

„Willst du mit mir spielen?“ - Google Play™ Game Services in Android™

Im Mai 2013 war es soweit: Google veröffentlichte seine Google Play Game Services! Seitdem können Entwickler ihren Android Spielen das "gewisse Etwas" verpassen und den Spieler zum "Erfolgesammeln" und "Bestenlistentoppen" motivieren. Doch nicht nur das! Ein Multiplayer-Feature bringt die API auch mit und verspricht eine einfache Integration in meine Spiele-App. Ein exemplarisches Spiel [1] von Google zeigt mir den Gebrauch der Google Play Games Services und tatsächlich habe ich innerhalb kürzester Zeit meinen ersten Erfolg freigeschaltet. Sollte es wirklich so einfach sein? Hier ein kleiner Einblick in meine ersten Schritte mit den Game Services.

Google Play Developer Console

Zum Entwickeln mit den Game Services benötige ich zuerst einen Account für die Google Play Developer Console. Dieser kostet 25 Dollar und ist nicht nur ein (Google-)Leben lang gültig, sondern kann von einer unbeschränkt großen Horde Entwickler genutzt werden.

Nachdem ich die Registrierung abgeschlossen habe, muss ich nun in der Developer Console einen sogenannten Spielservice anlegen, um alle Erfolge und Bestenlisten meines Spiels zu verwalten. Dort kann auch eine Gruppe von Testern eingetragen werden, die meine App vor der Veröffentlichung testen können. Einmal angelegt, muss ich jetzt nur noch die soeben generierte "APP-ID" sowie die IDs der Erfolge und Bestenlisten als Ressource hinterlegen und schon habe ich Zugang zum Spielservice ([Abbildung 1](#)).

Für den Einsatz der Game Services auf dem Smartphone benötige ich nun einen Google+-Account und die Google Play Services App, welche ich als Entwickler später als "GamesClient" anspreche. Diese läuft als Hintergrundprozess auf dem Gerät und ich greife darauf zu, um Aktionen wie `signIn` oder `invitePlayer` durchzuführen. Die Google Play Services werden automatisch vom Play Store auf dem neusten Stand gehalten ([Abbildung 2](#)).

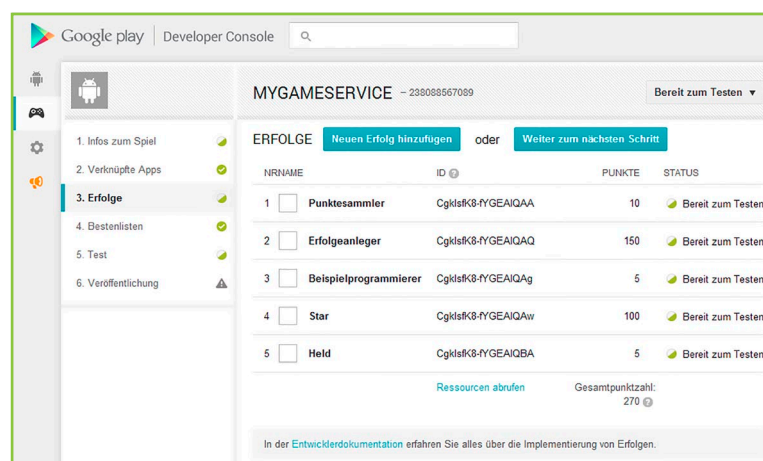


Abbildung 1: Screenshot - Erfolge in der Google Play Developer Console

Dass die Google Play Services App auf dem Gerät installiert ist, muss ich im Code selbst prüfen und sie gegebenenfalls nachfordern. Ein Fehlerdialog, der dem Nutzer das Herunterladen der App ermöglicht, wird zum Glück gleich mitgeliefert.

Meinen ersten Erfolg habe ich auch schon freigeschaltet. Da freut sich das Spielentwicklerherz. Doch wie kann ich nun jemals wieder das Freischalten dieses Erfolges testen? Jedes Mal einen neuen Google+-Nutzer anzulegen, wäre doch etwas zu umständlich. ▶

Also auf in die Developer Console, meinen Account aus der Liste des Spielservice gelöscht und wieder hinzugefügt. Jetzt sind all meine Erfolge wieder gesperrt und ich kann von neuem beginnen. Dies ist praktisch, um vor der Veröffentlichung noch einmal alles zu testen. Während der Entwicklung erweist sich eine Logausgabe jedoch als unkomplizierter.

