

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Projektová dokumentácia k inžinierskemu dielu

(Tím 22 - iBazár)

Akademický rok: 2017/2018

Predmet: Tímový projekt

Členovia tímu (študenti):

- Bc. Stanislav Kubík
- Bc. Vladimír Kuchár
- Bc. Viktor Matovič
- Bc. Slavomír Mazúr
- Bc. Matej Murín
- Bc. Daniel Nechala
- Bc. Patrik Papšo
- Bc. Márius Rak
- Bc. Martin Višvarda

Vedúci tímu:

- Ing. Jakub Ševcech, PhD.

Obsah

1	Úvod	2
2	Globálne ciele pre zimný semester	2
3	Globálne ciele pre letný semester	2
4	Dokumentácia - frontend	3
4.1	Používateľské rozhranie	3
4.2	Stav aplikácie	3
4.3	Komunikácia so serverom	4
4.4	Nasazdovanie frontendu	4
5	Dokumentácia - backend	4
5.1	Implementácia	4
5.2	Modul users	4
5.3	Modul places	5
5.4	Modul items	5
5.5	Prepojenie s Elasticsearch	5
5.6	Prepojenie s Redis	5
5.7	Nasádzanie backendu	6
5.8	Dátový model	6
6	Koncové body API	6
7	User Experience	9
7.1	Scenár používateľského testovania aplikácie iBazár	9
7.1.1	Poučenie	9
7.1.2	Úvodný rozhovor	9
7.1.3	Úlohy	9
7.1.4	Záverečný rozhovor	11
A	Príručka frontendu	12
A.1	Príprava spustenia projektu	12
A.2	Inštalácia	12
A.3	Spustenie	12
B	Príručka backendu	12
B.1	Príprava spustenia projektu	12
B.1.1	Nainštalovanie požiadaviek projektu	12
B.1.2	Virtuálne prostredie	13
B.1.3	Databáza	13
B.1.4	Logovanie	13
B.1.5	Mailing	14
B.1.6	Nastavenia	14
B.2	Prvé spustenie	14
B.3	Testovanie	14
B.4	Indexovanie do Elasticsearch	14

1 Úvod

Cieľom tohto dokumentu je popis inžinierskeho diela, ktoré je vytvárané počas práce jednotlivých členov tímu v rámci predmetu Tímový projekt, pričom tento popis poskytuje takzvaný "Big picture" vytváraného projektu. Dokument slúži ako technická dokumentácia projektu, pričom opisuje dve hlavné časti webovej aplikácie a to frontend a backend a zároveň koncové body API.

Dokument je členený do kapitol, pričom každá kapitola sa venuje inej časti našej webovej aplikácie. V rámci druhej kapitoly sa nachádza dokumentácia opisujúca frontend. Kapitola číslo tri naopak opisuje jednotlivé moduly backendu a prináša pohľad na dátový model aplikácie. V piatej kapitole sú uvedené jednotlivé koncové body API, pričom každý koncový bod obsahuje zoznam povolených operácií, ktoré sú v krátkosti opísané.

Šiesta kapitola obsahuje používateľskú príručku k backendu, kde sú uvedené potrebné požiadavky a zároveň sa tu nachádzajú kroky, ktoré je potrebné vykonať pred prvým spustením aplikácie.

2 Globálne ciele pre zimný semester

Cieľom projektu je vytvoriť webovú stránku na uverejňovanie a vyhľadávanie inzerátov na darovanie/predej/kúpu/výmenu vecí medzi jednotlivcami. Cieľom nie je vytvorenie kópie už existujúcich stránok ale zameranie sa na najväčšie problémy z pohľadu používania takýchto aplikácií ako napríklad:

- pokročilé vyhľadávanie na základe vlastností tovaru, geografického umiestnenia jeho majiteľa, veku inzerátu
- odstraňovanie neaktuálnych ponúk,
- podpora procesu výmeny alebo predaj a tovaru tak, aby mohli zákazník a ponúkajúci komunikovať priamo v aplikácii.

