Instantgram\* dokumentacija

# Uvod

Instantgram je sustav koji služi za upload slika sa mobilnih uređaja, kojima su pridružene određene oznake (tagovi), te njihov prikaz na web sučelju. Razvijen je u svrhu natjecanja [Best Code Challenge v3.0](http://www.best.hr/code-challenge/v3.0/) i u razvoju su sudjelovali studenti FER-a Matej Trbara i Luka Mijatović. Web sučelje je dostupno na adresi <http://www.team37.host25.com/>, a Android aplikacija se može skinuti kao instalacijska datoteka na [http:/www.team37.host25.com/android](http://www.team37.host25.com/android). Pored web sučelja i android aplikacije razvili smo i REST API u kojem je implementirana logika i na kojeg se oslanjaju web sučelje i mobilna aplikacija. API je dostupan na <http://www.team37.host25.com/api/v1/>. Glavne korištene tehnologije pri razvoju:

* Web aplikacija – [Django framework](https://www.djangoproject.com/)
* Baza podataka - MySQL
* REST API – [Django-Tastypie](http://django-tastypie.readthedocs.org/en/latest/) framework za izradu API-ja
* Android aplikacija (nativna aplikacija) – Standardni AndroidSDK

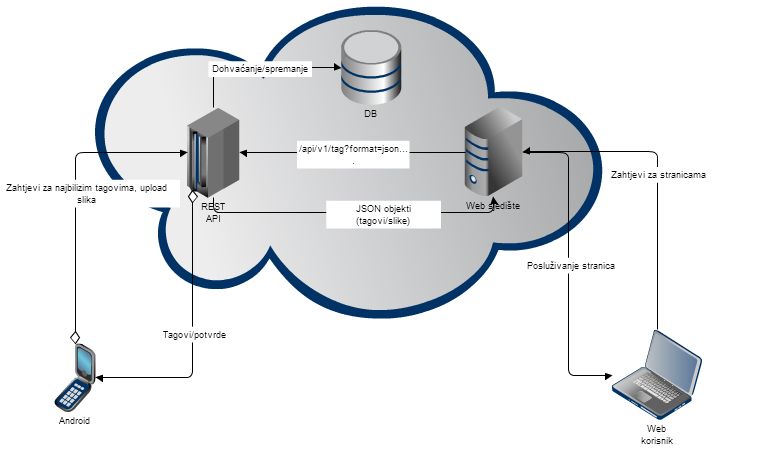
Pored navedenih tehnologija koristili smo i neke dodatne python i jquery biblioteke te HTML predložak:

* Računanje udaljenosti na osnovu GPS koodrinata – [geo.py](http://hoegners.de/Maxi/geo/)
* Kupili smo HTML predložak [Pinfinity](http://themeforest.net/item/pinfinity-responsive-tumblrlike-site-template/2776231)
* Jquery skripta za upload sa web sučelja – [jQuery.fineUploader-3.0.js](http://fineuploader.com/)
* Jquery skripta za generiranje oblaka tagova – [jqCloud-1.0.2.js](https://github.com/lucaong/jQCloud)
* Jquery skripta za prikupljanje geolokacija na web sjedištu – [jQuery.geolocator.js](https://github.com/fabiant7t/jQuery-Geolocator-Plugin)

Svi korišteni materijali imaju odgovarajuće licence, tj. nisu narušena prava korištenja.

# Arhitektura

Arhitektura sustava se zasniva na klijent/server modelu, gdje web sjedište i android aplikacija predstavljaju klijentski, a REST API serverski dio. Sva interakcija sa bazom podataka (dohvaćanje, dodavanje, filtriranje itd.) kao i poslovna logika (računanje najbližih tagova, podešavanje težina/važnosti tagova itd.) se obavlja kroz API na klijentski zahtjev. Komunikacija između web sučelja i API-ja, te android aplikacije i API-ja odvija se slanjem odgovarajućih HTTP zahtjeva sa potrebnim podacima (URL parametri i/ili JSON objekti).



