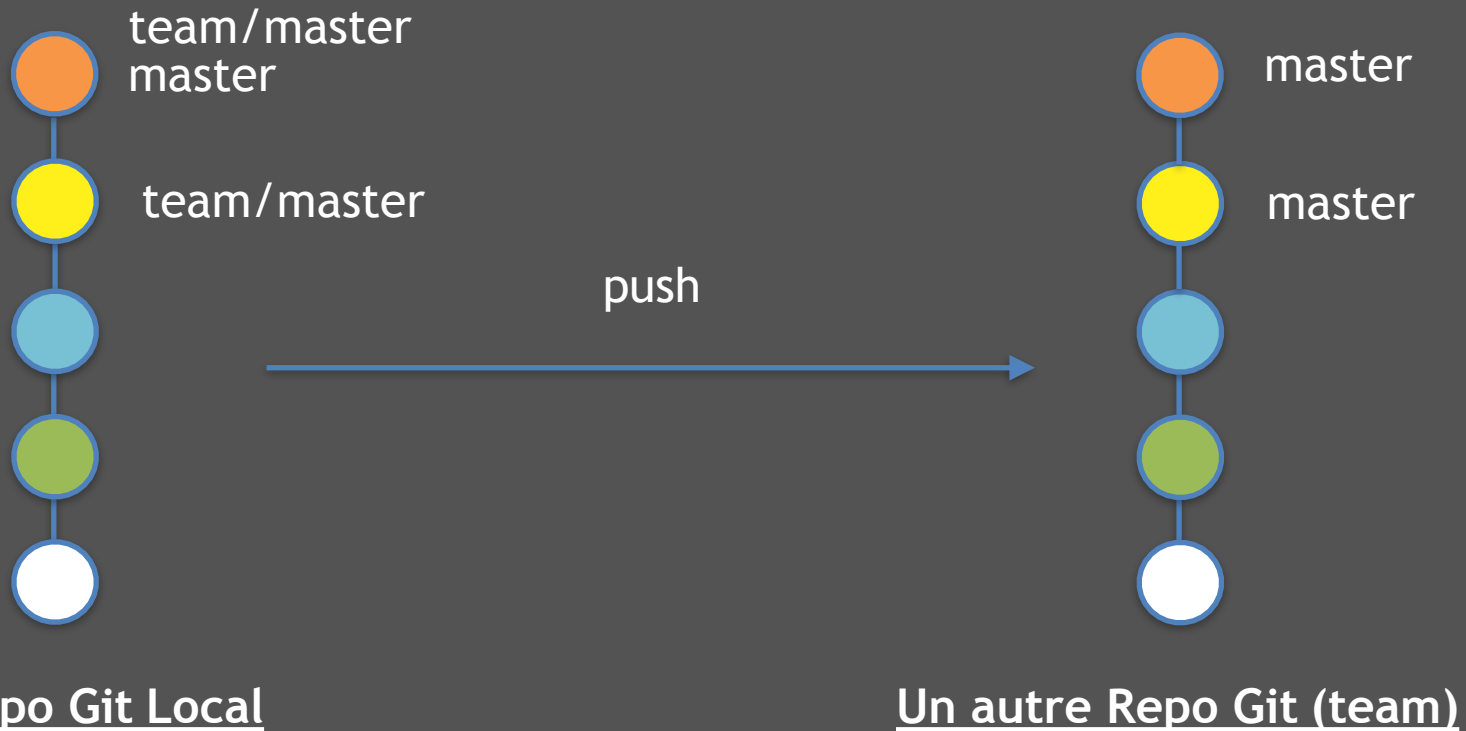
Mon Repo Git Local



## 2 - Git - niveau 3

`git push team master`

- Cherche l'ancêtre commun (ici **jaune**)
- **Les changements locaux doivent être au-dessus de team**





## 2 - Git - niveau 3

Bitbucket : serveur git pour repositories publics & privés

Alex Morel / org.miage.placesearcher

### Overview



SSH ▾

git@bitbucket.org:alexmorel/org.miage



Bitbucket

Last updated 2018-01-31

Access level Admin

0

Open PRs

1

Watcher

2

Branches

1

Fork



## 2 - Git - niveau 3

Bitbucket : serveur git pour repositories publics & privés

- Possibilité de « forker » un repository : crée une copie à vous
- Vous êtes libres sur votre fork : casser l'historique, push -f
- **ON NE PUSHE JAMAIS SUR TEAM !**
- Une fois vos changements prêts, possible de créer des Pull Request




## 2 - Git - niveau 3


### Bitbucket : créer une pull request


Alex Morel / org.miage.placesearcher.fork / Pull requests

### Create a pull request


 alexmorel / org.miage.placesearcher...  
Created 2010-01-09, Updated yesterday

→



 alexmorel / org.miage.placesearcher...  
Created 2010-01-09, Updated yesterday

 correction

→

 master

Title <sup>\*</sup> My Awesome pull request

Description <sup>\*</sup>   Preview

A magnificent pull request

- \* [Android] Exercise 14 : refactor PlaceService using Retrofit
- \* [Android] Exercise 15 - Step 1 : basic Google Maps Wiring
- \* [Android] Exercise 16 : Step 2 : basic Google Maps Wiring

Reviewers Start typing to search for a user

Close branch ☐ Close correction after the pull request is merged

Create pull request



## 2 - Git - niveau 3

### Bitbucket : revue de la Pull Request par l'équipe

My Awesome pull request

#1 **OPEN** [arg.niags.place](#) [correction](#) → [master](#) [Merge](#) [Edit](#) [Decline](#) [Approve](#) 0

Overview Commits Activity

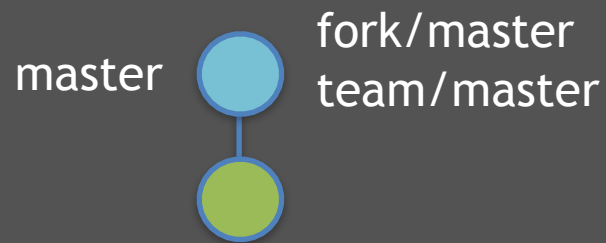
Author	Commit	Message	Date	Builds
Alex Morel	<a href="#">5e68011</a>	[Android] Exercice 17 - Step 3 : PlaceSearchService Update : first search locally, save results in DB	yesterday	
Alex Morel	<a href="#">66127bc</a>	[Android] Exercice 17 - Step 2 : add ORM annotations	yesterday	
Alex Morel	<a href="#">f1a0868</a>	[Android] Exercice 17 - Step 1: add ActiveAndroid to build and configure DB	yesterday	
Alex Morel	<a href="#">23247d4</a>	[Android] Exercice 16 - Step 6 : performance improvement : the right way	2018-01-26	
Alex Morel	<a href="#">4ba54fa</a>	[Android] Exercice 16 - Step 5 : performance improvement : naive way	2018-01-25	

- Possible de voir le détail de chaque commit de la Pull Request
- Possible d'accepter / commenter / éditer / refuser la pull
- Une fois la pull corrigée puis accepté, le code est intégré sur le repo cible





