Google play instant app

Contents

[**Uvod u Google Play Instant** 1](#_Toc125997163)

[Kako instant experience radi? 1](#_Toc125997164)

[“Instant Try” experience na Play Store-u 2](#_Toc125997165)

[“Instant Play” experience u Play Games aplikaciji 2](#_Toc125997166)

[Omogucavanje Instant iskustva smanjenjem velicine aplikacije ili igrice 2](#_Toc125997167)

[**Podesavanje Instant aplikacija** 3](#_Toc125997168)

[Podesavanje razvojnog okruzenja 3](#_Toc125997169)

[Informacije o potrebnim uslovima pokretanja 3](#_Toc125997170)

[Konfigurisanje instant enabled app bundle-a 4](#_Toc125997171)

[Dodavanje logike za instant experience workflow 5](#_Toc125997172)

[Build the app bundle 6](#_Toc125997173)

[Testiranje instant experience-a 6](#_Toc125997174)

[Publish-ovanje app bundle na produkciju 7](#_Toc125997175)

[**Smanjivanje velicine aplikacije u cilju poboljsanja instant experience-a** 7](#_Toc125997176)

[Refactor into multiple feature modules 7](#_Toc125997177)

# **Uvod u Google Play Instant**

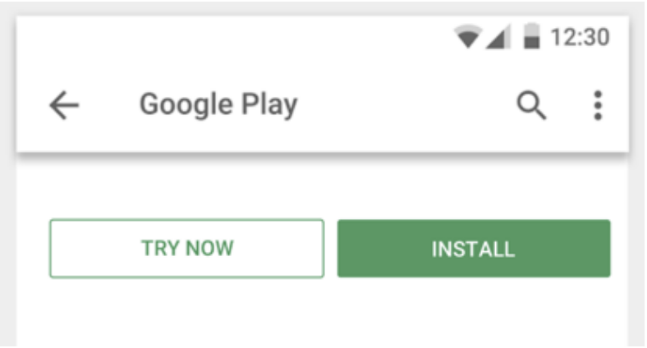
**Google Play Instant** omogucava native aplikacijama ili igricama da se pokrenu na uredjajima koji pokrecu Android > 5.0 bez instalacije. Kreiranje ovakvog ponasanja aplikacije (nazvan **instant apps** ili **instant games**) je moguce uz pomoc Android Studia. Omogucavajuci korisnicima da pokrenu instant app ili instant game, poznato kao pruzanje **instant experience**-a, unapredjujemo app discovery, sto ce doprineti vecem broju korisnika I preuzimanja sa Google Play store-a.

## Kako instant experience radi?

Uz pomoc Google Play Instant-a, korisnik moze odabrati opciju koja ce ovo aktivirati na Play Store-u bez dodatne instalacije aplikacije Ili igre. Kada Google Play primi zahtev za instant app ili igricu, salje samo nepotrebne fajlove Android uredjaju koji salje request. Uredjaj tada moze da pokrene tu aplikaciju Ili igricu.

Instant experience se deli u dve kategorije:

* **Try experience** na Play Store – za aplikacije
* **Instant play** – za mobilne igre



### “Instant Try” experience na Play Store-u

Ova opcija je dostupna koriscenjem Try now dugmeta na Play Store. Ovaj tip je obicno manja trial verzija aplikacije ili igrice sa ciljem da da uvid korisniku sta ga ocekuje kad preuzme kompletnu aplikaciju. Npr. game developer-i mogu da kreiraju prvi nivo igrice kao instant experience I onda da omoguce korisnicima da taj nivo probaju.

Google play Instant takodje omogucava prikazivanje dialoga unutar instant experience-a gde se korisnik poziva da instalira kompletnu igricu na svoj uredjaj.

### “Instant Play” experience u Play Games aplikaciji

Kompletne igre koje su upakovane koristeci Google Play Instant su dostupne za koriscenje na homepage stranici Google Play Games aplikacije. Ove igre su full experience igre, ne trial verzije. Korisnici koji pokrenu igricu u ovom rezimu mogu da je igraju bez prethodne instalacije.