Arhitektura sustava

## Web i API

Budući da se nalaze na istom serveru, podjela između Web sučelja i API-ja je logička. Implementacija je realizirana kao standardni Python Django projekt. U nastavku je objašnjena struktura projekta i implementacija najvažnijih dijelova.

### /team37

Korijenski direktorij projekta je /team37. U njemu se nalaze ostali poddirektoriji sustava te manage.py datoteka koja služi za interakciju sa django porjektom (sinkronizacija baze podatka, pokretanje debug servera...)

### /team37/team37

Sadrži zajedničke datoteke koje se koriste u ostatku projekta. settings.py sadrži postavke koje koje se koriste u projektu, osim defaultnih od django frameworka u ovoj datoteci su definirane i neke specifične za Instantgram, kao npr. NEAR\_DISTANCE koja predstavlja radius pri traženju okolnih tagova (radius je konfigurabilan, ova vrijednost se koristi kao defaultna). urls.py definira URL obrasce koji se koriste za pristup sustavu.

### /team37/team37/main

Glavni direktorij u kojem se nalazi implementacija konkretnog sustava. urls.py sadrži URL obrasce za web sučelje, a views.py je django specifična datoteka koja definira koje stranice treba korisniku poslužiti na određeni zahtjev (index, popis slika....) . admin.py služi za aktivaciju admin sučelja koje se nalazi na adresi http:/www.team37.host25.com/admin. geo.py je biblioteka pomoću koje se izračunava udaljenost između dvije zadane GPS koordinate.

### /team37/team37/main/models

U ovom direktoriju definirani su poslovni objekti s kojima radi sustav. Instantgram definira dva objekta, Image i Tag. Međusobno su povezani many-to-many vezom. Django framework koristi ove dvije klase za automatsko generiranje baze podataka i koristi ih u ugrađenom ORM sustavu što olakšava komunikaciju sa bazom podataka za ostatak sustava. Dodane ImageManager i TagManager klase proširuju ORM dodatnim funkcionalnostima za napredno filtriranje, poput dohvaćanja svih tagova koji su u nekom radijusu od zadane geolokacije ili dohvaćanaj svih tagova koji su dodani u zadnjih n sati.

### /team37/team37/main/api

REST API implementiran je koristeći django-tastypie framework. Framework zahtjeva definiranje REST resursa s kojima radi. Resursi su definirani u resources.py datoteci i analogni su definiranim objektima modela (ImageResource i TagResource). U ovim klasama definira se način posluživanja resursa u ovisnosti o vrsti prispjelog zahtjeva na API (metode apply\_filters). Svi zahtjevi (web i android klijent) u sustavu idu preko REST API-ja. ImageResource je proširen u odnosu na model iz razloga što se sadržaj slike ne sprema u bazu podataka nego na disk, te se iz tog razloga mora dodatno obraditi prije nego se proslijedi ORM-u. Fotografija se iščitava iz polja img pristiglog JSON objekta kao base64 enkodirani string te se u overrideanoj metodi hydrate\_img dekodira i sprema na disk. Dodatno se iz sadržaja base64 stringa generira md5 hash koji se provjerava pri spremanju slike, te ukoliko zadani hash postoji u bazi, slika se ne sprema (spriječava upload istih slika sa mobilnog uređaja).

### /team37/team37/main/templates

Definicija HTML stranica web klijenta. Base.html predstavlja osnovni okvir koji se koristi u svim stranicama. Sadržaj svake pojedine stranice se učitava pomoću ajax poziva u bloku content, čime se izbjegava refreširanje čitave stranice već samo dijela sa sadržajem. Tagcloud.html prikazuje oblak tagova koji su vraćeni od strane API-ja na pojedini zahtjev. Klikom na pojedini tag korisiku se prikazuju slike koje su označene tim tagom (images.html).