Das Eintragen der Punkte in die Bestenliste ist vergleichbar schnell umgesetzt und meine App kann veröffentlicht werden. Doch was passiert mit meiner Punktzahl an der Spitze der Bestenliste, die wegen eines vorübergehenden Berechnungsfehlers unerreichbar hoch ist? Auch das ist kein Problem, denn beim Veröffentlichen eines Spiels werden alle Bestenlisten zurückgesetzt.

Test it!

Doch mit dem Veröffentlichen ist es bei meiner App nicht vorbei! Ich möchte weiterhin Features hinzufügen und testen. Wenn ich nun aber einen Erfolg hinzufüge, erscheint dieser bereits in den Erfolgslisten der Spieler. Wie kann ich das umgehen? Am besten lege ich einen "Geheimen Erfolg" an, denn ich denke, wenn der Erfolg noch nicht in der Erfolgsliste der Nutzer steht, bekommt der Nutzer ja auch nichts von meinem neuen Feature mit. Schön wär's, denn geheime Erfolge werden den Nutzern tatsächlich als "Geheimer Erfolg" angezeigt! Schade eigentlich. Bei Bestenlisten ist es das Gleiche! Noch ehe die Liste überhaupt von einem App-Update genutzt wird, ist sie schon in der Bestenlistenauswahl zu sehen.

Mir bleibt also nur ein Ausweg: Ich muss einen zweiten Spieldienst anlegen, der nochmal alle Erfolge und Bestenlisten enthält und mir neue IDs generiert. Leider etwas umständlich, aber der "Ressourcen abrufen"-Button erleichtert mir das Leben etwas. Wenigstens die IDs muss ich so nicht

alle einzeln kopieren. Da jeder neue Spieldienst auch eine eigene digitale Signatur erfordert, erstelle ich zu guter Letzt noch ein neues Zertifikat, um dann endlich meinen Test fortzusetzen.

Make it Google Style

Die Gestaltung der Game Services nimmt mir Google letztendlich auch ab. Gar nicht unansehnlich werden mir Achievements, Leaderboards, Waiting Room etc. einheitlich in einem Google-Default-View zur Verfügung gestellt (Abbildung 3).

Einen eigenen View zu erstellen, ist etwas für ambitionierte Spielentwickler, denn da lässt mich der Developer Guide im Stich. Also hängele ich mich mühsam durch die Dokumentation, versuche dem `LeaderboardScoreBuffer` den Erstplatzierten zu entlocken und erkläre dann den Google-View für meinen ersten Versuch als ausreichend. ▶

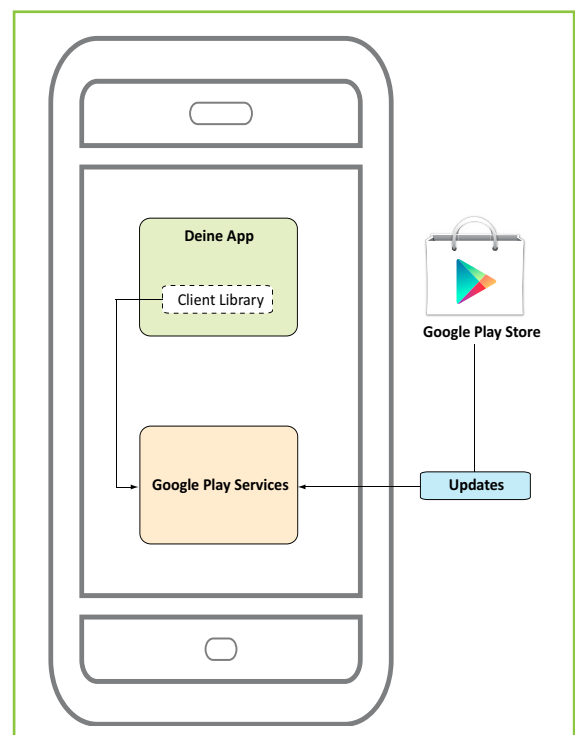


Abbildung 2: Die Google Play Services auf dem Gerät

Connect me

Für den Einsatz von Erfolgen und Bestenlisten muss ich während des Spielens bei Google+ angemeldet sein. Einen vorgefertigten SignIn-Button, den ich in meine `layout.xml` Datei einbinden kann und der bereits den Google Designrichtlinien entspricht, nehme ich natürlich dankend an.