## Omogucavanje Instant iskustva smanjenjem velicine aplikacije ili igrice

Aplikacija ili igrica mora biti manja od 15 mb da bi se omogucio instant rezim, ali sto je aplikacija manja to je lakse I bolje za koriscenje.

# **Podesavanje Instant aplikacija**

Google play experience se u aplikaciji ukljucuje kao deo Android App Bundle-a. Taj bundle je poznat kao instant-enabled app bundle. Ova sekcija ce pokazati kako se development okruzenje podesesava za instant app bundle razvoj, kao I kako se konfigurise, kreira, testira I publish-uje instant app bundle.

## Podesavanje razvojnog okruzenja

Da bi se omogucio instant mod unutar app bundle-a, potreban je pristup ka Google Play Instant Development SDK. Ovaj sdk se moze instalirati preko Android Studio ili preko komandne linije.

|  |
| --- |
| cd *path/to/android/sdk*/tools/bin && \  ./sdkmanager 'extras;google;instantapps' |

## Informacije o potrebnim uslovima pokretanja

Google Play Instant pokrece instalt enabled app bundle u specijalnom okruzenju SELinux sandbox sa odredjenim security nivoima. Ovaj sandbox ogranicava podniz permisija I ogranicava tipove interakciju sa drugim aplikacijama.

Instant enabled aplikacije podrzavaju samo sledece permisije:

* ACCESS\_COARSE\_LOCATION
* ACCESS\_FINE\_LOCATION
* ACCESS\_NETWORK\_STATE
* CAMERA
* INSTANT\_APP\_FOREGROUND\_SERVICE
* INTERNET
* READ\_PHONE\_NUMBERS
* RECORD\_AUDIO
* VIBRATE
* WAKE\_LOCK

Sledeca lista je lista permisija koje instant app bundle ne podrzava, I koje moraju biti uklonjene iz aplikacije u cilju omogucavanja ovog moda:

* ACCESS\_WIFI\_STATE – Umesto ovoga se moze koristiti ACCESS\_NETWORK\_STATE koji pruza slicne informacije
* BILLING – Ovo je deprecated permisija. Umesto toga treba koristiti Google Play Billing Library koja ne zahteva koriscenje ove permisije.
* READ/WRITE\_EXTERNAL\_STORAGE – Instant aplikacija ne sme imati pristup external storage-u, I umesto toga treba koristiti internal storage
* Com.google.android.c2dm.permission.RECEIVE I permission.C2D\_MESSAGE – C2DM je deprecated. Umesto toga treba odraditi migraciju na Firebase Cloud Messaging (FCM). FCM ne zahteva nekakvu odredjenu permisiju.

Dodatno, instant-enabled app bundle ne moze da odradjuje sledece stvari:

* Aplikacija ne moze da koristi background service
* Slanje notifikacija kada se aplikacija izvrsava u pozadini.

Kada razvijamo instant app, treba imati na umu da ta aplikacija ne moze komunicirati sa instaliranim aplikacijama na uredjaju sem u slucaju kada je jedna od ovih tacaka ispunjena:

* Jedan ili vise activity-ja unutar instalirane aplikacije treba da ima setovan flag **android:visibleToInstantApps** na **true**
* Instalirana aplikacija treba da sadrzi intent filter koji ukljucuje **CATEGORY\_BROWSABLE**
* Instant experience aplikacija salje intente sa akcijom **ACTION\_SEND, ACTION\_SENDTO, ACTION\_SEND\_MULTIPLE**.

## Konfigurisanje instant enabled app bundle-a

1. **Deklarisanje project dependency-ja** – Da bi se koristio Google play Instant API unutar aplikacije, potrebno je dodati sledeci dependency u build.gradle  
     