## Android aplikacija

Cilj nam je bio napraviti što jednostavniju Android aplikaciju, za korisnika i developera. Aplikacija se sastoji od jednog ekrana na kojem se prikazuje slika s kamere. Dodirom na bilo koji dio ekrana kamera snimi sliku, a korisniku se prikažu opcije za slanje. Ukoliko je uključena geolokacija na mobitelu, korisniku se prikazuju preporučeni tagovi. Korisnik može odabrati jedan ili više preporučenih tagova te upisati dodatno svoje (odvojene razmakom). Klikom na „Send pic“ slika se šalje na server, a korisniku se prikazuje poruka kao rezultat slanja.

### Implementacija

Pri pokretanju aplikacije pristupa se kameri uređaja. Kamera se inicjalizira tako da se slika prikazuje pomoću klase CamPreview. Prilikom izlaska iz aplikacije (onPause()) oslobađa se resurs kamere, a prilikom nastaka rada (onResume()) kamera se ponovo inicjalizira. Klasa MainActivity predstavlja glavni ekran aplikacije. Proces slanja slike započinje u trenutku kad korisnik dotakne ekran (onTouchEvent()):

* Asinkroni događaj onPictureTaken() označava da je slika spremljena u memoriju, te se na ekranu više ne prikazuje sadržaj iz kamere nego uslikana fotografija
* Odmah nakon toga se pokreće asinkroni zadatak (GetTagsTask) čiji je cilj poslati geolokacijske podatke na server i dohvatiti najbliže tagove
* Ako geolokacijski podaci nisu dostupni dohvaćaju se najpopularniji tagovi
* Po primitku odgovora sa servera, korisniku se prikazuje sučelje za odabir tagova i unos vlastitih (TagCallback)
* Nakon odabira tagova pritiskom na „Send pic“ pokreće se novi asinkroni zadatak (SendImageTask) koji pokreće postupak slanja slike na server
* Razmjena podataka izmedju aplikacije i servera se odvija pomoću JSON objekata. Radi toga se slika prije slanja prvo enkodira u base64 ASCII formatu te se sa dodatnim podacima (tagovi, geolokacija) šalje POST zahtjevom na server.
* Nakon prikaza statusne poruke, aplikacija se vraća na početni ekran

# Opis funkcionalnosti

U nastavku slijedi opis pojedinih funkcionalnosti sustava sa kratkim objašnjenjima i primjerima.

## Index stranica

Početna stranica po defaultu prikazuje do 50 najpopularnijih tagova. Tagovi sa većim brojem slika su prikazani veći. Tagove je moguće filtrirati na nekoliko načina:

* All – prikaz svih tagova
* New - prikaz tagova dodanih u zadnja 24 sata
* Near – sustav prikazuje tagove koji su u određenom radijusu u odnosu na geolokaciju web korisnika
* Popular – 50 najpoluarnijih tagova
* Search – Unosom teksta u search bar filtriraju se svi tagovi koji imaju kao podstring imaju unešeni tekst

Svi navedeni filteri se dinamički kreiraju i šalju kao upit API-ju. Npr. klikom na Near u pozadini se generira upit oblika:  
http:/www.team37.host25.com/api/v1/?format=json&lat=12.1234&lon=12.12345&rad=500 koji vraća sve tagove u krugu 500 metara od zadane zemljopisne širine i dužine.

Klikom na link Embed korisniku se ponudi Iframe embedd link koji se može ugraditi u druge stranice.

Upload link vodi na web formu za upload slika kroz web sučelje. Svaka slika može imati više tagova koji se unose sa odvojeni razmakom.

## Stranica za prikaz slika

Prikazuju se slike za zadani tag na način da novije slike imaju prednost nad starijima. Pri uploadanju nove slike sa istim tagom, ona se postavlja na početak dok se ostale pomjeraju prema kraju.