Dôraz je kladený nie len na funkcionality ale aj na formu (UI a hlavne UX). Proces umiestňovania inzerátu ako aj reagovania naň musí byť veľmi jednoduchý a rýchly, aby napríklad zdĺhavé vyplňovanie detailov o produkte neodradilo od ich vyplnenia. Je teda potrebné tento proces navrhnuť a implementovať tak, aby previedol používateľa zadávaním.

Hlavným cieľom je teda vytvoriť jednoduchý, prehľadný portál prístupný pre širokú masu na ponúkajúce vecí, ktorý by sa sústreďoval na vyhľadávanie produktov na základe ich atribútov a ich lokácie a nie len na základe fulltextového vyhľadávania, ako je to v súčasnosti bežné, a to všetko s využitím najmodernejších technológií.

3 Globálne ciele pre letný semester

Cieľom do letného semestra je optimalizovať existujúcu časť aplikácie, ako na frontende, tak aj na backende. Najväčšou výzvou a cieľom je hlavne vytvoriť prívetivý

používateľský dizajn, tak aby bol nie len jednoduchý ale aj použiteľný a zrozumiteľný. Vytvorený dizajn plánujeme otestovať na vybranej skupine ľudí a na základe testovania ešte viac zoptimalizovať.

Plánom je vytvoriť mechanizmus na odporúčanie podobných inzerátov, ktoré by malo využívať výhody Elasticsearchu, ktorý bol implementovaný v zimnom semestri a čo najlepšie tak odporúčať podobné inzeráty.

Ako posledný veľký cieľ je pridanie atribútov v závislosti od kategórie k jednotlivým inzerátom, čím sa ešte zefektívni predchádzajúca funkcionálnosť, zlepši sa odporúčanie a ako dúfame, tak aj používatelia sa dostanú k relevantnejším ponukám.

4 Dokumentácia - frontend

Frontend aplikácie je napísaný v jazyku JavaScript a môžeme ho rozdeliť na dva celky:

- používateľské rozhranie,
- globálny stav aplikácie jeho manažment.

4.1 Používateľské rozhranie

Používateľské rozhranie je implementované pomocou knižnice `React`¹ a oficiálneho štartéra `Create React App`², ktorý v sebe obsahuje požadovanú konfiguráciu. Pri vytváraní jednotlivých častí používateľského rozhrania sa riadime metodikou pre programovanie frontendu a teda komponentami riadeným vývojom. Prezentačné komponenty slúžia na zobrazenie dát, ktoré sú uložené v stave kontajnerov alebo globálnom stave. Kontajnery naopak čítajú dáta z globálneho stavu aplikácie a to za pomoci knižnice `react-redux`³ a taktiež môžu vyvolať akcie (napríklad po kliknutí tlačidla v prezentačnom komponente), ktoré tento stav zmenia.

Ďalej využívame knižnicu `Ant Design`⁴, ktorá nám poskytuje už predpripravené komponenty (tlačidlá, ikony ap.). Komponenty tejto knižnice sú naštýlované pomocou CSS frameworku `Less`⁵, ktorý na štýlovanie využívame aj my.

4.2 Stav aplikácie

Stav aplikácie je oddelený od prezentačnej vrstvy čiže používateľského rozhrania. Na manažovanie stavu využívame knižnicu `Redux`⁶. Stav aplikácie môže byť menený za pomoci knižnice `Redux Saga`⁷.

¹<https://reactjs.org/>

²<https://github.com/facebookincubator/create-react-app>

³<https://github.com/reactjs/react-redux/>

⁴<https://ant.design/>

⁵<http://lesscss.org/>

⁶<https://redux.js.org/>

⁷<https://redux-saga.js.org/>

4.3 Komunikácia so serverom

Na komunikáciu so serverom využívame knižnicu `axios`⁸. Táto knižnica nám umožňuje vykonávať HTTP požiadavky na aplikačné rozhranie, ktoré je opísané v časti 5 Dokumentácia - backend. Tieto požiadavky sú volané akciami, ktoré sú implementované pomocou vyššie spomenutej knižnice `Redux Saga`. Komunikácia so serverom je rozdelená do viacerých logických celkov, z ktorých každý obsahuje priečinky *api*, kde sú pokryté samotné endpointy pre spojenie s backendom, a *ducks* v ktorých sú tieto endpointy zaobalené do funkcií (akcii), ktoré sú následne využité v samotných kontajneroch (zabezpečujú logiku pre komponenty), ktoré túto funkcionality poskytujú prezentačným komponentom, čo je bližšie opísané v sekcii 4.1. Medzi tieto logické celky patria:

- `/auth` - obsahuje logiku potrebnú pre prihlasovanie, zmenu hesla a pod.
- `/items` - obsahuje logiku potrebnú pre získanie listu inzerátov, konkrétnych inzerátov, vytvárania inzerátov a pod.
- `/misc` - obsahuje doplnujúce funkcie pre autocomplete geografických lokácií,
- `/users` - obsahuje logiku potrebnú pre registráciu a vymazanie účtu.

4.4 Nasazdovanie frontendu

Vyššie spomenutý štartér `Create React App` má v sebe prednastavenú konfiguráciu pre tzv. `build proces`. Po jeho prevedení dostaneme priečnik, ktorý obsahuje minifikovaný JavaScriptový kód, `index.html` a potrebné `assets` (obrázky, písma ap.). Build sa vykoná po nasadení do `development` vetvy.

5 Dokumentácia - backend

Backend je napísaný v jazyku Python nad frameworkom Django. Kód je členený do modulov, ktoré obsahujú logické celky aplikácie. Tieto moduly sú popísané nižšie v tejto kapitole.

Backend poskytuje REST API pre frontend, ktorý je bližšie opísaný v časti 4 Dokumentácia - frontend. Vývoj API resp. backendu sa riadi metodikami, ktoré sú opísané v dokumentácii k riadeniu.

5.1 Implementácia

Dokumentácia kódu backendu je generovaná automaticky pomocou nástrojov Sphinx a Pycco. Jej aktuálna verzia je zverejnená na URL adrese: <http://docs.api.catch22.sk>

5.2 Modul users

Hlavnou úlohou modulu `users` je poskytovanie manažmentu užívateľov, užívateľských profilov, registráciu a autentifikáciu. Tento modul je závislý od knižnice tretej

⁸<https://github.com/axios/axios>

strany `allauth`⁹, ktorá okrem základnej registrácie, aktivácie účtu a prihlasovania poskytuje aj rôzne iné možnosti autentifikácie pomocou sociálnych sietí. Na umožnenie volania metód poskytovaných knižnicou `allauth` pomocou REST API je využitá knižnica `django-rest-auth`¹⁰. Rozšírením funkcionality týchto knižníc, umožňuje modul `users` registrovať užívateľa s vyplnením informácií v profile, autentifikovať užívateľa JWT tokenom alebo deaktivovať užívateľa s overením jeho hesla.

5.3 Modul places

Modul `places` poskytuje prepojenie aplikácie s Google Places API¹¹. Pomocou tohto modulu aplikácia poskytuje dokončovanie názvu vyplňovaného mesta a jeho GPS súradnice.

5.4 Modul items

Ide o modul, ktorý sa stará o správu inzerátov, t.j. pridávanie, úprava, *soft-delete*, ale taktiež má na starosti nahrávanie fotiek a následne priradenie k danému inzerátu. Tento modul obsahuje tiež kategórie inzerátov a ich hierarchiu.

5.5 Prepojenie s Elasticsearch

Dátový model *items* a s ním úzko súvisiace modely sú indexované v Elasticsearch, kvôli inteligentnejšiemu vyhľadávaniu, ako v relačnej databáze. Prepojenie je podmodulom *items*, keďže s ním úzko súvisí a poskytuje API pre vyhľadávanie nad indexami. Do tohto API je možné poslať query parametre, ktoré sú následne spracované a sú vyvolané vyhľadávacie query nad Elasticsearch databázou.

- existujúce dáta je možné dodatočne indexovať
- nové dáta sa automaticky indexujú, pričom sú využívané signály - pri pridaní nového, aktualizácii existujúceho alebo zmazaní inzerátu sa indexy tejto udalosti prispôbia

Rovnako ako *items* sa indexuje aj dátový model *categories* a tiež poskytuje v rámci modulu *items*, svoje API vyhľadávania nad indexmi *items/categories/search*.

