

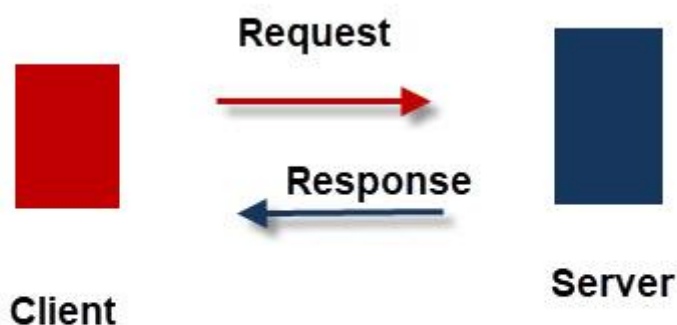
# Laboratorijska vježba br. 5: HTTP protokol i Postman alat

## 1. HTTP protokol

HTTP protokol je protokol aplikacijskog sloja OSI (*Open system interconnection model*) modela. On je osnova za razmjenu podataka na internetu.

Osobine HTTP protokola:

- baziran je na zahtjevima i odgovorima (nije konekcijskog tipa)
- ne ovisi o tipu sadržaja
- bez stanja je, tj. server i klijent se brinu samo za aktuelni zahtjev



Slika 1.1. Osnove HTTP protokola

U nastavku će biti opisani osnovni pojmovi vezani za HTTP protokol:

- **Klijent** – inicira zahtjev putem HTTP metode, na određeni URI (*Uniform Resource Identifier*) i putem protokola određene verzije šalje poruku.
- **Server** – odgovara na zahtjeve klijenata porukom. Poruka sadrži liniju koja govori o stanju odgovora (404 – Not found, 403 – Forbidden, 503 – Service unavailable), sadrži meta informacije o odgovoru i sadržaju odgovora.
- **URI** – string koji jednoznačno opisuje neki resurs.

### 1.1. Razlika između URI i URL

Često se diskutuje o razlikama između **URI** (*Universal Resource Identifier*) i **URL** (*Universal Resource Locator*) tako da će ovdje biti prikazano detaljnije objašnjenje ova dva pojma.

- I URL i URI predstavljaju načine za pronalazak nečega online.
- URI može biti samo naziv ili naziv sa protokolom.
- URL je **podtip** URI-a koji se sastoji i od resursa i od protokola.

Drugim riječima, svi URL-ovi su ujedno i URI, ali nisu svi URI ujedno i URL-ovi. Dovoljno je zapamtiti da ako je sadržan protokol, onda je riječ o URL-u (jer je URL konkretniji). Npr. *google.com* bi bio URI, jer sadrži samo naziv resursa, dok bi *https://google.com* bio URL, jer sadrži i naziv resursa i način kako doći do njega (tj. protokol). U 99% situacija koje ćete svakodnevno susretati, trebali biste koristiti termin URL koji je konkretniji od termina URI.

## 1.2. Struktura URL-a

URL ima posebnu strukturu (slika 1.2.), koja se sastoji od nekoliko komponenti:

1. **scheme** - govori o protokolu koji se koristi
2. **authority** - govori o adresi servera i portu (ali može sadržavati i informacije o korisniku). Dodatno se dijeli na *userinfo*, *host* i *port*.
3. **path** - može se sastojati od više segmenata odvojenih znakom “/”, a govori o putanji do resursa koji se zahtijeva od *host*-a.
4. **query** - daje dodatne informacije o resursu putem parametara koji se koriste u sklopu web aplikacije.
5. **fragment** - daje indirektnu identifikaciju nekog resursa u odnosu na imenovani (najčešće se odnosi na ciljani dio web stranice na koji se treba preusmjeriti).



Slika 1.2. Struktura URL-a

*Scheme* dio može uključivati: HTTP, HTTPS, FTP, MAILTO, IRC, FILE itd. HTTP i HTTPS se najčešće koriste za pristupanje resursima na internetu, ali mogu ukazivati i na lokalne resurse. *FILE scheme* se odnosi na datoteku koja se nalazi lokalno na računaru.