   **implementation “com.google.android.gms:play-services-instantapps:17.0.0”**
2. **Definisanje korektnog version koda –** Version kod aplikacije sa instant app experience-om treba da bude manji od version koda aplikacije koja se instalira. Ideja ovog pristupa jeste da korisnike privuce instant experience I da onda aplikaciju instaliraju na svom uredjaju. Tada ce tranzicija ove aplikacije na uredjaju biti jednaka update-tu te aplikacije.  
   Da bi se to postiglo, treba ispostovati jednu od dve strategije:
   1. Restartovanje version code-a za Google Play instant app na 1
   2. Povecavanje version koda za aplikaciju koja se instalira za neki veliki broj, poput 1000 da bi se postiglo da instant app version code ima prostora za povecanje
3. **Update target sandbox version –** Manifest fajl aplikacije mora biti izmenjen tako da gadja sanbox enviorement koje Google Play Instant podrzava. Ova izmena ukljucuje dodavanje android:targetSandboxVersion atributa unutar <manifest> elementa, kao sto je prikazano u sledecem parcetu koda:  
     
   <manifest  
   …  
     
   android:targetSandboxVersion=”2” … >
4. **Deklarisanje instant enabled modula aplikacije –** Unutar aplikacije, moguca je deklaracija da nasa aplikacija podrzava instant experience koristeci jednu od metoda:
   1. Ukoliko vec imamo postojecu aplikaciju koja sadrzi samo jedan modul, instant experience se moze ukljuciti sledecim koracima
      1. Otvoriti Project panel iz **View->Tool Window -> Project**
      2. Desni klik na base module, obicno nazvan app I onda izabrati **Refactor->Enable Instant App Support**
      3. U dialogu koji se prikaze, izabrati base module iz menija
      4. Kliknuti na **OK**
5. **Implementirati podrsku za SIGN IN –** Ukoliko aplikacija podrzava sign in korisnika, instant enabled aplikacija mora podrzavati **SmartLock for Passwords on Android**. Ukoliko pravimo Instant Play igru, potrebno je koristiti Google Play Games Service umesto sign in-a.
6. **Podrzati execution environment –** Da bi bili kompatibilni sa SELinux sandbox u kome se instant experience izvrsava, treba imati na umu sledece cinjenice prilikom kreiranja ovakve aplikacije:
   1. **Ne treba shareovati vrednost myUid()**, sto predstavlja UID aplikativnog procesa u kome se aplikacija izvrsava
   2. Ukoliko aplikacija gadja Android 8.1 ili manji, potrebno je kreirati Network Security Config fajl, I setovati **cleartextTrafficPermitted** na false. Instant experience ne podrzava HTTP saobracaj. Ukoliko aplikacija targetira Android 9 ili veci, cleartext traffic je disabled po default.
   3. Instant experience podaci ostaju download-ovani na korisnickom uredjaju dok se ne obrise instant experience cache, sto se dogadja u sledecim situacijama:
      1. Instant experience cache je garbage collected kada uredjaj ostaje bez dostupne memorije
      2. Korisnik restartuje uredjaj
   4. Ukoliko sistem ostaje bez storage prostora, moguca je situacija da korisnicki podaci iz instant experience aplikacije ce biti obrisani iz internal storage. Samim tim, preporuka je da se periodicno sinhronizuju korisnicki podaci sa aplikativnim serverom u cilju cuvanja korisnickog progresa.

## Dodavanje logike za instant experience workflow

Nakon sto se odradi konfiguracija aplikacije u cilju podrzavanja instant experience-a, potrebno je dodati logiku koja bi unapredila korisnicko iskustvo aplikacije. Neki primeri su:

* Provera da li se aplikacija izvrsava u instant experience modu – Ukoliko logika aplikacije zavisi od toga da li korisnik koristi aplikaciju kao instant experience mode-u, programski to mozemo proveriti sa metodom **isInstantApp()**. Ova metoda vraca true ukoliko je trenutni process aplikacije u instant experience modu.
* Prikazivanje install prompt dialoga – Ukoliko radimo na trial verziji aplikacije ili igre, Google Play Instant nam omogucava da prikazemo korisniku dialog gde pozivamo korisnika da instalira full experience aplikaciju na uredjaj. Da bi se ovaj dialog prikazao, treba koristiti InstantApps.showInstantPrompt() metodu

|  |
| --- |
| public class MyInstantExperienceActivity extends AppCompatActivity {     // ...     private void showInstallPrompt() {         Intent postInstall = new Intent(Intent.ACTION\_MAIN)                 .addCategory(Intent.CATEGORY\_DEFAULT)                 .setPackage(your-installed-experience-package-name);          // The request code is passed to startActivityForResult().         InstantApps.showInstallPrompt(MyInstantExperienceActivity.this,                 postInstall, request-code, /\* referrer= \*/ null);     } } |

