Performance Matrices Implementation Analysis

Objective: Design a generic HTTP API endpoint which is capable of filtering, grouping and sorting.

Application Name: Performance Matrices Application
BaseUrl: http://127.0.0.1:8000/
Endpoint: /AdjustDataApi/dataSearch/

Technology:
Python - 3.8.1
Django - 3.0.4
Postgresql - PostgreSQL 12
Swagger - rest_framework_swagger

Configuration:
DB Setting: In settings.py file, please configure your database in DATABASES section. I have used PostgreSql. In my case like:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'performancematrics',
        'USER': 'postgres',
        'PASSWORD': '********',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

Migration DB: Please run the belowing script for migrating database
python manage.py makemigrations adjustdataapp
python manage.py migrate

Initialize DB: Please run the bellowing script for migrating database
python manage.py data_upload_to_db

Is everything well configured, then run the server bellowing command
Run the server: python manage.py runserver

How to set Api Param?
from_date : datatype: string, fomat:yyyy-mm-dd. Example: from_date: 2017-01-30
and to_date: datatype: string, fomat:yyyy-mm-dd. Example: from_date: 2017-01-31
channel: datatype: string. query parameter case sensitive Example:adcolony
country: datatype:string. query parameter case sensitive Example: US

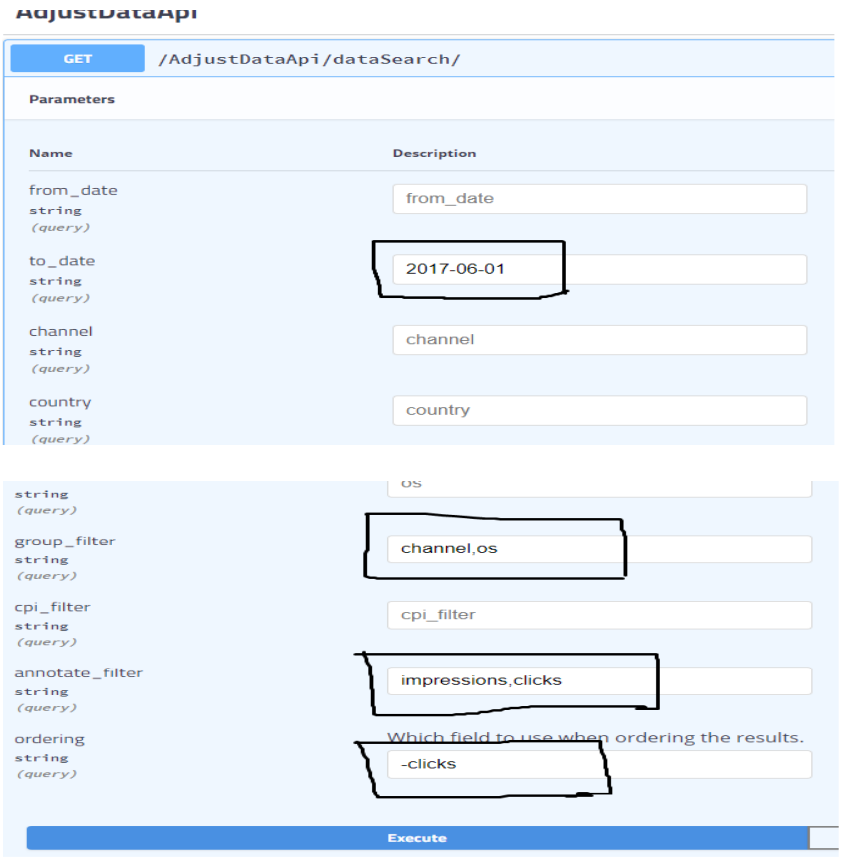
os: datatype:string. query parameter case sensitive Example:ios
group_filter: datatype:string. Multi value accepted, multi param should be comma separated, Example:channel,country .
cpi_filter: datatype: bool, if you set the value true, then it calculates cpi where cpi = spend/installs
annonate_filter: datatype:string. Return sum of fields one or more(impressions, clicks, spend, revenue,installs). Accept multi value which should be comma separated. Example: impressions,installs
ordering(asc/desc): datatype: string. If you set the param with prefix '-' then it calculates desc, otherwise asc. Exmple: ordering = -clicks it is calculate desc and ordering = clicks it is calculates asc.

