**Google APIs via OAuth 2.0 on Android**

1. To handle  [Google APIs Client Library](http://code.google.com/p/google-api-java-client/) using OAuth 2.0 you may need to get so called ***token*** for future API requests. This task can be highly simplified and secured on all Android 2.2+ devices with Google Play services.
   1. Preparation. In view of the fact that there can be devices on which Google Play services is unavailable or can be disabled or invalid - you need to check method [isGooglePlayServicesAvailable()](https://developers.google.com/android/google-play-services/reference/com/google/android/gms/common/GooglePlayServicesUtil#isGooglePlayServicesAvailable(android.content.Context)).
2. To get ***token*** using GoogleAuthUtil you`ll need ***account name*** and ***scope***.
   1. ***Account name*** is string,that contains email, for example [nexus.moc@gmail.com](mailto:nexus.moc@gmail.com). You can get it using AccountPicker, which let user choose one of loginned accounts, or add existing or even create a new one.
   2. ***Scope*** is space delimited set of permissions the application requests. In our case this is string, which begins from “oauth2:”, and contains permissions requests.

***Scope*** *example:* “oauth2:https://www.googleapis.com/auth/plus.me https://www.googleapis.com/auth/userinfo.profile https://www.googleapis.com/auth/plus.moments.write”

Avaliable sets of permission requests available at <https://developers.google.com/oauthplayground/?hl=ru>

* 1. Than you need to call *GoogleAuthUtil.getToken(Context context,String email, String scopes);,* which return string ***token***, or throw exception. UserRecoverableAuthException can be thrown in case if user did not accept app permissions request yet. If so – you need to start activity, which requests user acceptation using intent of thrown exeption. After that you can restart *getToken* method.

*Notice: Good practice here is to invalidate with GoogleAuthUtil.invalidateToken(Context context, String token) method each* ***token*** *after each API request and get the new one before next.*