V Elasticsearch využívame tiež pre lepšie výsledky slovenský analyzátor¹². Analyzátor používame v kombinácii so synonymickým slovníkom, ktorý sme obohatili o niekoľko moderných výrazov.

5.6 Prepojenie s Redis

Na počítanie návštev inzerátov a požiadaviek na server používame databázu **Redis**. Táto databáza funguje na princípe ukladania dát vo forme "kľúč "hodnota". Pri tejto implementácii na hodnote nezáleží, pretože všetky informácie sú obsiahnuté v kľúči.

⁹<https://www.intenct.nl/projects/django-allauth/>

¹⁰<http://django-rest-auth.readthedocs.io>

¹¹<https://developers.google.com/places/>

¹²<https://github.com/SlovakNationalGallery/elasticsearch-slovenčina>

našej API sa nachádza na adrese: <https://catch22.docs.apiary.io/>

Koncový bod: **/auth/**

Povolené operácie:

- POST - Po poslaní používateľských údajov vracia JWT Token, ktorý slúži ako autentifikácia pre ďalšie dopyty

Koncový bod: **/verify-email/**

Povolené operácie:

- POST - Po poslaní tokenu, ktorý používateľ obdržal emailom po registrácii aktivuje používateľský účet

Koncový bod: **/reset-password/:uidb64/:token/**

- POST - Po poslaní

Koncový bod: **/users/**

Povolené operácie:

- POST - Vytvorenie používateľa

Koncový bod: **/users/:id**

Povolené operácie:

- GET - Vypísanie informácií o používateľovi
- PUT - Úprava informácií o používateľovi
- DELETE - Vymazanie používateľa (soft-delete, t.j. deaktivácia účtu)

Koncový bod: **/items/**

Povolené operácie:

- GET - Vypísanie listu existujúcich inzerátov
- POST - Vytvorenie inzerátu, používateľ musí byť prihlásený

Koncový bod: **/items/:slug**

Povolené operácie:

- GET - Vypísanie daného inzerátu identifikovaného podľa parametra *:slug*
- PUT - Editovanie daného inzerátu, používateľ musí byť vlatstníkom
- DELETE - Vymazanie inzerátu (soft-delete)

Koncový bod: **/items/photos/**

Povolené operácie:

- POST - Nahrať fotku, ktorá bude následne priradená k inzerátu, používateľ musí byť prihlásený

Koncový bod: **/items/search/**

Povolené operácie:

- GET - Vyhľadanie konkrétnych inzerátov zodpovedajúcich zadaným parametrom dopytu povolených knižnicou `django-elasticsearch-dsl-drf`¹³. Povolené parametre na vyhľadávanie sú: `title`, `description`, `owner`, `city`, `country` a `slug`. Pričom `title` a `slug` sú zohľadnené pri vyhľadávaní viac ako napr. `description`, pre dosiahnutie vrátenia najlepších výsledkov a ich zoradenia. Tento kon-

¹³<http://django-elasticsearch-dsl-drf.readthedocs.io/>

cový bod tiež poskytuje možnosť vrátenia niekoľkých agregácií. Pomocou parametru dopytu `facet=price_stats` alebo/a zároveň `facet=price_histogram`, môžeme získať min, max, priemernú cenu alebo histogram cien v celej databáze pre konkrétny dopyt.

Koncový bod: `/items/search/?search=:text`

Povolené operácie:

- GET - Fulltextové vyhľadávanie na základe zadaného reťazca. Koncový bod vracia zoznam vecí, ktoré vyhovujú hľadanému výrazu

Koncový bod:

`/items/search/?location__geo_distance=<distance>km|<latitude>|<longitude>`

Povolené operácie:

- GET - Umožňuje vyhľadávanie inzerátov v rámci určitej vzdialenosti, na základe zadaných parametrov. Prvým parametrom je vzdialenosť v kilometroch od bodu, ktorý určujú ďalšie dva parametre (zemepisná šírka a dĺžka).

Koncový bod: `/items/search/?price__range=<price_from>|<price_to>`

Povolené operácie:

- GET - Vyhľadávanie inzerátov podľa rozsahu cien, pričom sa zadáva minimálna a maximálna cena. Koncový bod vracia pole vyhovujúcich inzerátov.