Api Purpose:
Client of this API should be able to:

1. filter by time range (date_from / date_to is enough), channels, countries, operating systems
2. group by one or more columns: date, channel, country, operating system
3. sort by any column in ascending or descending order
4. see derived metric CPI (cost per install) which is calculated as cpi = spend / installs

Common API use-cases: I have shown api calling and response result with image attach

Case 1: Show the number of impressions and clicks that occurred before the 1st of June 2017, broken down by channel and country, sorted by clicks in descending order.



Request URL

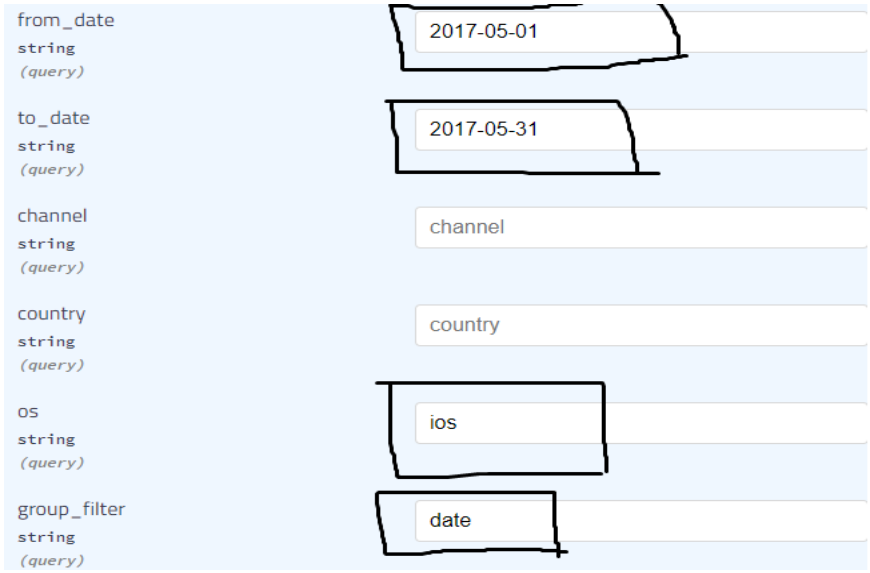
```
http://127.0.0.1:8000/AdjustDataApi/dataSearch/?to_date=2017-06-01&group_filter=channel%2Cos&annotate_filter=impressions%2Cclicks&ordering=-clicks
```