## 2 - Git - niveau 3



Mon Repo Git Local



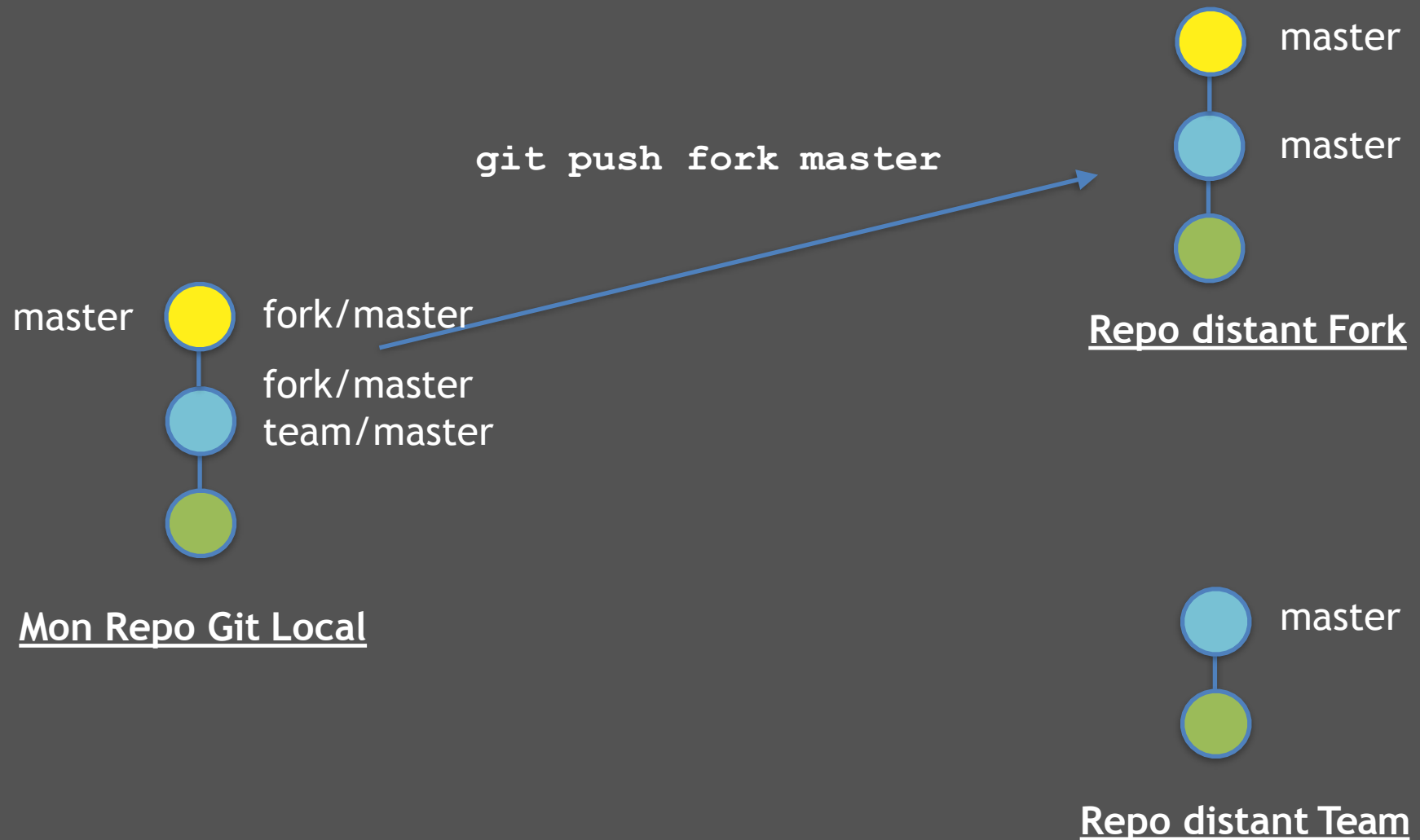
Repo distant Fork



Repo distant Team



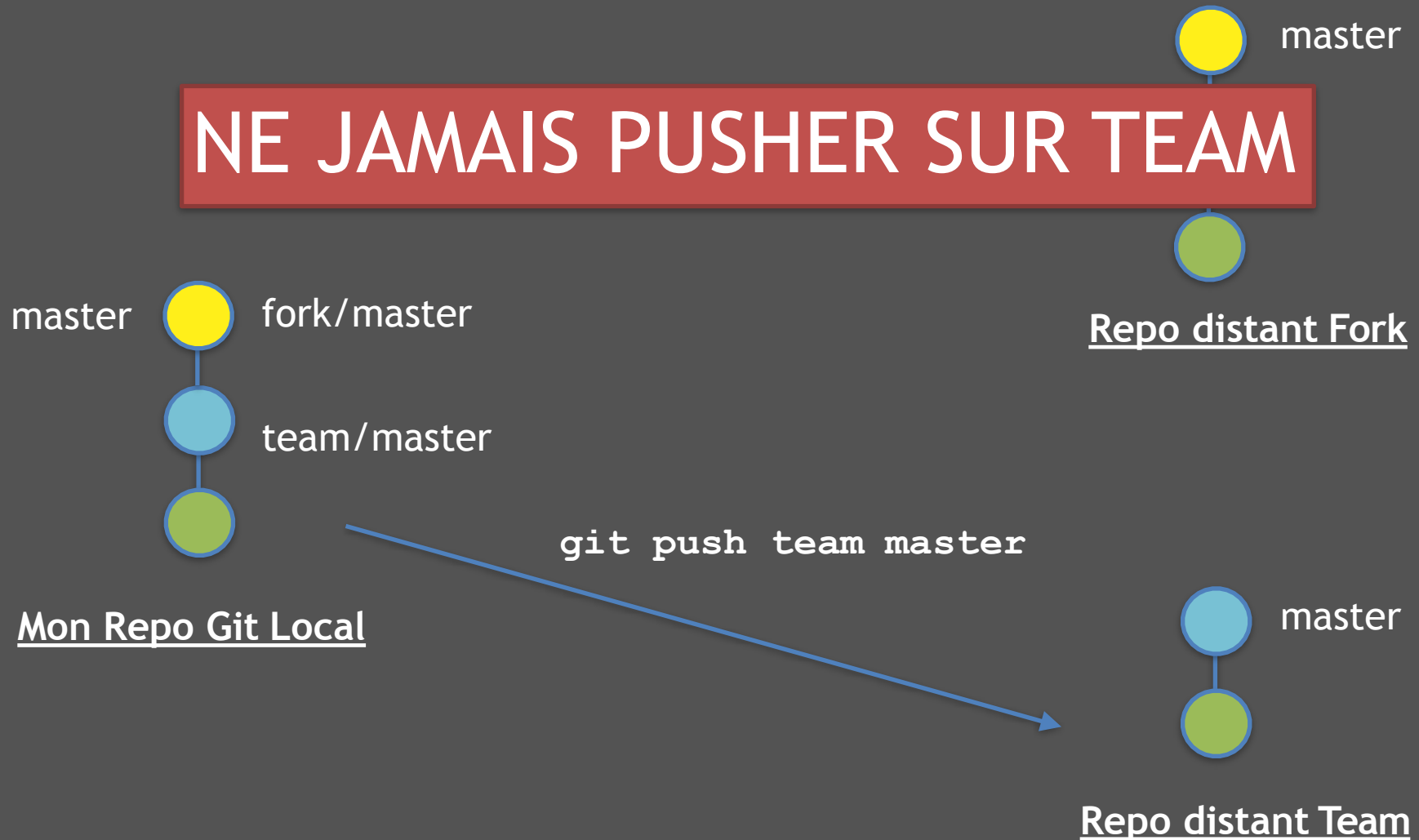
## 2 - Git - niveau 3





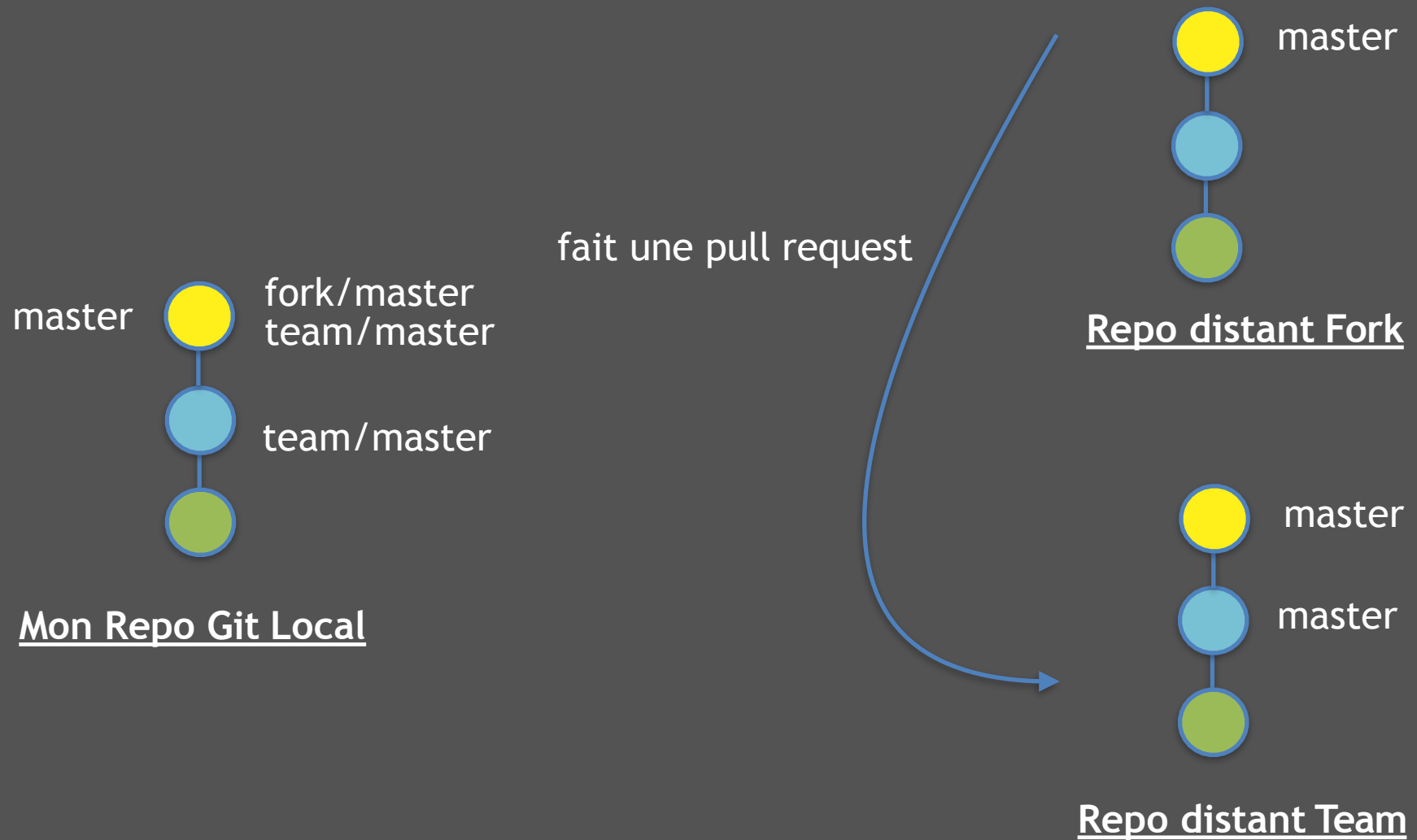
## 2 - Git - niveau 3

**NE JAMAIS PUSHER SUR TEAM**





## 2 - Git - niveau 3



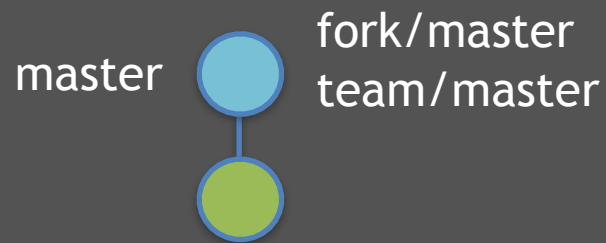
# LES 3 CHOSES À GARDER EN TÊTE

1. Je fais mes modifs sur mon repo local
2. Je pushe sur MON fork
3. Je fait des pull request pour intégrer sur team

NOU  
VELLES  
INTER  
FACES



## 2 - Git - niveau 3



Mon Repo Git Local



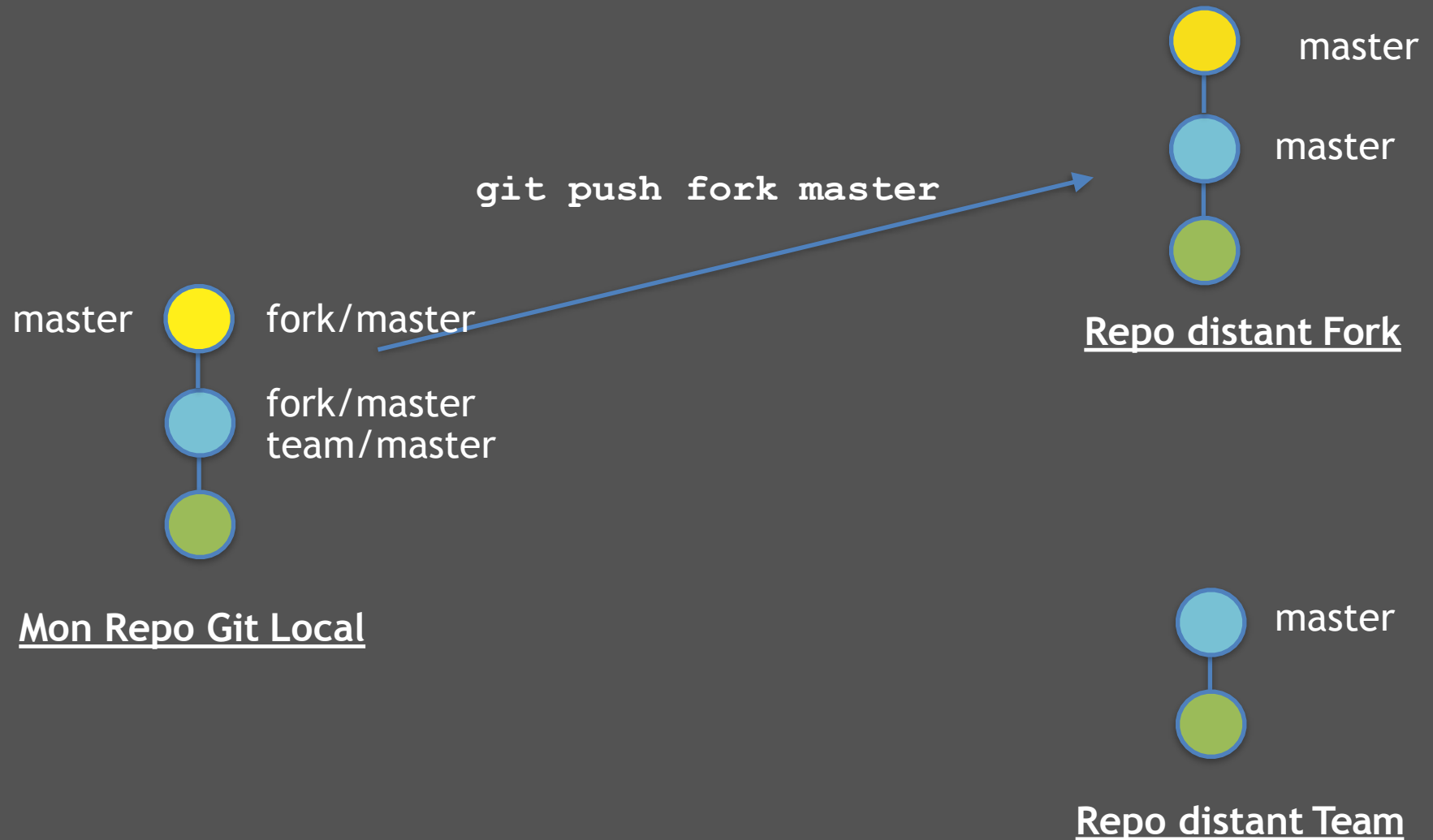
Repo distant Fork