*Host* dio može imati i oznaku porta koja “prepisuje” zadani port za dati protokol. Na primjer:

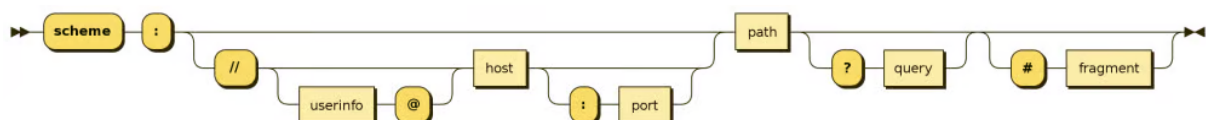
`https://google.com`

će ići na *host* koji je *google.com* i *port* 443, jer je 443 zadani port za HTTPS. Međutim, ako bi se eksplicitno specificirao broj porta:

`https://google.com:9023`

klijent će se pokušati spojiti sa portom 9023 korištenjem HTTP protokola.

Kao što je već rečeno, URL-ovi mogu imati i *query* parametre i *fragment* indetifikatore (slika 1.3.).



Slika 1.3. Detaljniji prikaz dijelova URL-a

*Query* parametri označavaju argument koji se prosljeđuje web aplikaciji. Primjer može biti funkcionalnost pretraživanja web stranice:

`https://google.com/search?s=bing`

Ovo bi obavilo pretragu za termin “bing” na *Google*-u.

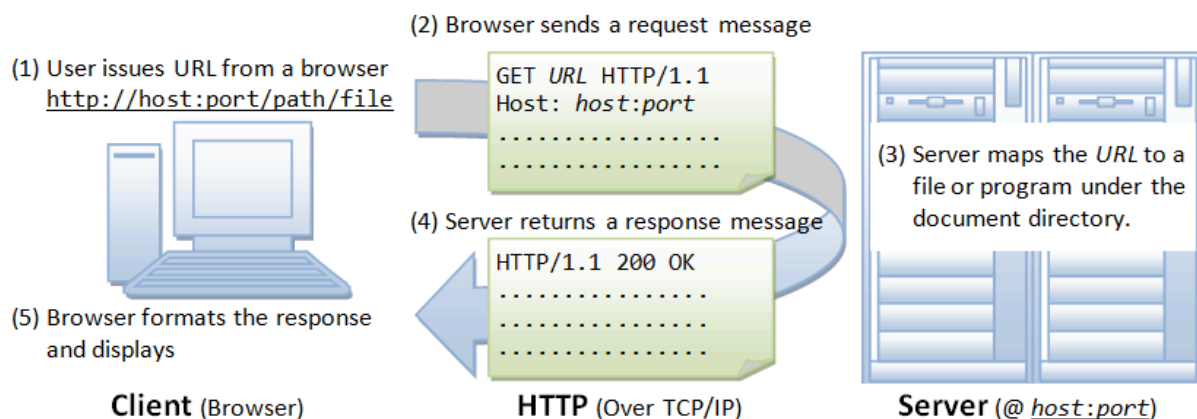
*Fragment* dio omogućava prelazak na određeni dio stranice putem URL-a, kao:

`https://google.com/results.html#worse`

Prethodni primjer bi preusmjerio aplikaciju na dio označen kao “worse” u sklopu web stranice naziva *results.html*.

### 1.3. HTTP zahtjevi

Svaki put kada iskoristite URL u web pretraživaču kako biste dobili neki resurs na web-u putem HTTP protokola (npr. `http://www.nowhere123.com/index.html`), pretraživač pretvara URL u *poruku zahtjeva* i šalje je HTTP serveru. HTTP server interpretira poruku zahtjeva i vraća vam odgovarajuću *poruku odgovora*, koja je ili resurs koji ste zatražili ili poruka o grešci. Ovaj proces je ilustriran na slici 1.4.