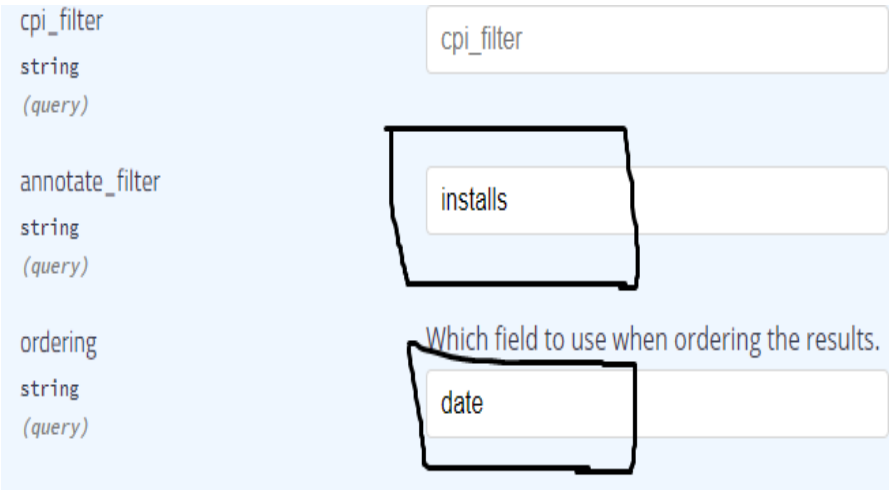
Server response

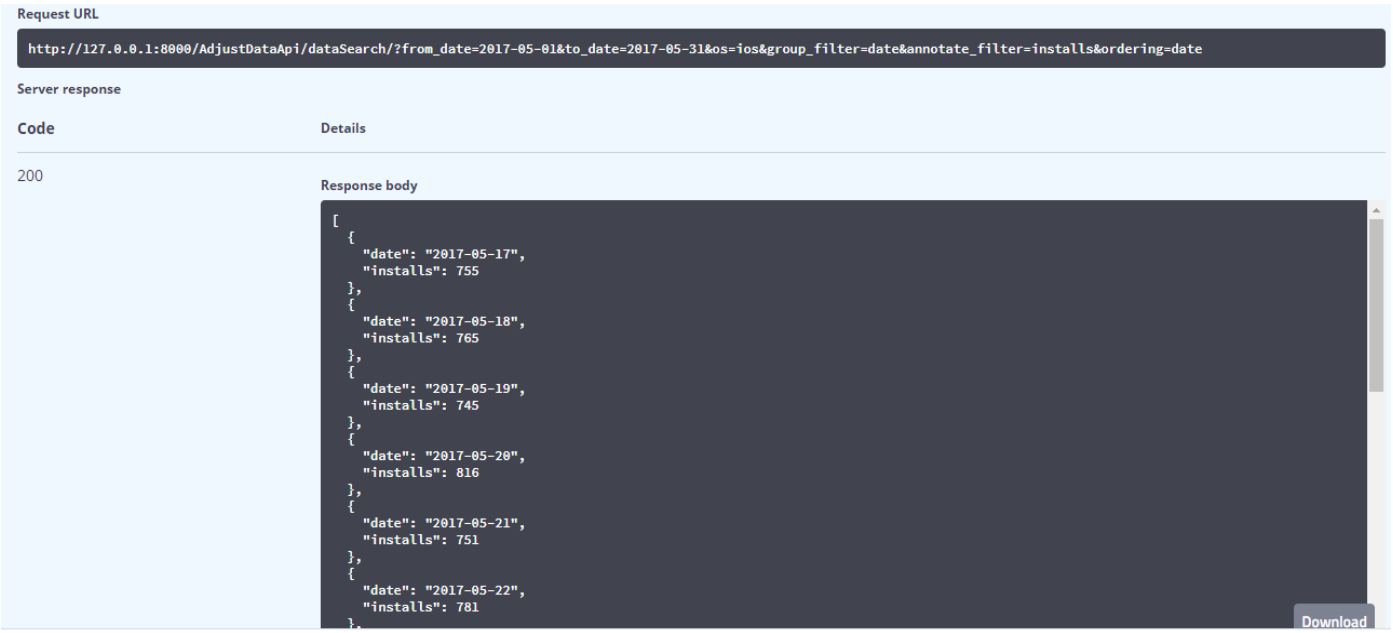
Code	Details
200	Response body


```
[
  {
    "channel": "apple_search_ads",
    "os": "ios",
    "impressions": 529186,
    "clicks": 17254
  },
  {
    "channel": "unityads",
    "os": "android",
    "impressions": 296859,
    "clicks": 9532
  },
  {
    "channel": "vungle",
    "os": "android",
    "impressions": 277143,
    "clicks": 9096
  },
  {
    "channel": "google",
    "os": "android",
    "impressions": 274930,
    "clicks": 8447
```


Case 2: Show the number of installs that occurred in May of 2017 on iOS, broken down by date, sorted by date in ascending order.







Case 3: Show revenue, earned on June 1, 2017 in US, broken down by operating system and sorted by revenue in descending order.

from_date
string
(query)

2017-06-01

to_date
string
(query)

2017-06-01

channel
string
(query)

channel

country
string
(query)

US

os
string
(query)

os

group_filter
string
(query)

os

cpi_filter

cpi_filter

cpi_filter
string
(query)

cpi_filter

annotate_filter
string
(query)

revenue

ordering
string
(query)

Which field to use when ordering the results.