Koncový bod: `/items/search/similar/?parameter=:value`

Povolené operácie:

- GET - umožňuje vyhľadávať podobné inzeráty na základe zadaných parametrov. Povolené parametre sú: *title*, *description*, *geo_distance*. Pri použití parametra *geo_distance* je potrebné zadať hodnoty rovnako ako pri vyhľadávaní inzerátov na základe vzdialenosti, čiže: `<distance>km|<latitude>|<longitude>`

Koncový bod: `/items/categories/`

Povolené operácie:

- GET - Vypísanie listu existujúcich kategórií, parameter dopytu `parent` umožňuje filtrovanie na základe nadkategórie a tým umožňuje postupné sťahovanie stromu kategórií. Špeciálna možnosť `parent=0` vráti len kategórie najvyššej úrovne.

Koncový bod: `/items/categories/search`

Povolené operácie:

- GET - Vyhľadanie konkrétnych kategórií zodpovedajúcich zadaným parametrom dopytu povolených knižnicou `django-elasticsearch-dsl-drf`. Povolené parametre na vyhľadávanie sú: *title*, *description* a *slug*. Pričom *title* a *slug* sú zohľadnené pri vyhľadávaní viac ako napr. *description*, pre dosiahnutie vrátenia najlepších výsledkov a ich zoradenia.

Koncový bod: `/places/cities/suggest/?query=:place`

Povolené operácie:

- GET - Podľa daného *:place* odporúči existujúce mestá

7 User Experience

Pre aktuálnu verziu systému je potrebné otestovať používateľský zážitok z práce s aplikáciou. Testovanie je vhodné realizovať s piatimi participantmi. Z každého testovania je potrebné vyhotoviť audiovizuálny záznam participanta a tiež videozáznam obrazovky. Podľa možnosti aj využiť eye tracker.

Testovania neboli realizované z dôvodu, že systém ešte nie je kompletne funkčný. Pre testovanie je potrebné aby aplikácia bola funkčná. Odhalenie funkčných a technických nedostatkov alebo bugov je nutné pred používateľským testovaním.

Nasleduje pripravený scenár testovania spolu s úlohami.

7.1 Scenár používateľského testovania aplikácie iBazár

7.1.1 Poučenie

Rozhovor dizajnéra s participantom

Potrebuje otestovať použiteľnosť našej aplikácie iBazár určenú na bazárové online obchodovanie. Ide o testovanie aplikácie, nie Teba.. Ak niečo nejde podľa Tvojich očakávaní, je to problém aplikácie, nie Tvoja chyba. Je pre nás dôležité poznať Tvoj názor na všetko. Ten pozitívny aj negatívny. Hovor preto, prosím, všetko, čo Ti napadne.

Potrebuje vedieť ako chápeš aplikáciu a teda potrebuje poznať aj Tvoje myšlienkové pochody. Prosím, komentuj všetko čo vidíš, čo ideš robiť, čo by si očakával, čo Ťa prekvapilo a podobne. Každý komentár má pre nás veľkú hodnotu. Počas testovania Ti nemôžem radiť, ak nevieš pokračovať v úlohe povedz to a prejdeme k ďalšej úlohe.

Informácie, ktoré zadáš nebudú poskytnuté tretím stranám. Všetky dáta, ktoré do systému zadáš budú po testovaní zmazané. Pri tomto testovaní by ste Ťa potrebovali nahrávať pre neskoršiu analýzu. Súhlasíš? [Dať podpísať súhlas]

Máš ešte nejaké otázky?

7.1.2 Úvodný rozhovor

- Aký je Tvoj vek?
- Čo študuješ / čomu sa profesionálne venuješ? Čo je náplňou Tvojej profesie?
- Používaš internet pravidelne?
- Predával/a alebo kupoval/a si už používaný tovar na internete?
 - Aké s tým máš skúsenosti?
 - Chýbalo Ti niečo pri predaji/kúpe?
 - Nastali nejaké problémy pri obchode?

