

Apple assignment  
Pratik Jogdand

Goals: First goal is to come up with a Ruby on Rails application that allows its users to access two distinct reports in the realm of web stats using a REST API:

1. Number of page views per URL, grouped by day, for the past 5 days;
2. Top 5 referrers for the top 10 URLs grouped by day, for the past 5 days.

Steps:

- 1) Installed Ruby, Rails and PostgreSQL on a 2019 Macbook Pro.
- 2) Configure rvm locally.

```
rvm which ruby  
/Users/pratikjogdand/.rvm/versions/3.3.6/bin/ruby
```

- 3) Create a new ruby on rails application

```
rails new . --database=postgresql --api
```

- 4) Create these files in the ruby on rails application apart from all the default files generated.

```
|— Gemfile  
|— app/  
|   |— models/  
|   |   |— page_view.rb  
|   |— controllers/  
|   |   |— api/  
|   |   |   |— v1/  
|   |   |       |— stats_controller.rb  
|— db/  
|   |— migrate/  
|   |   |— 20240116000000_create_page_views.rb  
|— lib/  
|   |— tasks/  
|   |   |— generate_data.rake # Create with provided rake task  
|— spec/  
|   |— models/  
|   |   |— page_view_spec.rb # Create with provided spec  
|   |— requests/  
|   |   |— api/  
|   |   |   |— v1/  
|   |   |       |— stats_spec.rb # Create with provided spec
```

- 5) Install dependencies by running: bundle install

- 6) Setup the database and run migrations.
  - a) rails db:create
  - b) rails db:migrate
  - c) Rails db:generate # This generates a million database entries
- 7) Write Code to meet goals, code package included in the zip file.
- 8) Run Unit Tests: bundle exec rspec --format documentation

```
(base) → apple_assignment git:(main) ✖ bundle exec rspec --format documentation

PageView
  is valid with valid attributes
  is invalid without a URL
  is invalid without a created_at
  is invalid without a digest
  generates a digest before validation

Stats API
  GET /api/v1/top_urls
    returns page views per URL for the past 5 days
  GET /api/v1/top_referrers
    returns top referrers for top URLs in the past 5 days

Finished in 0.12098 seconds (files took 1.49 seconds to load)
7 examples, 0 failures
```

- 9) Manual testing to test both endpoints

- a) [GET] /top\_urls

Only using top 3 to display or it prints around a million results from the database

```
curl -X GET http://localhost:3000/api/v1/top_urls | jq 'to_entries |
map({date: .key, top_urls: .value | sort_by(.visits) | reverse | .[:3]})'
```

You can use curl -X GET [http://localhost:3000/api/v1/top\\_urls](http://localhost:3000/api/v1/top_urls) to get a result of number of page views per URL, grouped by day, for the past 5 days

- b) [GET] /top\_referrers

To access this you can run

```
curl http://localhost:3000/api/v1/top_referrers
```

OUTPUTS:

[GET] /top\_urls

```
{
  "date": "2024-11-16",
  "top_urls": [
    {
      "url": "http://developer.apple.com",
      "visits": 4876
    },
    {
      "url": "http://en.wikipedia.org",
      "visits": 4861
    },
    {
      "url": "https://apple.com",
      "visits": 4857
    }
  ]
},
{
  "date": "2024-11-15",
  "top_urls": [
    {
      "url": "http://en.wikipedia.org",
      "visits": 4951
    },
    {
      "url": "http://developer.apple.com",
      "visits": 4888
    },
    {
      "url": "https://apple.com",
      "visits": 4833
    }
  ]
},
{
  "date": "2024-11-14",
```

[GET] /top\_referrers

```
"2024-11-16": {
  "http://apple.com": {
    "visits": 4817,
    "referrers": [
      {
        "url": "http://apple.com",
        "visits": 24
      },
      {
        "url": "https://www.apple.com",
        "visits": 24
      },
      {
        "url": "http://developer.apple.com",
        "visits": 24
      },
      {
        "url": "https://apple.com",
        "visits": 24
      },
      {
        "url": "http://okon.example/eli",
        "visits": 2
      }
    ]
  },
  "https://apple.com": {
```

To test the rails app under load I have used apollo benchmarking suite

# Test top URLs endpoint under load (100 requests, 10 concurrent)

ab -n 100 -c 10 http://localhost:3000/api/v1/top\_urls

# Test top referrers endpoint under load (100 requests, 10 concurrent)

ab -n 100 -c 10 [http://localhost:3000/api/v1/top\\_referrers](http://localhost:3000/api/v1/top_referrers)



```
Server Software:
Server Hostname:      localhost
Server Port:          3000

Document Path:        /api/v1/top_urls
Document Length:      19506805 bytes

Concurrency Level:    10
Time taken for tests:  238.160 seconds
Complete requests:    100
Failed requests:      0
Total transferred:    1950744000 bytes
HTML transferred:     1950680500 bytes
Requests per second:  0.42 [#/sec] (mean)
Time per request:     23815.992 [ms] (mean)
Time per request:     2381.599 [ms] (mean, across all concurrent requests)
Transfer rate:        7998.93 [Kbytes/sec] received

Connection Times (ms)
              min    mean[+/-sd] median   max
Connect:        0      0    0.0      0      0
Processing:    2646  22503  4277.6  23725  25647
Waiting:       2639  22469  4313.7  23718  24958
Total:         2647  22503  4277.6  23725  25648
```

```
Server Software:
Server Hostname:      localhost
Server Port:          3000

Document Path:        /api/v1/top_referrers
Document Length:      9197 bytes

Concurrency Level:    10
Time taken for tests:  10.161 seconds
Complete requests:    100
Failed requests:      0
Total transferred:    982314 bytes
HTML transferred:     919700 bytes
Requests per second:  9.84 [#/sec] (mean)
Time per request:     1016.126 [ms] (mean)
Time per request:     101.613 [ms] (mean, across all concurrent requests)
Transfer rate:        94.41 [Kbytes/sec] received

Connection Times (ms)
              min    mean[+/-sd] median   max
Connect:        0      0    0.0      0      0
Processing:     21    162  938.7      69   9455
Waiting:        21    162  938.7      69   9455
Total:          21    162  938.8      70   9455
```