Slika 1.4. Tok slanja HTTP zahtjeva i primanja odgovora

HTTP zahtjev se može zamisliti kao web pretraživač koji se povezuje sa serverom i traži određeni resurs ili mu šalje podatke. Postoji nekoliko tipova HTTP metoda, koje potpuno mijenjaju vrstu odgovora koji dobivate od servera. Najčešće su:

- **GET** - Ovo je daleko najčešće korištena metoda HTTP zahtjeva. GET zahtjev od servera traži određenu informaciju ili resurs. Kada se povežete na web stranicu, web pretraživač obično šalje nekoliko GET zahtjeva za primanje podataka koji su mu potrebni za učitavanje stranice.
- **HEAD** - Sa ovim zahtjevom se primaju samo informacije o zaglavlju stranice koju želite učitati. Ovu vrstu HTTP zahtjeva možete koristiti da saznate veličinu dokumenta prije nego što ga preuzmete pomoću GET.
- **POST** - Web pretraživač koristi ovu metodu kada treba poslati podatke serveru. Na primjer, ako ispunite formu za kontakt na web stranici i pošaljete ga, koristite POST zahtjev i server prima te informacije.
- **PUT** - Ovi zahtjevi su, po funkcionalnosti, slični metodi POST. Međutim, umjesto unosa novih podataka, PUT zahtjevi se koriste za ažuriranje informacija koje već postoje na serveru.

## 1.4. Struktura HTTP zahtjeva i odgovora

Svaki HTTP zahtjev počinje linijom koja označava vrstu metode i verziju HTTP protokola koje se koriste. Na primjer, početak HTTP GET zahtjeva može izgledati ovako:

```
GET /XXX HTTP/1.1
```

U ovom primjeru, “XXX” predstavlja datoteku koju želite da vam server posluži.

Početak HTTP odgovora ponavlja verziju protokola koju koriste obje strane. Također uključuje HTTP kod koji odgovara statusu odgovora. Ako je zahtjev uspješan, dobit ćete poruku o uspjehu. Npr:

```
HTTP/1.1 200 OK
```

Ovaj dio HTTP odgovora prikazat će kodove grešaka ako se resurs ne uspije učitati iz bilo kojeg razloga. Ako server ne može pronaći resurs, vidjet ćete zaglavlje odgovora poput:

```
HTTP/1.1 400 Bad request
```

### 1.4.1. Zaglavlja zahtjeva (eng. *request headers*)

Zaglavlja zahtjeva dolaze odmah nakon redova koji se odnose na zahtjev i pružaju dodatne informacije o transakciji. Zaglavlja HTTP zahtjeva imaju sljedeći izgled:

```
Host: website.com

User-Agent: Chrome/5.0 (Windows 10)

Accept-Language: en-US

Accept-Encoding: gzip, deflate

Connection: keep-alive</code.
```

Naravno, postoji veliki broj zaglavlja koje možete pronaći u dokumentaciji a u ovoj vježbi će biti opisana samo osnovna:

- **Host** - odnosi se na IP ili URL servera kojem se šalje zahtjev.
- **User-agent** - ovaj parametar sadrži informacije o klijentu i njegovom operativnom sistemu. Obično se ovdje navodi web pretraživač koji se koristi i njegova verzija.
- **Accept-language** - govori serveru koji jezik klijent preferira, u slučaju da postoji više verzija datoteke koju tražite.
- **Accept-encoding** - označava vrstu kodiranja ili kompresije koju klijent može obraditi.

Na osnovu svega rečenog, općenita struktura koja se koristi za zahtjeve je sljedeća:

```
GET /XXX HTTP/1.1

Host: website.com

User-Agent: Chrome/5.0 (Windows 10)
```

**Accept-Language:** en-US

**Accept-Encoding:** gzip, deflate

Ovaj primjer šalje GET zahtjev na *website.com* i zahtijeva neki konkretan resurs. Zaglavlja odgovora bi mogla imati sljedeći izgled:

HTTP/1.1 200 OK

**Date:** Mon, 27 Jul 2022 12:28:53 GMT

**Server:** Apache/2.2.14 (Win32)

**Last-Modified:** Wed, 22 Jul 2022 19:15:56 GMT

**Content-Length:** 88

**Content-Type:** text/html

### 1.4.2. Tijelo poruke

Tijelo poruke sadrži podatke koje ili šaljete ili primete, zavisno od metode zahtjeva koju koristite. Na primjer, ako ste tražili HTML datoteku pomoću metode GET, mogli biste dobiti odgovor koji je strukturiran na sljedeći način:

HTTP/1.1 200 OK

**Date:** Mon, 27 Jul 2022 12:28:53 GMT

**Server:** Apache/2.2.14 (Win32)

**Last-Modified:** Wed, 22 Jul 2022 19:15:56 GMT

**Content-Length:** 88

**Content-Type:** text/html

**Connection:** Closed

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"></code.