* Transfer podataka iz instant appa ka instaliranoj aplikaciji – Ukoliko korisnik instalira aplikaciju na osnovu instant app-a, moramo voditi racuna da se korisnicki podaci iz instant appa prenesu na full verziju aplikacije. Ukoliko korisnik koristi uredjaj koji se izvrsava na Androidu 8 ili veci, ili ako aplikacija specificira targetSandboxVersion = 2, tada ce korisnicki podaci biti preneseni automatski. U suprotnom slucaju, podaci se moraju preneti rucno. Da bi se to postiglo, potrebno je koristiti neki od ponudjenih API:
  + Za korisnike koji koriste instant app na uredjaju koji izvrsava Android 8 ili veci, mozemo koristiti Cookie API.
  + Ukoliko korisnicki koriste uredjaje koji pokrecu Android 7.1 ili manji, mozemo koristiti Storage API.

## Build the app bundle

Za build instant enabled aplikacije mozemo koristiti Android Studio ili interfejs komandne linije. Prilikom koriscenja Android Studio-ja build mozemo pokrenuti selekcijom opcije **Build -> Build Bundle(s)/APK(s) -> Build Bundle.**

## Testiranje instant experience-a

Pre publish-ovanja instant aplikacije, instant experience se moze testirati sa jedne od sledecih lokacija u cilju verifikacije funkcionalnosti:

* Instalirati na lokalni uredjaj preko Android Studio-a
* Instalirati na lokalnu uredjaj preko command line interfejsa
* Publish-ovati na **internal test track** preko Google Play Console

Ukoliko testiramo aplikaciju na lokalnoj masini preko Android Studio-a, treba obratiti paznju na sledece korake:

* Ukoliko imamo instaliranu aplikaciju na uredjaju, potrebno je obrisati je
* U android studiju, prilikom pokretanja aplikacije preko Run/Debug Configuration opcije, potrebno je enable-ovati checkbox **Deploy as instant application**

Ukoliko koristimo Google Play Console I opciju Internal test track, za publish aplikacije treba da ispostujemo sledece korake:

* Upload-ovati aplikaciju na osnovu upustva za upload-ovanje bundle-a na Play Consolu
* Spremiti upload-ovan bundle za release na internal test track
* Sign-in ovati se na internal test account na uredjaju I onda pokrenuti instant experience na Play Store-u sa opcijom Try Now

## Publish-ovanje app bundle na produkciju

Da bi se publish-ovao instant enabled app bundle, potrebno je ispratiti sledece korake:

* Ukoliko nismo pre toga, potrebno je potpisati app bundle za **release key** i upload-ovati app bundle preko Play Console
* U Play Console, otvoriti **Release management > Android Instant Apps**, I onda treba otici do **instant app production track.**
* Izabrati **Update From Library**, i selektovati instant enabled app bundle koji smo upload-ovali

# **Smanjivanje velicine aplikacije u cilju poboljsanja instant experience-a**

Google Play Instant pruza bogata I izvorna iskustva pritiskom na web vezu. Ljudi mogu da iskuse aplikaciju bez instalacije, sto omogucava bolji nivo I kvalitet angazovanja. Da bi se instant aplikacija ucitavala brzo kao tipicna mobilna web stranica, aplikacija mora biti sto manja. Da bi se to ispostovalo, ova sekcija opisuje neke best practices postupke koji ce unaprediti rad aplikacije.

## Refactor into multiple feature modules

Najvece unapredjenje u cilju smanjenja velicine apk-a se desava kada refaktorisemo aplikaciju u vise feature modula. Treba krenuti sa base feature modulom, I onda izdvajati odredjene workflow-ove u sopstvene module. Treba proglasiti odredjeni activity po modulu kao entry point I definisati unique URL za svaki feature module.