Repo distant Team

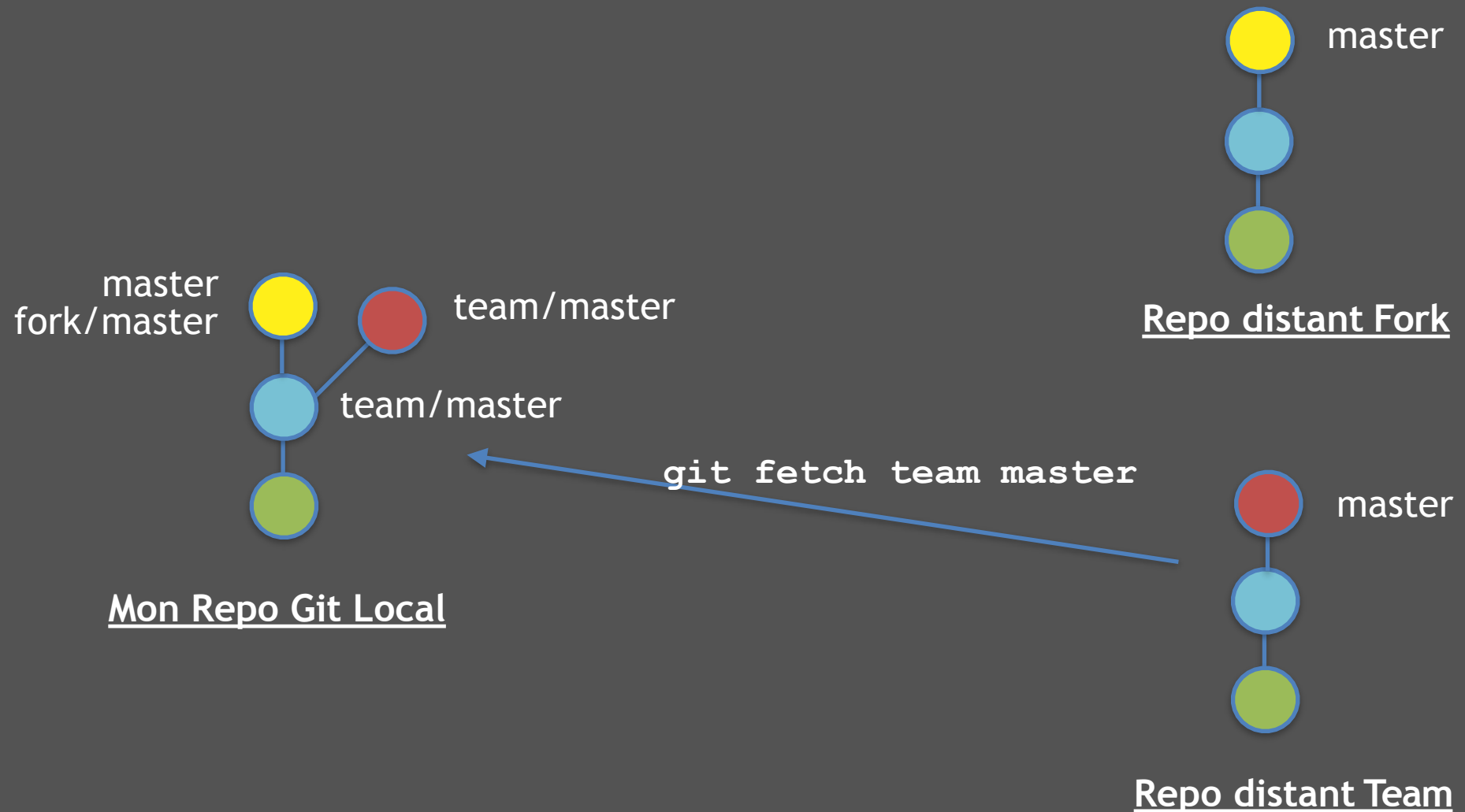


## 2 - Git - niveau 3





## 2 - Git - niveau 3



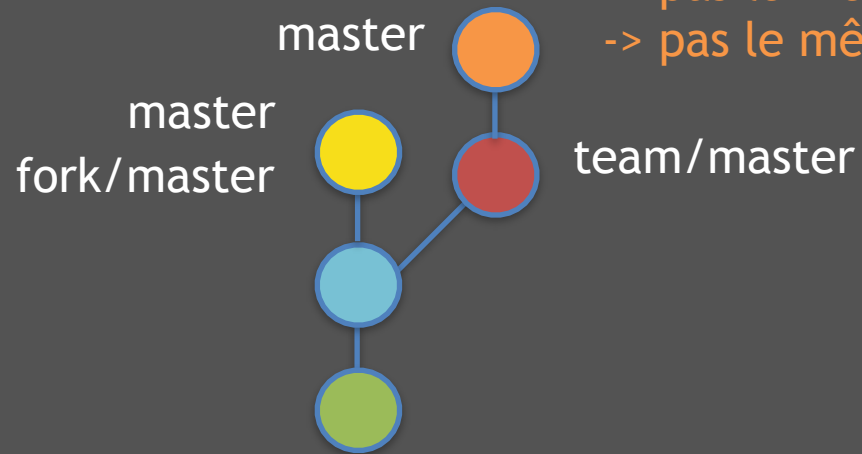




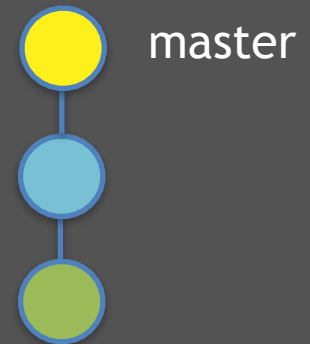
## 2 - Git - niveau 3

```
git rebase team/master master
```

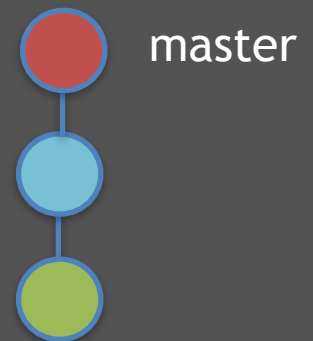
Pas le même parent  
-> pas le même sha1  
-> pas le même commit



Mon Repo Git Local



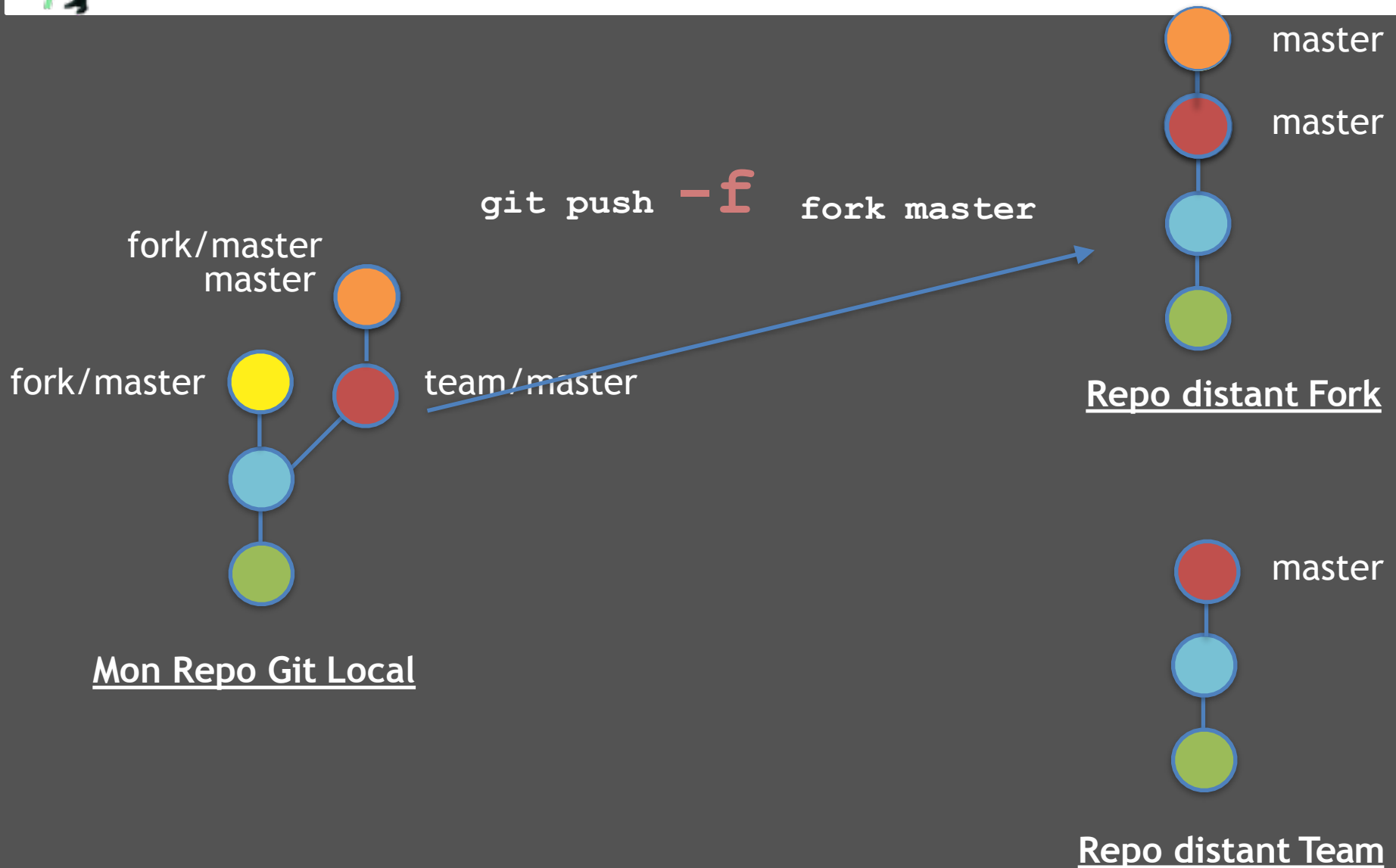
Repo distant Fork



Repo distant Team



## 2 - Git - niveau 3





## 2 - Git - niveau 3

team/master  
master  
fork/master



team/master

Mon Repo Git Local

Fait une pull request



master

Repo distant Fork



master



master



Repo distant Team

# LES 3 CHOSES À GARDER EN TÊTE

1. Je fais mes modifs sur mon repo local
2. Je pushe sur MON fork (et j'ai le droit de -f)
3. Je fetch/rebase team PUIS fait pull request pour intégrer

NOU  
VELLES  
INTER  
FACES



## 2 - Git - niveau 3

### Exercice GIT

- Clonez le repository <https://bitbucket.org/alexmorel/org.miage.placesearcher>
- Inscrivez-vous sur Bitbucket et forcez ce repository
- Donnez les droits en lecture à l'utilisateur alexmorel
- Ajouter votre fork en tant que remote nommé (« mine »)
- Attendez que le professeur pushe le corrigé de l'exercice précédent
- Intégrez ce commit en local et sur votre fork