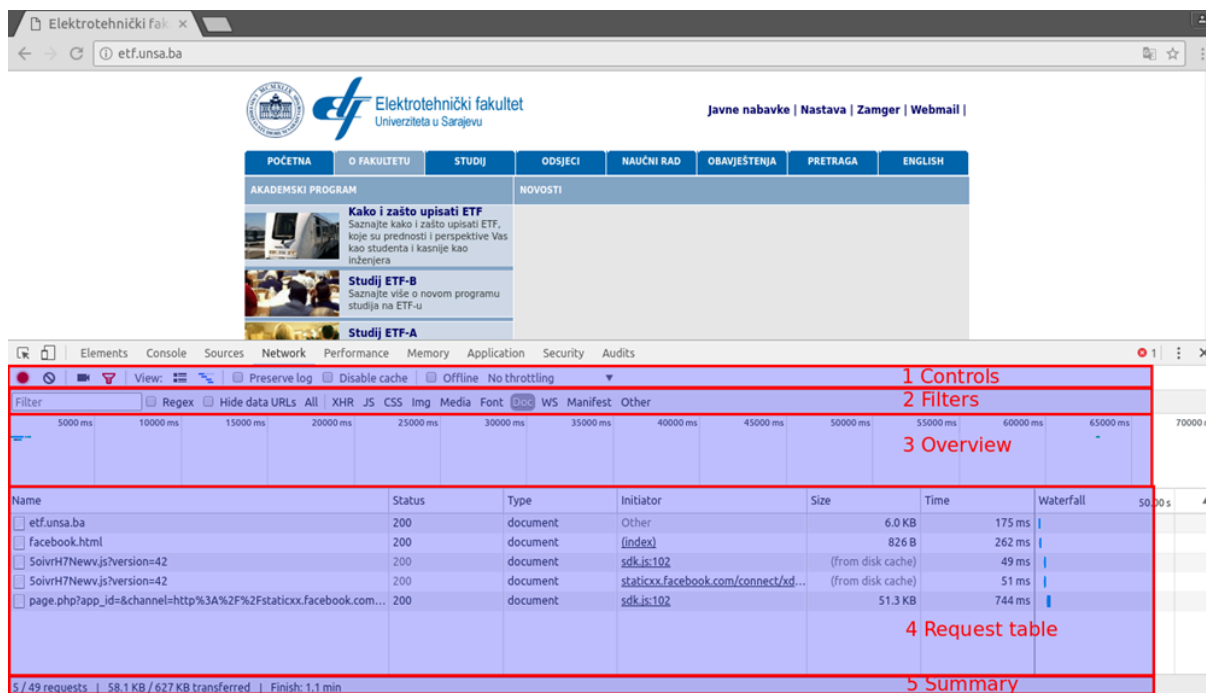
U primjeru nije prikazana cijela datoteka, jer bi onda odgovor bio glomazan a suština je jasna i na osnovu početnih linija HTML koda.

Naravno, i prilikom slanja zahtjeva je moguće navesti njegovo tijelo. U ovom slučaju je izuzetno bitno zaglavlje **Content-Type**. Ovo zaglavlje se koristi za označavanje izvorne vrste medija resursa (prije bilo kakvog kodiranja sadržaja tokom pripreme za slanje). U zahtjevima (kao što su POST i PUT), klijent govori serveru koji je stvarni tip podataka koje šalje. Ovo zaglavlje se može koristiti i u odgovorima, gdje klijentu daje do znanja koji je stvarni tip sadržaja koji je vraćen sa servera.

Content-Type: text/html; charset=utf-8

## 1.5. Snimanje zahtjeva klijenta i odgovora od servera

Postoji više alata putem kojih je moguće napraviti zahtjeve i snimiti ih, ali će se u ovoj vježbi koristiti *developer tools* koji je integriran u *Chrome* web pretraživaču. *Developer tools* otvaramo pritiskom na dugme *F12* na tastaturi ili putem menija u gornjem desnom dijelu web pretraživača (nakon klika na opciju *Customize and control* koja je obično označena sa tri tačke ili tri ravne crte, birajte *More tools -> Developer tools*). Za snimanje zahtjeva/odgovora se koristi *Networks* kartica.