Nun heißt es, Entscheidungen zu treffen. Ich bin nicht mit allen Empfehlungen einverstanden, die mir Google mit auf den Weg gibt und beginne von den Guidelines und Examples abzuweichen. Schneller als mir lieb ist, finde ich mich auf dem steinigen Weg des SignIn-Prozesses wieder.

Nein, ich möchte nicht bei jedem App-Start automatisch eingeloggt werden. **Ja**, ich möchte beim Beenden der App abgemeldet werden und **Ja**, ich möchte mehr als eine Activity in meinem Spiel benutzen. Für meine klaren Entscheidungen muss ich mit viel Nerven bezahlen, denn dafür ist die API nicht ausgelegt.

Schnell wird mir bewusst, dass "eingeloggt" nicht gleich "eingeloggt" bedeutet. Verschiedene Clients sorgen für die Verbindung zur Google Play Services App und zur Cloud. Über diese Clients kann ich dann z. B. `signOut` (zum Ausloggen von Google+) oder `loadState` (zum Laden von Daten aus der Cloud) aufrufen. Anfangs glaube ich irrtümlicherweise, dass mir `gamesClient.isConnected` Auskunft darüber gibt, ob der Nutzer bei Google+ eingeloggt ist oder nicht. Vielleicht glaube ich das, weil die `signIn`-Methode im mitgelieferten Beispiel einzig und allein die Clients verbindet und keinen separaten Google+-Login durchführt, vielleicht aber auch, weil die alleinige Verbindung zum GamesClient ohne eingeloggt zu sein, nutzlos ist und Fehler wirft. Wahrscheinlich aber glaube ich es, weil es die einzige Möglichkeit ist, überhaupt irgendeinen Verbindungsstatus des Nutzers zu erfahren. Der Moment, in dem mir mein Irrtum bewusst

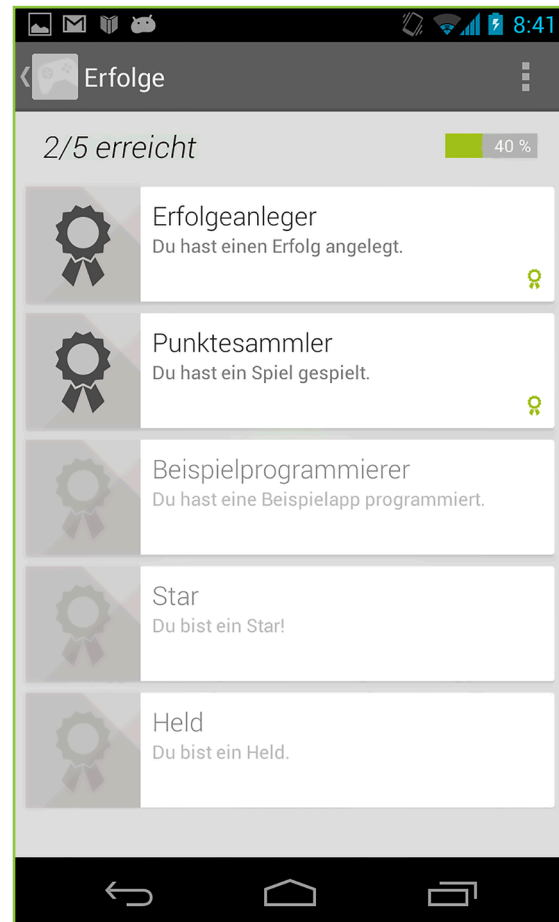


Abbildung 3: Screenshot - Erfolgsliste im Google-Default-View

wird, ist auch der Moment, in dem ich erkenne, dass ich nicht von den Vorgaben hätte abweichen sollen. Denn diese sorgen dafür, dass in den meisten Fällen `connected` auch `signedIn` bedeutet.

