

# Text Analytics Tools

- **Pregled zadatka**

Zadatak je napraviti web aplikaciju koja klijentu omogućava analizu teksta uz pomoć <https://dandelion.eu/> API-a.

Alati koje Dandelion nudi, a treba podržati su:

[Entity Extraction](#) - alat koji u tekstu pronalazi ključne entitete, kao što su mesta, poznate ličnosti, brendovi, događaji...

[Text Similarity](#) - alat koji računa koliko su dva teksta semantički slična.

[Language Detection](#) - alat koji iz unetog teksta zaključuje na kom je on jeziku.

[Sentiment Analysis](#) - alat koji radi sentimentalnu analizu teksta.

- **Scenario**

Potrebno je napraviti grafičku podršku za navedene alate koje nam pruža [dandelion.eu](https://dandelion.eu/). Kako bi mogli da koristimo Dandelion API, neophodna je registracija na [linku](#). Nakon registracije, na [dashboard](#) stranici će se nalaziti token neophodan za korišćenje API-a. Token se šalje kao query parametar pod nazivom "token".

Kako bi grafičko ogruženje moglo da šalje API pozive, potrebno je napraviti stranicu za konfiguraciju na kojoj će korisnik moći da unese/izmeni svoj token. Token je potrebno nekako sačuvati na nivou browser-a kako ne bi morali da unosimo token pri svakom otvaranju aplikacije.

Pri svakom API pozivu potrebno je koristiti token koji je prvobitno sačuvan.

Pored stranice za dodavanje/izmenu tokena, neophodno je podržati i 4 dodatne stranice u navigacionom meniju: entity extraction, text similarity, language detection, sentiment analysis. Ovim stranicama se ne sme dozvoliti pristup ako token nije postavljen.

Izgled stranica:

- **Entity extraction:** Stranica na kojoj će se nalaziti alat za pronalaženje entiteta unutar prosleđenog teksta uz opcione parametre. Na stranici je potrebno prikazati

tekstualno polje za unos teksta iz koga će se pronalaziti entiteti. Pored tekstualnog polja, potrebno je postaviti podršku za parametar “**min\_confidence**” u vidu slajdera, broja između 0 i 1. Parametar `min_confidence` predstavlja prag pouzdanosti ispod kog će mogući entiteti (sa manjom pouzdanošću) biti odbačeni. Pored toga, potrebno je podržati i opcioni parametar “**include**” kome se mogu proslediti jedna ili više vrednosti odvojenih zarezom. Moguće vrednosti su: `image`, `abstract`, `categories`. Ovo je potrebno implementirati korišćenjem checkbox-a kako bi korisnik mogao da čekira željene vrednosti. Ukoliko je korisnik čekirao neke od vrednost parametra `include`, nakon API odgovora, potrebno je prikazati dobijenu reč/reči (tj. entitet) i ispod nje prikazati rezultat za svaku od odabranih vrednosti: `image`, `abstract`, ili `categories`. Na primer, ako je vrednost `include` parametra bila *images,abstract*, nakon dobijenog odgovora, potrebno je prikazati dobijene reči, i za svaku reč prikazati sliku i vrednost polja `abstract`. Pritiskom na submit, poslati zahtev ka dandelion API-u i nakon dobijenog odgovora prikazati rezultat.

- **Text similarity:** stranica na kojoj će se nalaziti alat za poređenje teksta po semantičkoj sličnosti. Potrebno je prikazati dva tekstualna polja za unos tekstova koji će se porediti i submit dugme. Pritiskom na submit, poslati zahtev ka dandelion API-u i nakon dobijenog odgovora prikazati rezultat koliko su uneti tekstovi slični.
- **Language detection:** Stranica na kojoj se nalazi jedno tekstualno polje za unos teksta čiji jezik želimo da detektujemo. Pored teksta, potrebno je postaviti checkbox input “**clean**” - opcioni parametar za language detection poziv koji predstavlja indikator Dandelion-u da želimo da očisti tekst od linkova, mejl adresa i heš tagova pre samog procesiranja. Pritiskom na submit, poslati zahtev ka dandelion API-u i nakon dobijenog odgovora prikazati rezultat ponuđenih jezika uz confidence level koji se nalazi u odgovoru.
- **Sentiment analysis:** Stranica na kojoj se nalazi jedno tekstualno polje za unos teksta nad kojim želimo da uradimo sentimentalnu analizu. Potrebno je obezbediti podršku za opcioni parametar “**lang**” koji Dandelion prihvata za ovaj zahtev kao mogućnost za definisanje jezika koji će se koristiti za sentimentalnu analizu. Moguće opcije za jezik su: *en* - engleski, *it* - italijanski i *auto* - za automatsku detekciju jezika. Pritiskom na submit, poslati zahtev ka dandelion API-u i nakon dobijenog odgovora prikazati dobijeni skor od -1 do 1, kao i dobijeni tip (“positive”, “neutral” ili “negative”). Tip treba obojiti odgovarajućom bojom, tj. interpolirati između crvene koja označava -1 i zelene koja označava 1, gde bi neutralna boja bila braon (između crvene i zelene). [Interpolacija](#).



-1

0

1

Tokom životnog ciklusa stranice (taba), tj. dok se ne desi refresh cele stranice, neophodno je evidentirati svaki API poziv ka Dandelion API-u. Potrebno je osmisliti strukturu gde bi se čuvala vrednost svakog zahteva poslatog ka Dandelion API-u, kao i datum i vreme poziva. Stanje ove strukture treba prikazati klikom na novu stavku u meniju - **history**. Nakon refresh-a stranice, očekuje se da je ova struktura prazna.

	Primer	kako	bi	trebalo	prikazati	istoriju:
[13.10.2021.	21:41:55]	GET	<a href="https://api.dandelion.eu/datatxt/li/v1?text=Hello">https://api.dandelion.eu/datatxt/li/v1?text=Hello</a>			world&token=1234
[13.10.2021.	21:43:14]	GET	<a href="https://api.dandelion.eu/datatxt/li/v1?text=Hello">https://api.dandelion.eu/datatxt/li/v1?text=Hello</a>			RAF&token=1234
[13.10.2021.	21:50:55]	GET	<a href="https://api.dandelion.eu/datatxt/li/v1?text=Hello">https://api.dandelion.eu/datatxt/li/v1?text=Hello</a>			RG&token=1234

- **Tehnički zahtevi**

Uslov je da web aplikacija bude single page i napisana upotrebom Angular frontend framework-a. CSS biblioteke su dozvoljene i opcione - estetski izgled nije bitan.