Slika 1.5. Snimanje zahtjeva/odgovora korištenjem *Networks* kartice u sklopu *Developer tools* web pretraživača

Kartica *Networks* se sastoji od nekoliko dijelova (slika 1.5.):

1. *Controls* – podešavanje izgleda *Networks* kartice
2. *Filters* – odabir uslova za filtriranje zahtjeva koji će se prikazati u *request* tabeli
3. *Overview* – vremenska linija koja sadrži dužine trajanja obrade zahtjeva, kao i redoslijed slanja zahtjeva/pristizanja odgovora
4. *Request table* – lista zahtjeva koji ispunjavaju uslov/e u *Filters* dijelu. U tabeli se nalazi naziv zahtjeva, status, tip sadržaja, veličina, vrijeme obrade zahtjeva itd.
5. *Summary* – ukupan broj zahtjeva koji su upućeni.

Klikom na pojedine zahtjeve iz *request* tabele, otvara se dio koji daje detaljnije informacije o zaglavljima zahtjeva i odgovora kao i tijelu zahtjeva.

## 1.6. Primjer zahtjeva i odgovora

Neka su zadani sljedeći:

### HTTP zahtjev

```
GET /api/products/123 HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0
Safari/537.36
Accept: application/json
```

### HTTP odgovor

```
HTTP/1.1 200 OK
Date: Mon, 02 Jan 2023 12:00:00 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 345
{
  "id": 123,
  "name": "Product X",
  "price": 49.99
}
```

1. Na osnovu zadanog HTTP zahtjeva, kojem resursu klijent pokušava pristupiti i koji je traženi format podataka?
2. U HTTP odgovoru, koji je statusni kod i što on govori o uspjehu zahtjeva?
3. Šta označava zaglavlje *Content-Type* i ukratko objasnite njegov značaj.

### Odgovori:

1. Klijent pokušava pristupiti resursu na putanji `/api/products/123` na serveru. Traženi format podataka naveden je u zaglavlju `"Accept"` zahtjeva, koje je postavljeno na `"application/json"`. Ovo znači da klijent očekuje da podaci o odgovoru budu u JSON formatu.
2. HTTP odgovor ima statusni kod `200 OK`. Statusni kod `200` označava da je zahtjev bio uspješan. Konkretno, `200 OK` znači da je server uspješno obradio zahtjev, a odgovor sadrži tražene podatke. U ovom slučaju, to sugerira da je server pružio informacije o proizvodu s *id*-em `123` u JSON formatu.
3. *Content-Type* zaglavlje navodi tip podataka odgovora. U ovom slučaju postavljeno je na `application/json charset=utf-8`, što ukazuje da tijelo odgovora sadrži JSON podatke kodirane u UTF-8. Ovo zaglavlje je ključno jer govori klijentu kako interpretirati podatke odgovora.

**Zadatak 1.** Snimite zahtjev i odgovor koji se šalje kada otvorite stranicu [etf.unsa.ba](http://etf.unsa.ba) u web pretraživaču. Napišite izvještaj i odgovorite na sljedeća pitanja:

- a) Kojom HTTP metodom se šalje zahtjev?
- b) Koliko je vremena trebalo da dođe prvi, a koliko za posljednji odgovor od servera?
- c) Snimite jedan od zahtjeva i odgovora za dobavljanje slike! Koja je veličina tijela odgovora? Nabrojite koja se sve zaglavlja nalaze u zahtjevu, a koja u odgovoru i opišite njihovo značenje na konkretnom primjeru.
- d) Pronađite zaglavlje u kojem piše koji web pretraživač koristite.
- e) Koliko je ukupno zahtjeva upućeno?