## Apparté : debugger vos apps



MOB  
ILE  
APPS

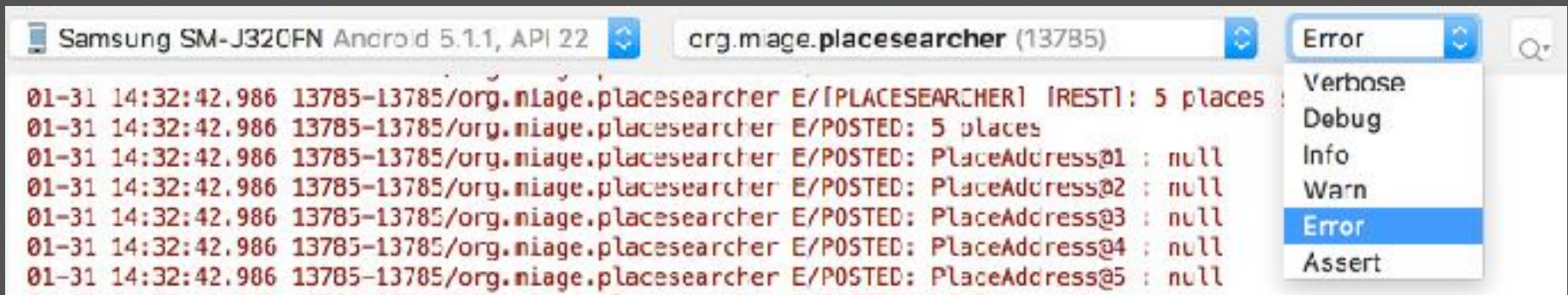


## Apparté : debugger vos apps

Les logs : niveau 0 du debug (mais peut être utile)  
Log.e(), Log.d(), Log.i()...

```
Log.d( tag: "[PLACESEARCHER] [REST]", msg: "This is a debug message");  
Log.e( tag: "[PLACESEARCHER] [REST]", msg: response.body().features.size() + " places saved");
```

Accessible depuis la vue « LogCat »  
Filtres par application et/ou type et/ou recherche expreg





## Apparté : debugger vos apps

Mieux : lancer votre application en mode debug



Lancer l'application en mode debug



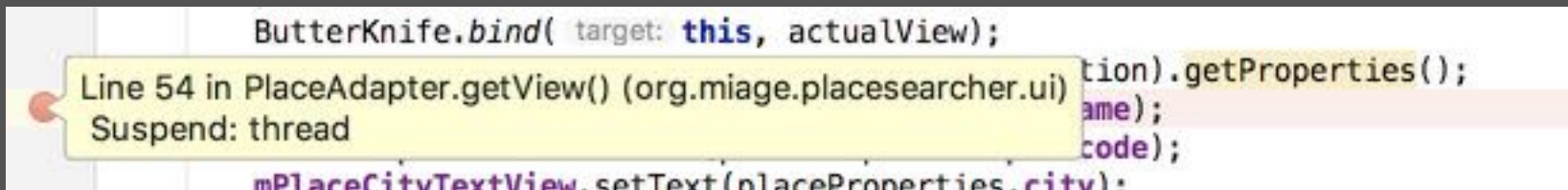
Attacher le debuggeur à une application déjà lancée



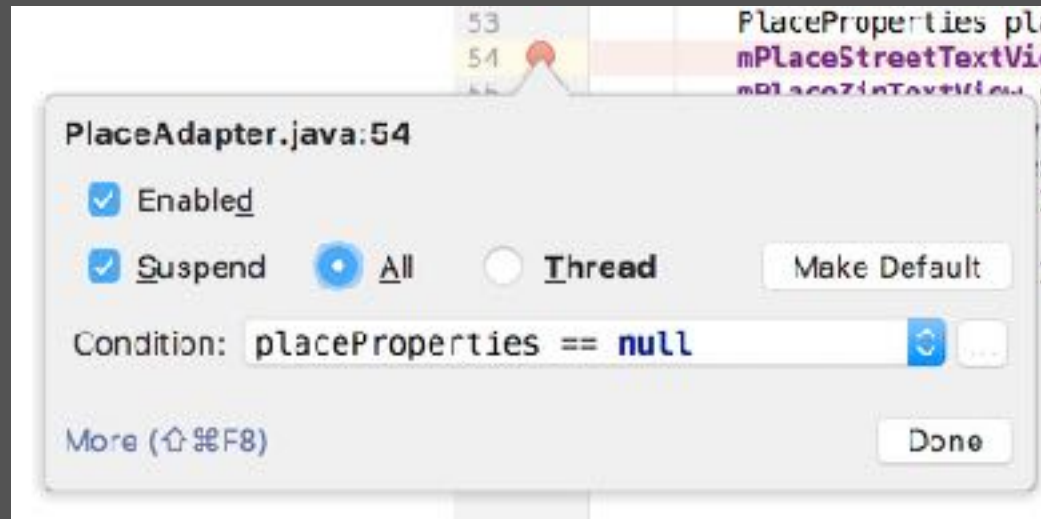


## Apparté : debugger vos apps

Les breakpoints (points d'arrêt) permettent de stopper l'application en mode debug



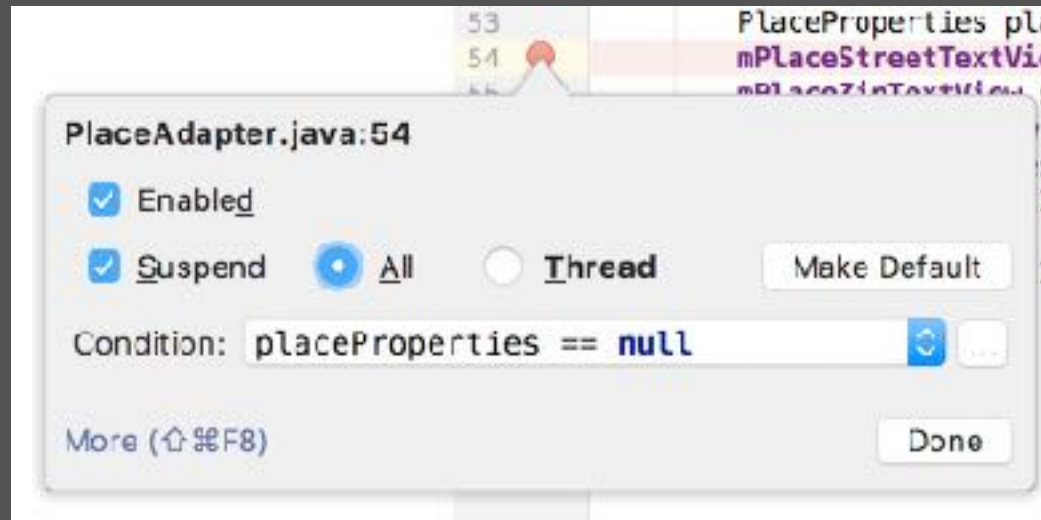
Possible de définir des conditions d'arrêts pour ne pas s'arrêter à chaque fois





## Apparté : debugger vos apps

Possible de définir des conditions d'arrêts pour ne pas s'arrêter à chaque fois



Vous pouvez choisir de suspendre le Thread concerné ou toute l'application (All)  
Attention cela n'a pas du tout le même impact



## Apparté : debugger vos apps

Une fois arrêtés sur un breakpoint, vous avez accès à des informations utiles

Pile d'appel (call stack) : pour voir qui a appelé le code

```
*main*@3 648 in group *main*: WAIT  
getView:54, PlaceAdapter (org.mlage.placesearcher.ui)  
obtainView:2828, AbsListView (android.widget)  
makeAndAddView:1932, ListView (android.widget)  
fillDown:715, ListView (android.widget)  
fillFromTop:781, ListView (android.widget)  
layoutChildren:1709, ListView (android.widget)  
onLayout:2626, AbsListView (android.widget)  
layout:16982, View (android.view)
```