7.1.3 Úlohy

Pokračuje sa úlohami (tester dostane nasledujúcu stranu papiera s uvedenými úlohami)

V úlohách "Vyhľadávanie tovaru" Dizajnér dohliadne aby kontakt neprebehol, len sa začal (napr. otvorený koncept mailu, zadané číslo v telefóne)

Úlohy

Úvodná stránka

- Čo vidíš na tejto stránke?
- Aké je podľa Teba zameranie stránky, čo u Teba evokuje?
- Bez navigácie po stránke opíš, čo vidíš na stránke a na čo to podľa Teba slúži?

Používateľský účet

1. Vytvor si používateľský účet na portáli a prihlás sa.
2. Následne si nastav kontaktné údaje

Vyhľadávanie tovaru

1. Vyhľadaj telefón Samsung Galaxy S8 do 600 eur so zárukou minimálne pol roka a kontaktuj majiteľa.
2. Vyhľadaj byt na prenájom v Bratislave za približne 700 eur vrátane energií a zavolaj majiteľovi.

Predaj

- Zadať inzerát na predaj notebooku Lenovo ThinkPad T580.
- Z inzerátu by malo vyplývať, že:
 - notebook je možné prevziať len osobne v Košiciach,
 - má ešte jeden rok záruky a
 - predávaš ho lebo si si kúpil nový.
- Parametre notebooku sú uvedené na <https://www.lenovo-shop.sk/lenovo-thinkpad-t480-2015000bxs>
- Fotky notebooku nájdeš na pracovnej ploche.
- Po dokončení sa odhlás zo systému

Tovar predaný

- Po čase si notebook predal/a ale ľudia Ti stále volajú. Pokús sa to vyriešiť.

7.1.4 Záverečný rozhovor

- Aké sú Tvoje celkové dojmy z aplikácie?
- Ako si vnímal navigáciu po stránke? Vadilo Ti niečo?
- Chýbalo Ti niečo v aplikácii?
- Bolo v aplikácii niečo čo si nečakal/a ale bolo to vhodné?
- Vieš porovnať túto aplikáciu s inými bazárovými stránkami?
- Ako by si celkovo ohodnotil túto aplikáciu na stupnici od 1 (najlepšie) po 5?
- Chceš ešte niečo - čokoľvek dodať?

Ďakujeme pekne :)

A Príručka frontendu

Požiadavky:

- NodeJS verzia: 8.0.0+
- yarn verzia: 1.0.0+

A.1 Príprava spustenia projektu

Ako prvé je potrebné vytvoriť `.env` súbor podľa existujúceho `.env.example`. Obsahom tohoto súboru sú `key=value` hodnoty. Momentálne sa tam nachádza jedna hodnota, ktorá predstavuje webovú adresu aplikačného rozhrania.

A.2 Inštalácia

Sputením tohoto príkazu sa nainštalujú knižnice potrebné k spusteniu do priečniku `node_modules`:

```
yarn
```

A.3 Spustenie

Pred prvým sputením je potrebné spustiť nasledovný príkaz:

```
yarn build
```

Spustenie:

```
yarn start
```

B Príručka backendu

Požiadavky:

- Python verzia: 3.5+

B.1 Príprava spustenia projektu

Kroky v tejto sekcii je potrebné vykonať pred prvým spustením projektu.

B.1.1 Nainštalovanie požiadaviek projektu

Požiadavky projektu sa nainštalujú príkazom:

```
python3 -m pip install -r requirements.txt
```

Pridanie požadovaných súčastí projektu prebieha pridaním požiadavky do súboru `requirements.in`. Následne sa z tohto súboru vygeneruje súbor `requirements.txt` príkazom:

```
python3 -m pip tools compile --output-file requirements.txt requirements.in
```

B.1.2 Virtuálne prostredie

Pri práci na projekte je vhodné využívať nástroj `virtualenv`, ktorý vytvára izolované Python prostredia. `Virtualenv` sa nainštaluje príkazom:

```
python3 -m pip install virtualenv
```

`Virtualenv` pre projekt sa vytvorí príkazom:

```
python3 -m virtualenv env
```

Nakoniec sa virtuálne prostredie aktivuje príkazom:

```
source env/bin/activate
```