Kada kreiramo feature module-e, treba obratiti paznju da base feature modul bude sto manji. Konkretno treba obratiti paznju na delove aplikaciju koji zahtevaju pristup na neku zavisnu biblioteku. Ukoliko jedan feature modul koristi odredjenu biblioteku, ta biblioteka treba da bude ukljucena samo u tom feature modulu, a ne da bude importovan u base feature modul. Treba imati na umu da , kad dodje do publish procesa odredjenog feature modula, velicina fajla koji ukljucuje taj feature modul I base feature modul mora biti manja od 15 mb.

Neke preporuke:

* **Koristiti isti codebase za oba tipa aplikacije –** mozemo pojednostaviti process upravljanja projektom tim sto cemo koristiti isti modularni codebase za kreiranje I aplikacije koja se instalira I za aplikaciju koja prestavlja instant app
* **Od pocetka dizajnirati aplikaciju podeljenu na modularne feature –** cak iako aplikacija ima jedan workflow I zahteva samo jedan feature modul za sada, uvek je dobra ideja da se aplikacija od pocetka dizajnira za vise feature modula. Na taj nacin mozemo dodavati postojece module aplikaciji bez promena na originalnoj velicini prvobitnog feature-a.
* **Ne treba se fokusirati na velicinu feature modula na pocetku –** Velicina Feature modula se ne racuna za locally built biblioteke. Instant aplikacija se moze publishovati na internal test track bez obzira na velicinu, ali na **alpha** I **production** kanalima mora biti ispod 15 mb

Neke aplikacije, pogotovo one koje se razvijaju duzi vremenski period, sadrze odredjene resurse koje sama aplikacija vise ne koristi. Pored toga sto vodimo racuna da su aplikacioni moduli manji, treba obratiti paznju na ciscenje nepotrebnih resursa. Neke preporuke su:

* **Smanjiti velicinu fajlova slika** – Umesto koriscenja PNG, JPG I ostalih formata slika, mozemo koristiti WebP format koji ce znacajno smanjiti velicinu binarnog fajla.
* **Obrisati jezike koje ne koristimo** – Ukoliko aplikacija podrzava vise jezika, pozeljno je smanjiti lokalizacione resurse sto je vise moguce. Ovaj korak je donekle resiv ukoliko koristimo app compat biblioteku poput android.support.v7.appcompat. Ova biblioteka ukljucuje poruke u vise jezika koje aplikacija potencijalno ne podrzava.
* **Obrisati extra fajlove** – Aplikacija potencijalno ne koristi sve resurse koje je nekad koristila dodavanjem novih ili update-ovanjem postojecih. Da bi smo izbacili te nekoriscenje fajlove, mozemo koristiti Lint checker za specificne situacije.
* **Obrisati nekoriscenje biblioteke** – Kako aplikacija bude rasla, moze doci do porasta broja dependency-ja koje koristi u svom radu, pogotovo na dependency-je sa sledecim tipovima:
  + **Native biblioteke** – Biblioteke koje sadrze native kod koji instant app nikad nece izvrsavati
  + **Transitive dependencies** – biblioteke koje su importovane kroz biblioteke koje su direktno ukljucene u aplikaciju.
  + Android Studio ima nekoliko korisnik alata za identifikaciju nepotrebnih dependency biblioteka koje se nalaze u nasem projektu:
    - External libraries – Android Studio Project view ukljucuje External Libraries sekciju. Ova sekcija sadrzi sve biblioteke koje aplikacija koristi, ukljucujuci native kod I sve tranzitivne dependency-je. U ovom view-u mozemo traziti nekoriscene I duplikate biblioteka koje aplikacija ne treba
    - APK Analyzer – Ovaj alat se moze koristiti za uporedjivanje razlicitih build-ova ukljucujuci instant app build-ova.

Kada utvrdimo koje dependency-je ne trebamo, u build.gradle mozemo exclude-ovati te biblioteke na sledeci nacin:

|  |
| --- |
| dependencies {     implementation('some-important-but-large-library') {         exclude group: 'com.example.imgtools', module: 'native'     } } |