Variables accessibles dans le contexte

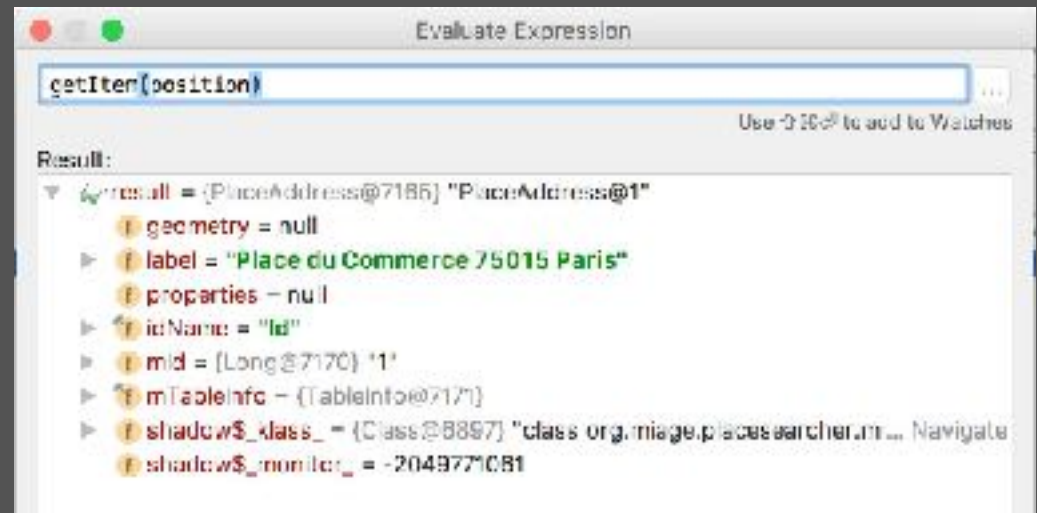
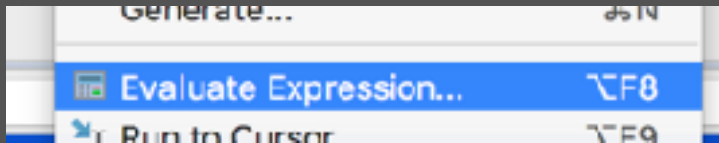
```
Debugger Console  
Variables  
this = (PlaceAdapter@7106)  
position = 0  
convertView = null  
parent = {ListView@7107} "android.widget.ListView[42875da VFED.VC. .... ID 16,127-704,578 #7107004d appid/listView]"  
actualView = {LinearLayout@7108} "android.widget.LinearLayout[19baca0b VFED.VC. .... ID 0,0-0,0]"  
placeProperties = null  
mPlaceStreetTextView = {AppCompatActivity@7110} "android.support.v7.widget.AppCompatActivity[199d9301 VFED.VC. .... ID 0,0-0,0]"  
placeProperties.name = java.lang.NullPointerException  
placeProperties.city = java.lang.NullPointerException  
mPlaceZipTextView = {AppCompatActivity@7111} "android.support.v7.widget.AppCompatActivity[199d9301 VFED.VC. .... ID 0,0-0,0]"
```



## Apparté : debugger vos apps

Une fois arrêtés sur un breakpoint, vous avez accès à des informations utiles

Besoin de plus ? Evaluate Expression

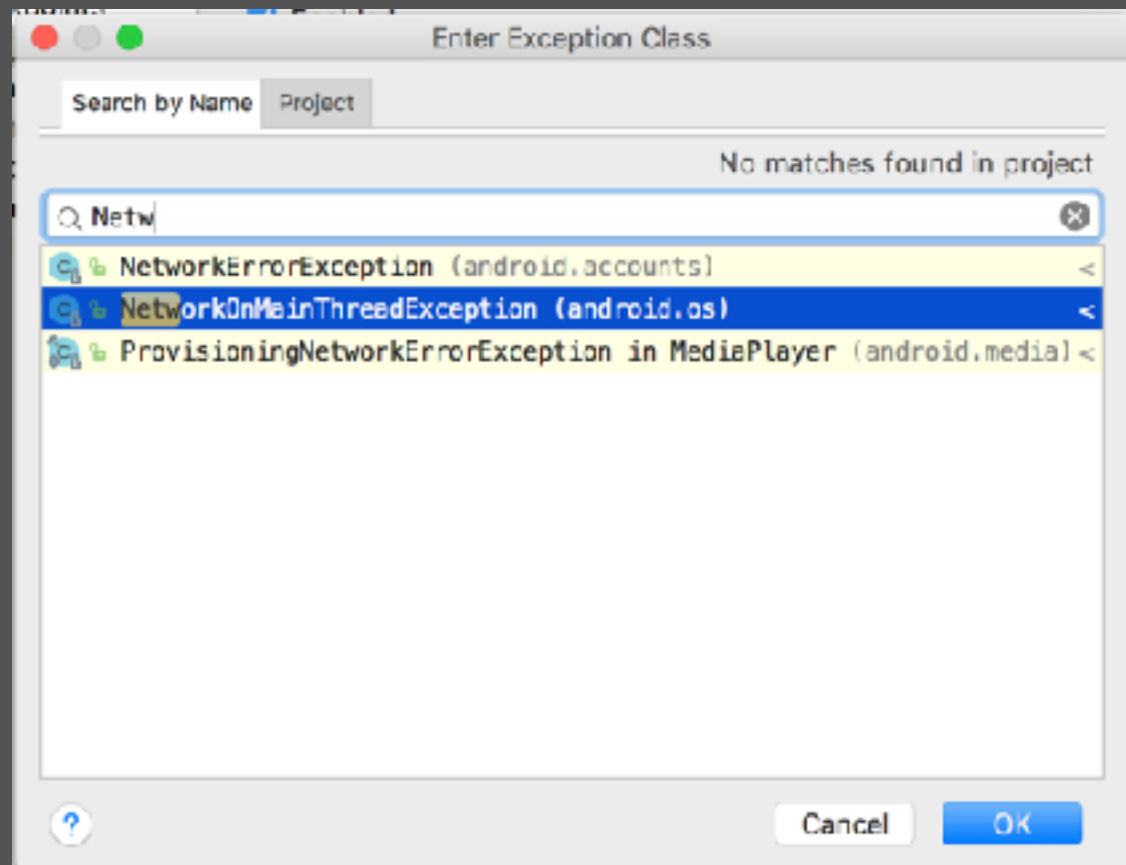
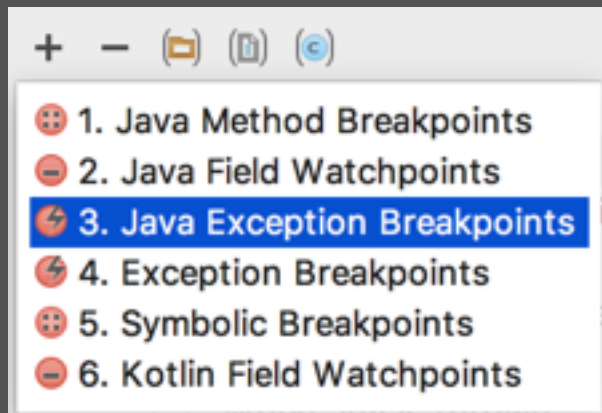


Vous pouvez ajouter une expression aux « Watches »  
Elle sera évaluée montrée dans la vue variable en permanence