Wenn ich meine App tatsächlich einmal beende und nicht nur in den Hintergrund verschwinden lasse, wird automatisch die Verbindung zum GamesClient getrennt. Ein `signOut` findet dabei aber nicht statt. Das führt dazu, dass ich beim nächsten App-Start nicht mehr mit dem GamesClient verbunden, aber dennoch bei Google+ eingeloggt bin. Eine `isSignedInToGooglePlus`-Prüfung gibt es nicht. Ich ►

versuche, dem Prinzip `connected==signedIn` konsequent zu folgen und in den meisten Fällen funktioniert das ganz gut. Das Ausloggen beim Beenden wird mir also mehr oder minder aufgezwungen. Wie gut, dass ich das sowieso wollte.

Wie sollte es anders sein, rächt sich in diesem Moment auch die Entscheidung für den Google-View bei Erfolgen und Bestenlisten, denn über diesen hat der Spieler die Möglichkeit, sich in den Einstellungen abzumelden. *"You are about to sign out of this game."* verspricht mir der erscheinende Dialog. Faktisch bestehen jedoch alle Verbindungen zum Client weiterhin. Auf `gamesClient.isConnected` kann ich also nicht mehr vertrauen. Woher soll ich nun also meinen Login-Status beziehen?

Das Einhängen in `onActivityResult` bringt ein eventuell stattgefundenes Ausloggen im Google-View ans Licht. Hier ist allerdings der `ResponseCode` `"INCONSISTENT_STATE"` eher vage. Ich kann also nur hoffen, dass ich die Verbindung nicht fälschlicherweise kappe und in einen `"EVEN_MORE_INCONSISTENT_STATE"` falle. Sieht so aus, als hätte ich nun alle Probleme des SignIn-Prozesses erkannt und behoben.

Multiplayer

Kommen wir nun zum Multiplayer-Feature. Es ermöglicht, ein Spiel mit verschiedenen Nutzern und verschiedenen Geräten an verschiedenen Orten und sogar von verschiedenen Betriebssystemen zu spielen. Was sonst aufwendige Programmierung erforderte, wird über eine API bereitgestellt. `InvitePlayer`, `WaitingRoom`, `AutoMatching` usw. - all dies ist über meinen Google+-Account möglich. Ich kann Freunde aus meinen Kreisen zum Spielen einladen oder mich anonym mit einem Wildfremden verbinden lassen.

Das Verbinden der Spieler miteinander wird von der Google Play Services App übernommen. Einladungen werden von dieser in der Toolbar am oberen Bildschirmrand meines Android-Gerätes angezeigt. Ich kann so jederzeit eine Einla-

dung erhalten, auch wenn die App gerade nicht geöffnet ist, ich ausgeloggt bin oder ich die App überhaupt nicht besitze. Zum Glück weist mich der Google Play Service beim Annehmen der Einladung darauf hin und stellt mir auch gleich einen Login- (bzw. Download-) Dialog zur Verfügung. Einige Leser ahnen es vielleicht schon: Das Einloggen über diesen Dialog funktioniert nicht wie erhofft. Ich werde bei Google+ eingeloggt, und diesmal wird tatsächlich versucht, eine Verbindung von meiner App zum `GamesClient` herzustellen. Dieser Versuch scheitert jedoch. Erst ein `OnConnectionFailedListener`, mit welchem ich die Verbindung erneut herstellen kann, bringt mir dann die gewünschte Lösung.

Unschön finde ich hierbei, dass beim Annehmen von Spieleinladungen meine Applikation einen kompletten Lifecycle durchlebt. Sie wird also zerstört und wieder kreiert. Dies führt in meinem Fall auch dazu, dass ich ausgeloggt werde und erst nach erneutem manuellen Einloggen das Multiplayer-Spiel startet, denn ich habe mich ja dummerweise gegen ein automatisches Einloggen beim App-Start entschieden. Ein `android:launchMode=SingleInstance` für die Activity in der `AndroidManifest` beseitigt nun auch dieses Problem. Glaube ich, denn so einfach ist es dann doch nicht: Die Information, dass ich eine Einladung erhalten habe, bekomme ich nämlich ausschließlich als Antwort auf das Verbinden mit dem `GamesClient`. Doch bin ich dank meines `SingleInstance` schon mit dem `GamesClient` verbunden und komme an keine Einladung heran. Eine Zwickmühle, aus der ich mich momentan nur zu behelfen weiß, indem ich den `GamesClient` nochmal verbinde, obwohl er schon verbunden ist. Unschön, aber es funktioniert und ich erhalte in der `onConnected`-Methode der Google Play Services meine Einladung mitgeliefert.