Po práci je možné virtuálne prostredie deaktivovať príkazom:

```
deactivate
```

B.1.3 Databáza

Pred začatím ostatných krokov potrebujete mať nainštalovanú PostgreSQL databázu:

```
sudo apt-get install postgresql-9.6
```

Následne sa vytvorí užívateľ, databáza a heslo pre užívateľa:

```
sudo -u postgres createuser <username>
```

```
sudo -u postgres createdb <dbname>
```

```
sudo -u postgres psql
```

```
psql=# alter user <username> with encrypted password '<password>';
```

Po týchto krokoch je ešte potrebné prideliť oprávnenia príkazom:

```
psql=# grant all privileges on database <dbname> to <username> ;
```

Zadané meno a heslo je následne potrebné vložiť do nastavení aplikácie.

B.1.4 Logovanie

Logovanie sa nastavuje v súbore nastavení. Existujú 4 rôzne ovládače logovania:

- `file` - vytvára logy na dennej báze s maximálnou úrovňou `DEBUG`, logy sú ukládané na 10 dní,
- `console` - vypisuje logy do konzoly s maximálnou úrovňou `DEBUG`
- `sentry` - odosiela logy s úrovňou `ERROR` do služby `Sentry.io`, na správne fungovanie musí byť nakonfigurovaná služba `Sentry.io`,
- `logentries` - odosiela logy úrovne `DEBUG` do služby `Logentries`, na správne fungovanie musí byť nakonfigurovaná služba `Logentries`.

Na používanie `Sentry.io` a `Logentries` je potrebné vložiť správny token do príslušnej premennej v súbore nastavení.

B.1.5 Mailing

Táto aplikácia potrebuje na správne fungovanie odosielať e-mailové správy. Pre použitie SMTP serveru použijete `django.core.mail.backends.smtp.EmailBackend` ako `EMAIL_BACKEND`, tiež je potrebné vyplniť ostatné SMTP nastavenia v príslušných poliach, ktoré špecifikujú užívateľa, heslo a port. V prípade vývoja je možné použiť hodnotu `django.core.mail.backends.console.EmailBackend` ako `EMAIL_BACKEND`, ktorý vypisuje e-maily priamo do konzoly.

B.1.6 Nastavenia

Nastavenia spoločné pre rôzne prostredia sú uložené v `settings/base.py`. Každé prostredie má aj svoju vlastnú špecifickú konfiguráciu, `settings/production.py` pre produkčné prostredie a `settings/development.py` pre vývojové prostredie. Tajné kľúče sa ukládajú v `.env` súboroch. šablóna súboru `.env` je uložená v `.env.example`. Použite ju na vytvorenie súborov `.env.development` pre vývojové prostredie alebo `.env.production` pre produkčné prostredie.

B.2 Prvé spustenie

Migrácia dátového modelu:

```
python3 manage.py migrate
```

Spustenie lokálneho serveru:

```
python3 manage.py runserver
```

B.3 Testovanie

Spustenie testov je možné príkazom:

```
python3 manage.py test tests
```

Vhodnejšie je ale použiť nástroj `tox`:

```
tox
```

B.4 Indexovanie do Elasticsearch

Pre zjednodušenie vývoja používame jednu vzdialenú inštaláciu Elasticsearch, aby si nemuseli ostatní členovia tímu spojzdržovať vlastné inštalácie. Táto inštalácia beží na serveri, ktorý nám bol poskytnutý.

Indexovanie je možné za predpokladu, že má vývojár všetko korektne nastavené. To znamená, že v `.env.development` má vyplnené premenné:

```
ES_HOST=https://<username>:<password>@<host>
```

```
ES_PREFIX=<prefix>_
```

Ak index ešte nie je vytvorený, tak je potrebné spustiť príkaz:

```
python3 manage.py search_index --create
```

Po spustení tohto príkazu sa vytvorí index, ktorý je v súlade s nastaveniami v aplikačnom kóde a vývojárových nastavení. Od tohto momentu, pokiaľ pribudne nejaký inzerát, tak je automaticky indexovaný a dá sa s ním ďalej operovať.

Ak existovali nejaké inzeráty ešte pred vytvorením indexu, tak aj tieto je možné dodatočne indexovať spustením príkazu:

```
python3 manage.py search_index --populate
```