## Apparté : debugger vos apps

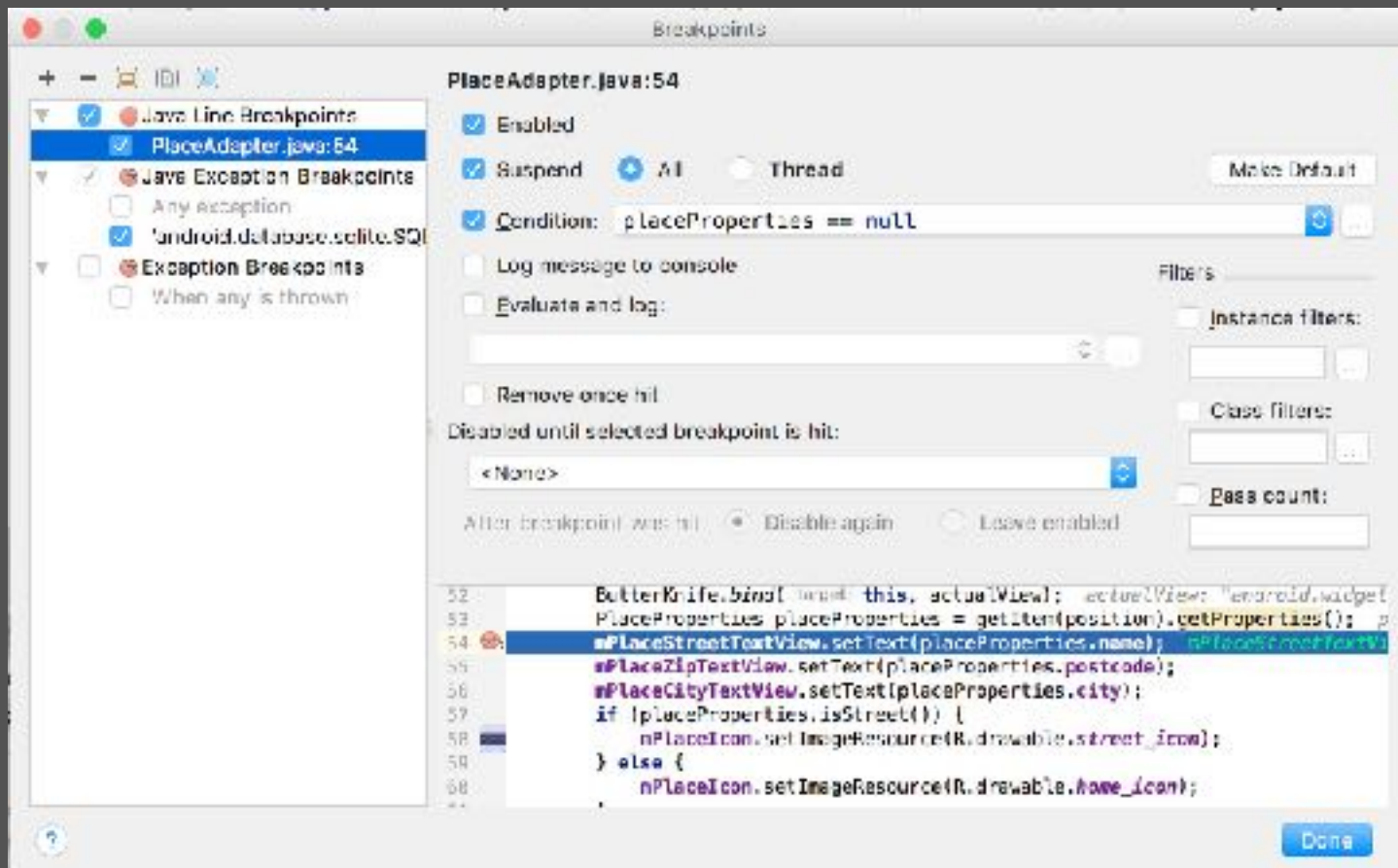
Possible de créer des breakpoints sur des exceptions données, sur le changement d'une variable...





## Apparté : debugger vos apps

Il est même possible de Logguer / exécuter du code / et bien plus sur chaque breakpoint





## Apparté : debugger vos apps

Quelques conseils/retours d'expérience :

- Savoir coder inclue savoir bien debugger
- Prenez le temps de vous former au debug, cela vous sera rendu au centuple
- Lire le code avant de lancer une session debug, c'est souvent suffisant pour trouver le bug
- N'appellez pas Log.e et Log.d pour debugger dans votre code mais dans des breakpoints
- Utilisez des Breakpoints d'exception pour comprendre les bugs mystiques
- Conservez vos breakpoints et désactivez-les, ils resserviront probablement dans quelques bugs

### 3 - Google Map sur Android

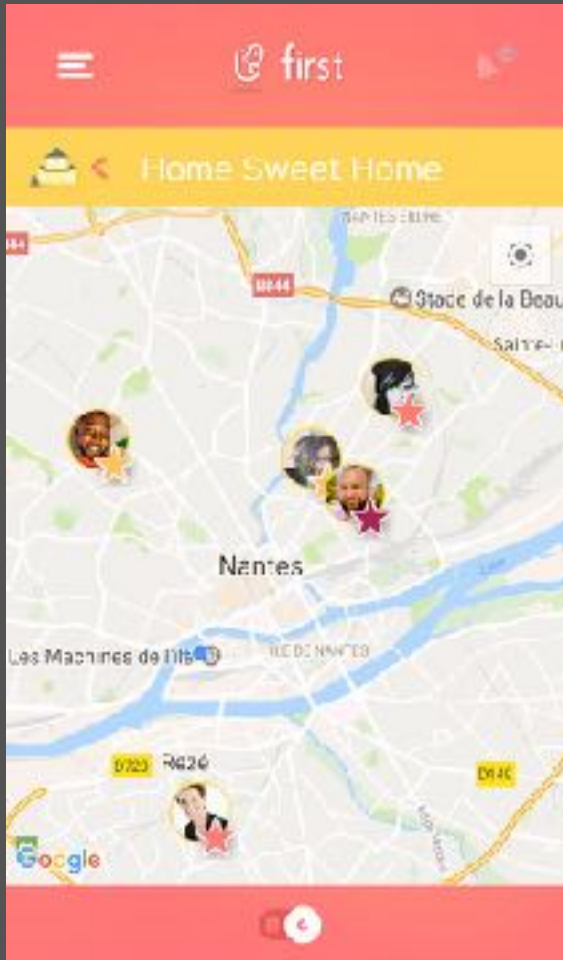


MOB  
ILE  
APPS





### 3 - Google Map sur Android



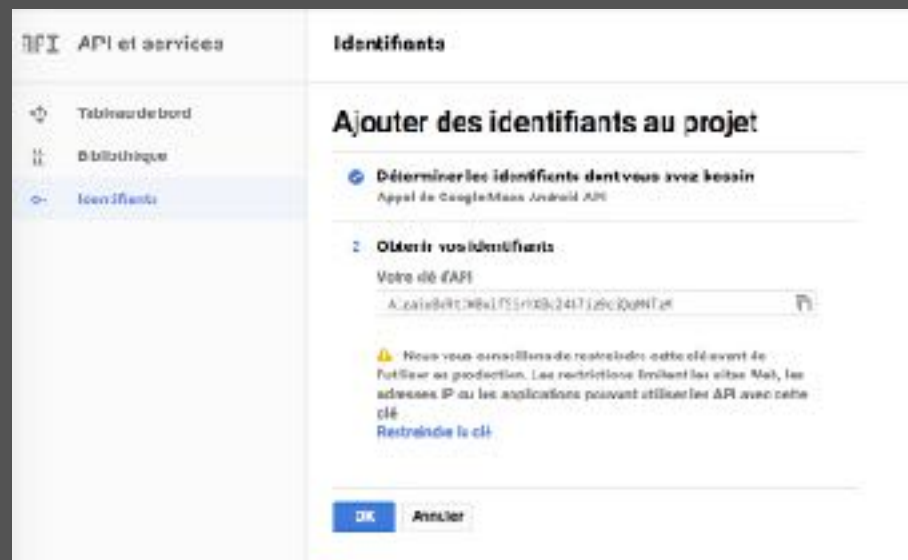
- Framework multi-plateforme pour afficher des cartes

Sur Android :

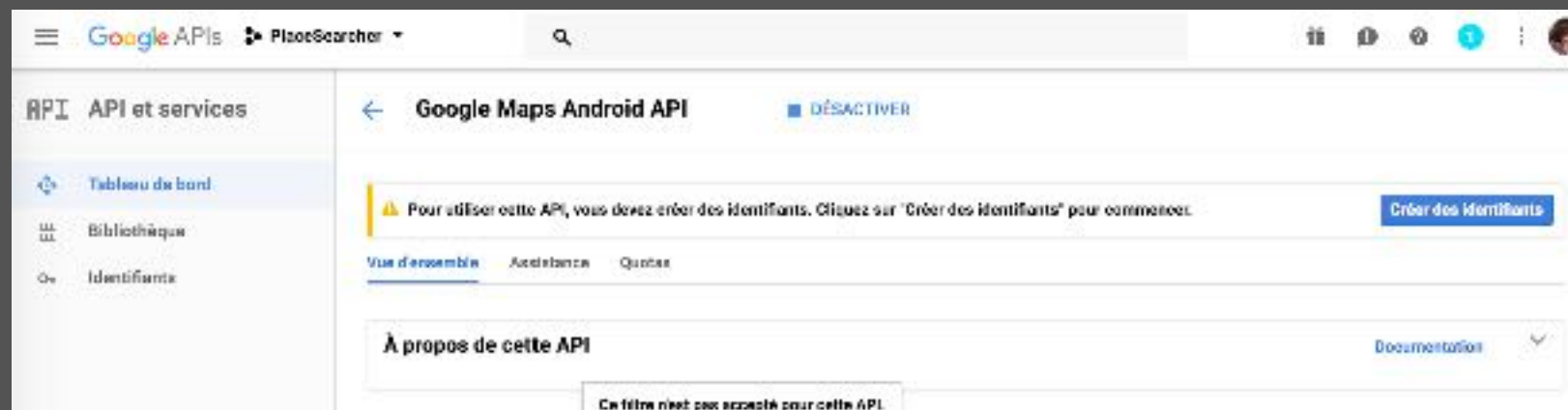
- Personnalisable à 100%
  - UI disponible (zoom, boutons)
  - Type de carte (vue satellite, mixte...)
  - Contrôles tactiles (pinch rotation, zoom...)
  - Position de l'utilisateur
- Facile d'accès pour des besoins simples
- Peu très vite demander de bien connaître Android
  - Passage à l'échelle (clusters, asynchronicité)
  - Customisation (drawables & bitmaps...)