## 2. Postman alat

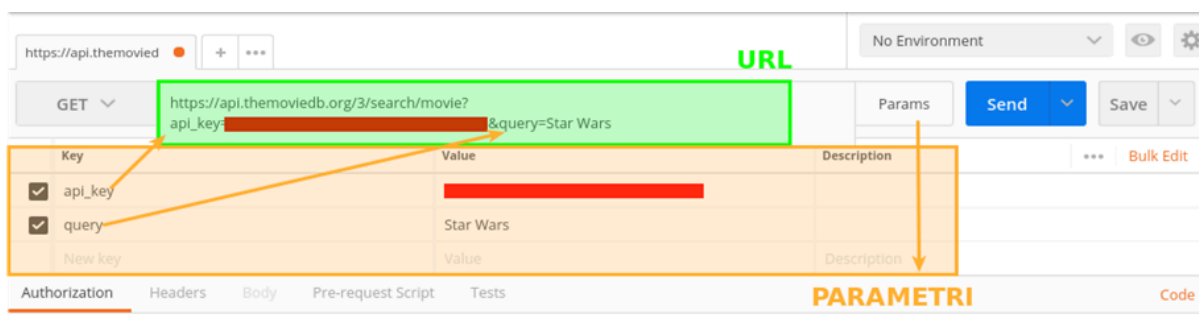
*Postman* je alat koji omogućava jednostavno pravljenje i slanje HTTP zahtjeva i iščitavanje odgovora od servera. Posjeduje niz opcija koje omogućavaju testiranje metoda web servisa. Omogućava ponovno slanje zahtjeva, kao i pravljenje skripti koje će poslati više zahtjeva automatski kako bi se olakšalo testiranje web servisa. Alat se može preuzeti na lokalni računar a može se koristiti i online verzija alata. Obje varijante se mogu pronaći na [linku](#).

### 3. Zadaci

Zadatke označene **zelenom bojom** je potrebno uraditi u toku laboratorijske vježbe. Ukoliko studenti tokom laboratorijske vježbe ne urade te zadatke, onda kući (**najkasnije 8 sati** prije početka naredne laboratorijske vježbe) moraju uraditi i zadatke označene **zelenom bojom** i zadatke označene **narandžastom bojom**. Bez obzira da li se zadaci rade na vježbi ili kod kuće, **obavezno** ih je postaviti na odgovarajući *Bitbucket* repozitorij za vježbe.

**Zadatak 2.** Putem Postman alata uputite jedan GET zahtjev na neki od API-a koje pronađete u dokumentaciji [The Movie DB](#) i zabilježite odgovor od servera. Pratite sljedeće korake:

1. Registrujte se na stranici API-a kako biste dobili ključ za korištenje web servisa (*API key*)
2. U dokumentaciji API-a, u dijelu za pretragu filmova, prepoznajte koji je URL i koji su parametri potrebni za upućivanje jednog GET zahtjeva.



3. Uputite zahtjev i snimite koji je odgovor. Iz odgovora iščitajte:
  - a. zaglavlja - pronađite šta svaki od njih znači
  - b. statusni kod
  - c. vrijeme
  - d. tip sadržaja odgovora

History

Collections

Clear all

▼ Today

GET https://api.themoviedb.org/3/search/movie?api\_key=a1ae4918896c5e634c61d597d471e7fd&query=Star Wars

GET https://api.themoviedb.org/3/search/movie?api\_key=&quer... Params Send Save

Authorization Headers Body Pre-request Script Tests Code

Type No Auth

ODGOVOR

Status: 200 OK Time: 623ms

Body Cookies Headers (13) Test Results

Pretty Raw Preview JSON

```
1 {
2   "page": 1,
3   "total_results": 117,
4   "total_pages": 6,
5   "results": [
6     {
7       "vote_count": 6758,
8       "id": 11,
9       "video": false,
10      "vote_average": 8.1,
11      "title": "Star Wars",
12      "popularity": 42.896122,
13      "poster_path": "/btIDwqjvOlOFFlp1sPuZiZoQC6.jpg",
14      "original_language": "en",
15      "original_title": "Star Wars",
16      "genre_ids": [
17        12,
18        28,
19        878
20      ],
21      "backdrop_path": "/4LjFYyQzZcONB9HNzg0J0wMyPH.jpg",
22      "adult": false,
23      "overview": "Princess Leia is captured and held hostage by the evil Imperial forces in their effort to take over the galactic Empire. Venturesome Luke Skywalker and dashing captain Han Solo team together with the loveable robot duo R2-D2 and C-3PO to rescue the beautiful princess and restore peace and justice in the Empire. ",
24      "release_date": "1977-05-25"
25    },
26    {
27      "vote_count": 7965,
28      "id": 140607,
```