-revenue

Request URL

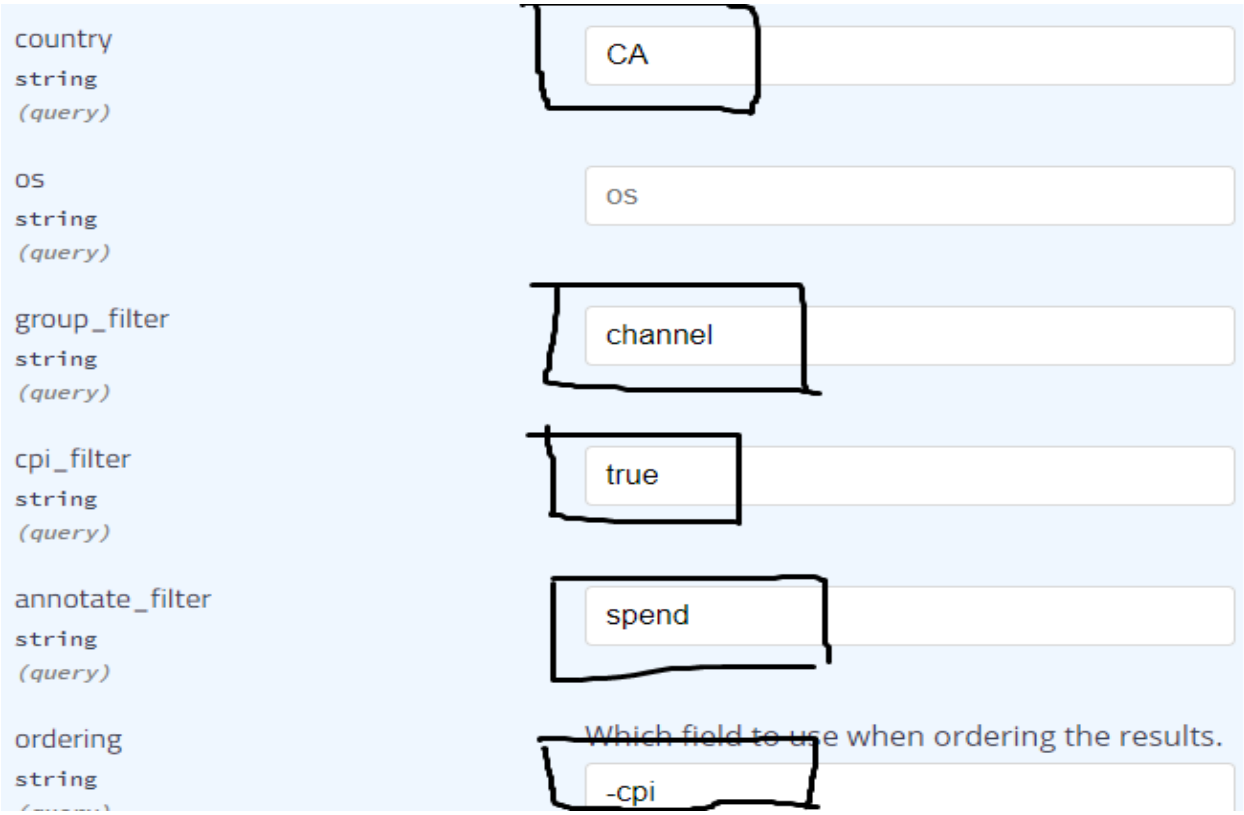
```
http://127.0.0.1:8000/AdjustDataApi/dataSearch/?from_date=2017-06-01&to_date=2017-06-01&country=US&group_filter=os&annotate_filter=revenue&ordering=-revenue
```

Server response

Code	Details
200	**Response body**

```
[
  {
    "os": "android",
    "revenue": "1205.21"
  },
  {
    "os": "ios",
    "revenue": "398.87"
  }
]
```

Case 4: Show CPI and spend for Canada (CA) broken down by channel ordered by CPI in descending order. Please think carefully which an appropriate aggregate function is for CP



Request URL

```
http://127.0.0.1:8000/AdjustDataApi/dataSearch/?country=CA&group_filter=channel&cpi_filter=true&annotate_filter=spend&ordering=-cpi
```

Server response

Code	Details
200	Response body

```
[
  {
    "channel": "facebook",
    "spend": "1164.00",
    "cpi": 2.0748663101604277
  },
  {
    "channel": "chartboost",
    "spend": "1274.00",
    "cpi": 2
  },
  {
    "channel": "unityads",
    "spend": "2642.00",
    "cpi": 2
  },
  {
    "channel": "google",
    "spend": "999.90",
    "cpi": 1.74198606271777
  }
]
```

4 Cases URL:

Case 1: http://127.0.0.1:8000/AdjustDataApi/dataSearch/?to_date=2017-06-01&group_filter=channel%2Cos&annotate_filter=impressions%2Cclicks&ordering=-clicks

Case 2: http://127.0.0.1:8000/AdjustDataApi/dataSearch/?from_date=2017-05-01&to_date=2017-05-31&os=ios&group_filter=date&annotate_filter=installs&ordering=date

Case 3: http://127.0.0.1:8000/AdjustDataApi/dataSearch/?from_date=2017-06-01&to_date=2017-06-01&country=US&group_filter=os&annotate_filter=revenue&ordering=-revenue

Case 4:
http://127.0.0.1:8000/AdjustDataApi/dataSearch/?country=CA&group_filter=channel&cpi_filter=true&annotate_filter=spend&ordering=-cpi