Zum "Einladung annehmen" wird mir wieder ein Google-View zur Verfügung gestellt. Wieder hat der Nutzer die Möglichkeit, sich über diesen abzumelden. Für den abgemeldeten Spieler kenne ich die Lösung des Problems, doch der Einladende wartet vergeblich auf das Annehmen oder Ablehnen seiner Einladung. ►

Ich könnte jetzt Folgendes tun: Ich trenne die Verbindung des abgemeldeten Nutzers zum GamesClient, um ihn danach wieder einloggen zu können. Dann warte ich auf den `onConnected`-Aufruf, lehne die Einladung ab und logge den Nutzer endgültig aus. Ich habe mich dagegen entschieden und lasse den Gegenspieler warten.

Einmal die Einladung angenommen, sollte mein Nutzer sich besser nicht vom Fleck bewegen, denn sobald ich meine Internetverbindung verliere, werde ich vom Multiplayer-Raum getrennt. Die daraus entstehende Misere des fehlenden Mitspielers darf der Entwickler ausbaden, denn über das Verlassen des Raumes wird der Gegenspieler in diesem Fall nicht informiert. Eben dieses Problem zeigte sich auch bei der ersten Live-Präsentation des Multiplayer-Dienstes bei der Google I/O Keynote, bei der man nicht mehr als ein *"Waiting for other Players to join"* zu sehen bekam. Erklärung des Sprechers: *"This is not the most networking friendly room"*.

Ist die Einladung angekommen und angenommen, sind alle Clients verbunden, alle Teilnehmer bei Google+ angemeldet und im Besitz einer stabilen Internetverbindung, dann ist die weitere Implementierung des Multiplayer-Dienstes einfach.

Die eigentliche Spielkommunikation funktioniert über Nachrichten, die ich an einen bestimmten Mitspieler schicken kann. Will ich eine Nachricht an alle schicken, muss ich eben an alle eine Nachricht schicken. Laut Dokumentation beträgt die maximale Größe einer Nachricht 1.400 Bytes, was die Möglichkeiten der Kommunikation durchaus einschränkt. Unwissend dieser Tatsache habe ich bedeutend längere Nachrichten erzeugt und verschickt und siehe da, sie sind auch angekommen. Nun weiß ich nicht, ob ich an der Dokumentation oder an der dauerhaften Funktionalität meiner App zweifeln soll und hoffe auf Ersteres. Sollte ich einmal betriebssystemübergreifend senden wollen, würde ich mir die Nachrichtenlänge nochmal genauer anschauen.

Fazit

Alles in allem bieten die API eine einfache Integration der Google Play Game Services. Hält man sich bei den Details an den Developer Guide und nutzt den Google-View, kann man mit wenigen Codezeilen Bestenlisten und Erfolge hinzufügen. Eine Einbindung des Multiplayer-Features erfordert etwas Fingerspitzengefühl bezüglich Anmeldung, Einladung, Netzwerkproblemen sowie dem Umgang mit Nachrichten. Unter guten Bedingungen funktionieren die Game Services zuverlässig. Sind jedoch nicht alle Einstellungen exakt, wird es schnell umständlich, alle möglichen Fehlerquellen auszumerzen. Es gibt zu viele Varianten, den `SignIn`-Status zu stören und jede benötigt eine andere Vorgehensweise zur Behebung. Stünde mir ein `isSignedInToGooglePlus`-Check zur Verfügung, wäre die Welt auf jeden Fall ein bisschen einfacher und die Integration der Google Play Game Services etwas angenehmer. ■

Kathrin Heindl

Quellen:

- [1] <https://developers.google.com/games/services/android/quickstart>

Kontakt

buschmais GbR
Leipziger Straße 93
01127 Dresden

Tel +49 (0)351 3209230
Fax +49 (0)351 32092329
E-Mail info@buschmais.com
Web www.buschmais.de