## 3 - Google Map sur Android



Étape 1 : obtenir une clé  
créer un compte et une clé API  
sur la Google Developer Console





## 3 - Google Map sur Android

### Étape 2 : préparer le build

Modifier le build.gradle

```
// Add google maps  
compile 'com.google.android.gms:play-services-maps:11.8.0'
```

inclure la clé dans le Manifeste  
pour l'exercice : AlzaSyBcRt3W8uifSjrhXBu2467iz9ciQqMNTzM

```
<application  
    android:allowBackup="true"  
    android:icon="@mipmap/ic_launcher"  
    android:label="PlaceSearcher"  
    android:roundIcon="@mipmap/ic_launcher_round"  
    android:supportsRtl="true"  
    android:theme="@style/AppTheme">  
  
    <!-- Google Maps API Key -->  
    <meta-data  
        android:name="com.google.android.geo.API_KEY"  
        android:value="AlzaSyBcRt3W8uifSjrhXBu2467iz9ciQqMNTzM"/>
```



## 3 - Google Map sur Android

### Étape 3 : ajouter des cartes à vos vues

Ajouter un MapFragment à votre layout (on pourra reparler des Fragments plus tard)

```
<fragment
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment" />
</LinearLayout>
```





## 3 - Google Map sur Android

### Étape 4 : obtenir la map dans l'activité

Récupérer le fragment dans votre activité et obtenir la Map

```
// Get map fragment
SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
    .findFragmentById(R.id.map);
mapFragment.getMapAsync( onMapReadyCallback: this);
```

Votre Activity doit implémenter OnMapReadyCallback

```
MapActivity extends ... implements OnMapReadyCallback {
```

Et la méthode correspondante

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mActiveGoogleMap = googleMap;
```



## 3 - Google Map sur Android

### Exercice 15

- Ajoutez un bouton pour switcher de list à map dans la MainActivity
- Créer une MapActivity contenant un MapFragment et un champ texte (vous pouvez copier-coller puis modifier activity\_main.xml)
- Faites en sorte qu'on puisse passer de la MapActivity à la MainActivity
  - sans perdre la valeur du champ texte de recherche
  - faire de même de la MainActivity à la MapActivity
- Faire une pull request pour proposer votre exercice
- Optionnel : modifier les paramètres de la carte
  - Obtenir la carte avec `getMapAsync()` et `onMapReady()`
  - Changer le type de carte : satellite
  - Activer les boutons permettant de modifier le niveau de zoom
  - Consulter la doc Google pour voir les paramètres disponibles







### 3 - Google Map sur Android

Vous avez la main sur les marqueurs :

- position
- titre et snippet (affichés dans l'infoWindow)
- apparence (icône voir même vues complexes)
- onClick : sur le marqueur et/ou l'infoWindow

```
MarkerOptions markerOptions = new MarkerOptions()  
    .position(new LatLng( 49.5, -1.5))  
    .title("Place du commerce")  
    .snippet("Une bien fameuse place");  
Marker marker = mActiveGoogleMap.addMarker(markerOptions);
```

```
mActiveGoogleMap.setOnInfoWindowClickListener(new GoogleMap.OnInfoWindowClickListener() {  
    @Override  
    public void onInfoWindowClick(Marker marker) {  
        PlaceAddress associatedPlace = mMarkersToPlaces.get(marker.getId());  
        if (associatedPlace != null) {  
            // Do something with the place  
        }  
    }  
});
```





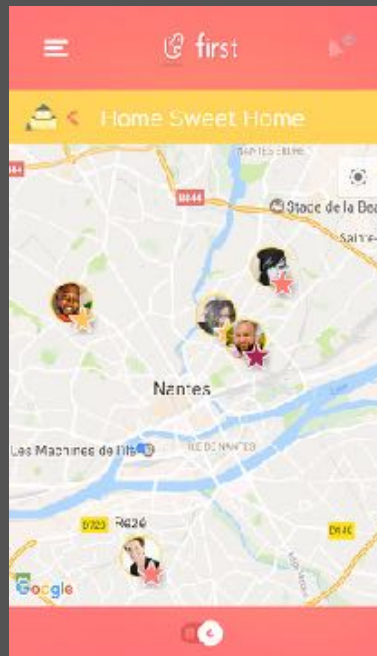
## 3 - Google Map sur Android

- Mouvements de caméra : devenez l'Edgar Wright des cartes !

```
mActiveGoogleMap.animateCamera(CameraUpdateFactory.newLatLngBounds(myBounds, width, height, padding));
```

Vous avez la main sur tout le reste :

- Listeners de région visible (pour comportements complexes e.g. live updates)
- Création de clusters (pour la montée en charge et l'user experience)
- Affichages complexes (itinéraires KML, formes GeoJSON...)



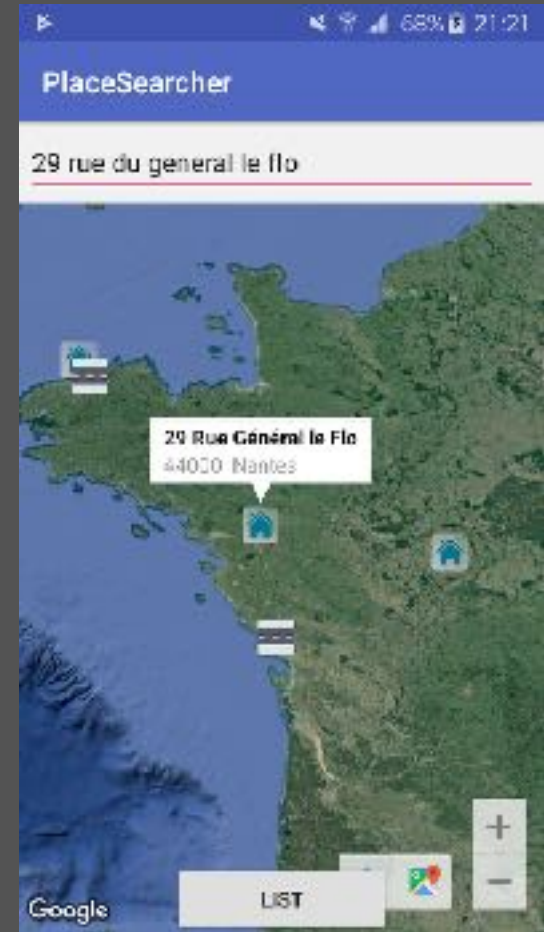




## 3 - Google Map sur Android

### Exercice 16

- Modifier le modèle : stocker la latitude/longitude des points retournés
- Faire en sorte que la MapActivity affiche les Place avec des marqueurs
- Au clic sur l'info window d'un marqueur, lancer la PlaceDetailActivity
- Faire une pull request pour proposer votre exercice
- Optionnel : centrer la caméra sur les résultats (cf LatLngBounds)
- Optionnel : fournir vos propres icônes de marqueurs
- Optionnel :
  - Vous devez constater que la recherche entraine des ralentissements
  - Comprendre pourquoi et m'appeler
  - Si votre explication est validée, mettre en place un correctif
- A faire chez vous (optionnel) : créer des clusters pour regrouper les marqueurs <https://developers.google.com/maps/documentation/android-api/utility/marker-clustering>





### 3 - Google Map sur Android

Pour éviter d'effectuer trop souvent la même tâche, le Scheduler est un excellent outil

```
public void searchPlacesFromAddress(final String search) {  
    // Cancel last scheduled network call (if any)  
    if (mLastScheduleTask != null && !mLastScheduleTask.isDone()) {  
        mLastScheduleTask.cancel(b: true);  
    }  
  
    // Schedule a network call in REFRESH_DELAY ms  
    mLastScheduleTask = mScheduler.schedule(new Runnable() {  
        // ...  
    })  
}
```

# THAT'S ALL FOLKS !

## NEXT TIME

- Base de données Mobiles
- Notifications
- Bonnes pratiques

## D'ICI LÀ

- Réfléchissez au sujet de TP
- Harcelez-moi ([alex.morel@irealite.com](mailto:alex.morel@irealite.com))

