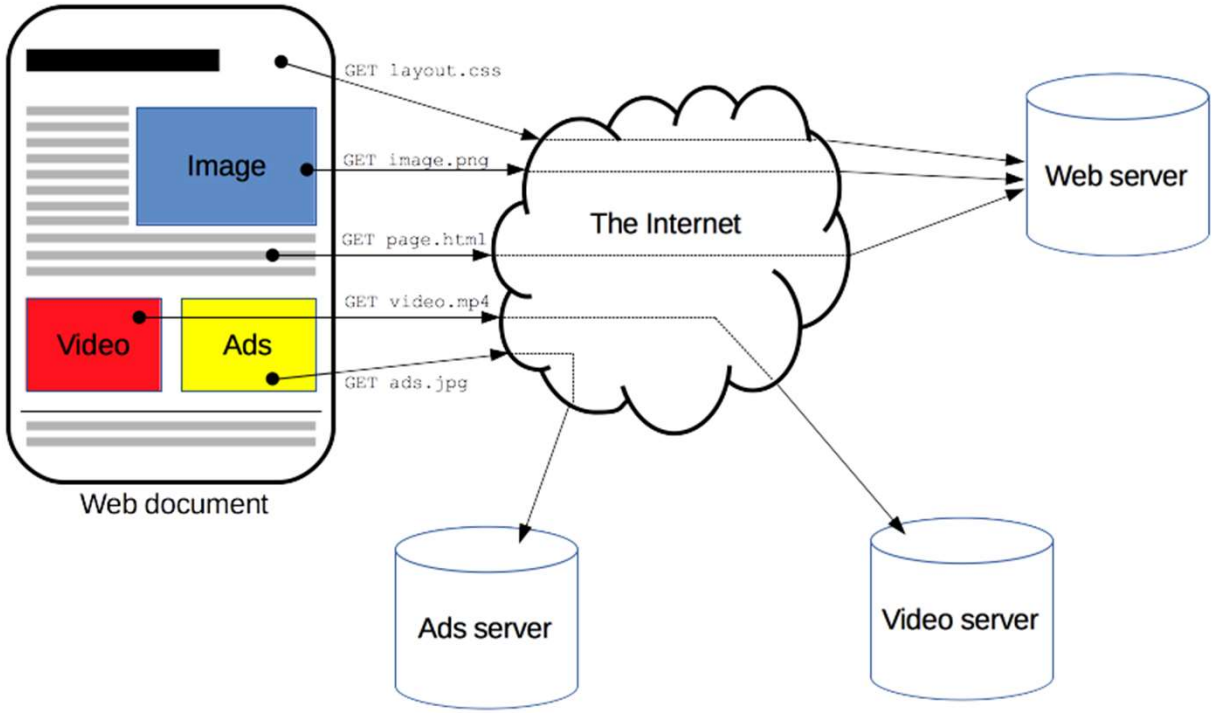
46. Основы работы с API

Цель:

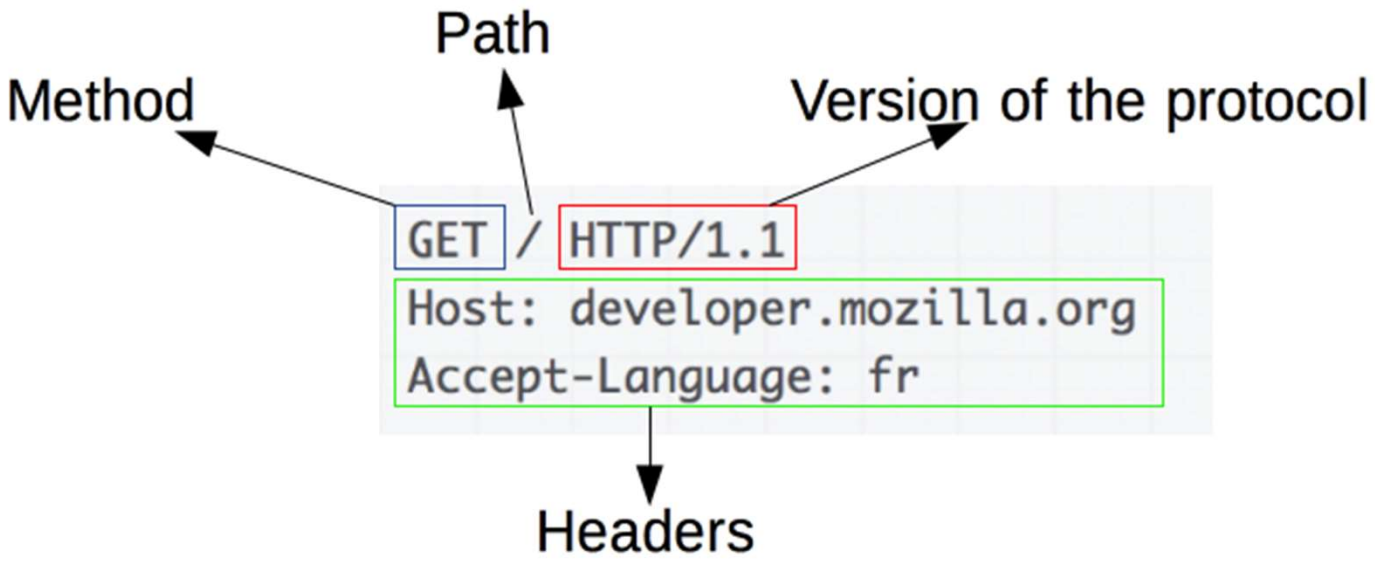
Познакомиться со следующими основами работы с API:

- обзор протокола HTTP
- HTTP запросы

HTTP:


GET layout.css
Image
GET image.png
GET page.html
GET video.mp4
Video
Ads
GET ads.jpg
Web document
The Internet
Web server
Ads server
Video server

HTTP запросы:

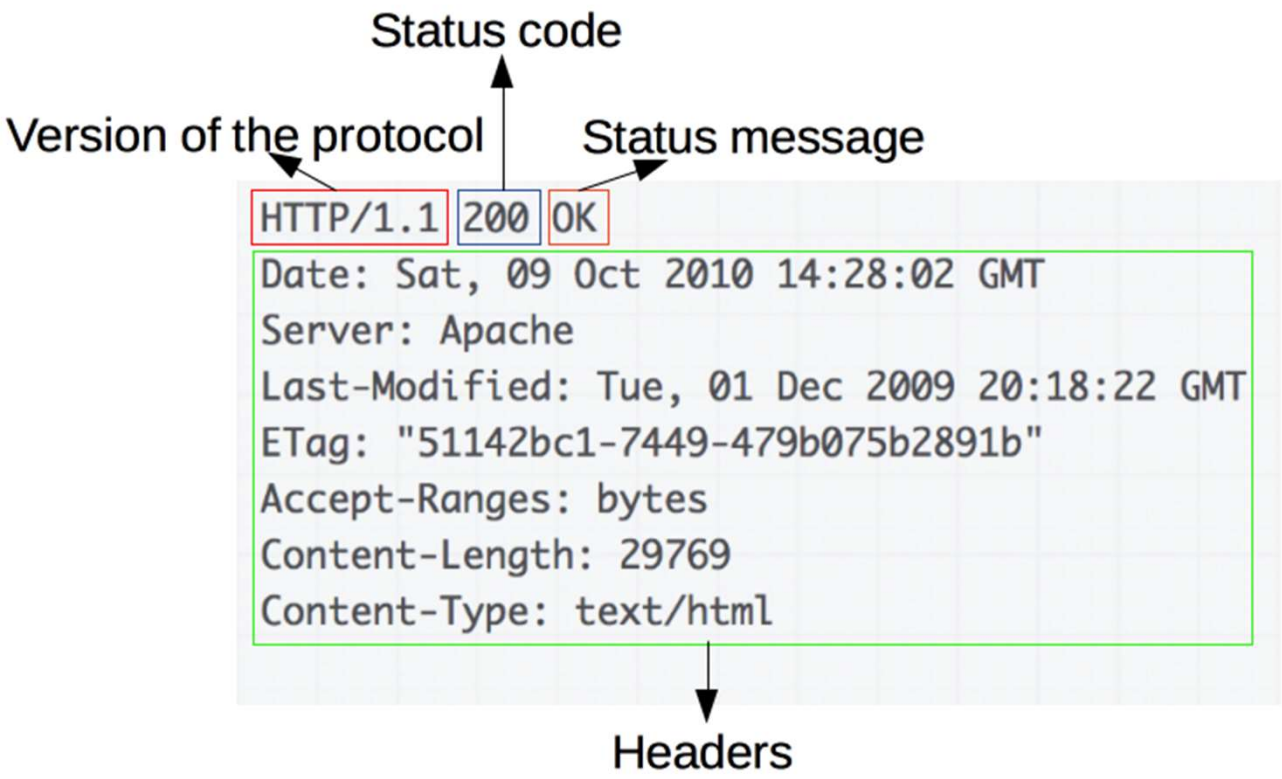


HTTP запросы:

Запросы содержат следующие элементы:

- HTTP-метод
- Путь к ресурсу: (например developer.mozilla.org)
- Версию HTTP-протокола.
- Заголовки (опционально), предоставляющие дополнительную информацию для сервера.
- Тело, для некоторых методов, таких как POST, которое содержит отправленные данные.

HTTP ответы:


Status code
Version of the protocol
Status message
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html
Headers

HTTP ответы:

Ответы содержат следующие элементы:

- Версию HTTP-протокола.
- HTTP код состояния, сообщающий об успешности запроса или причине неудачи.
- Сообщение состояния — краткое описание кода состояния.
- HTTP заголовки, подобно заголовкам в запросах.
- Опционально: тело, содержащее пересылаемый ресурс.

HTTP методы:

- **GET:** получить доступ к существующему ресурсу. В URL перечислена вся необходимая информация, чтобы сервер смог найти и вернуть в качестве ответа искомый ресурс.
- **POST:** используется для создания нового ресурса. POST запрос обычно содержит в себе всю нужную информацию для создания нового ресурса.

HTTP методы:

- **PUT:** обновить текущий ресурс. PUT запрос содержит обновляемые данные.
- **DELETE:** служит для удаления существующего ресурса.
- **HEAD:** аналогичен GET. Разница в том, что при данном виде запроса не передаётся сообщение. Сервер получает только заголовки. Используется, к примеру, для того чтобы определить, был ли изменен ресурс.

HTTP методы:

- **TRACE:** во время передачи запрос проходит через множество точек доступа и прокси серверов, каждый из которых вносит свою информацию: IP, DNS. С помощью данного метода, можно увидеть всю промежуточную информацию.
- **OPTIONS:** используется для определения возможностей сервера, его параметров и конфигурации для конкретного ресурса.

HTTP headers:

- **Accept-Language:** en-us,en;q=0.5
- **User-Agent:** Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5 (.NET CLR 3.5.30729)

  Этот заголовок может содержать несколько частей информации, таких как:
  - Имя и версия браузера.
  - Название и версия операционной системы.
  - Язык по умолчанию.
- **Cookie:** PHPSESSID=r2t5690jko5r4q7ib3vtdjq120; foo=bar
- **Authorization:** Basic bXl8jk00yOm15cGFzcw==

Коды ответа HTTP:

1. Информационные **100 - 199**
2. Успешные **200 - 299**
3. Перенаправления **300 - 399**
   Своеобразное сообщение клиенту о необходимости совершить ещё одно действие. Самый распространённый вариант применения: перенаправить клиент на другой адрес.
4. Клиентские ошибки **400 - 499**
   Данный класс сообщений используется сервером, если он решил, что запрос был отправлен с ошибкой.
5. Серверные ошибки **500 - 599**

Коды ответа HTTP. Примеры:

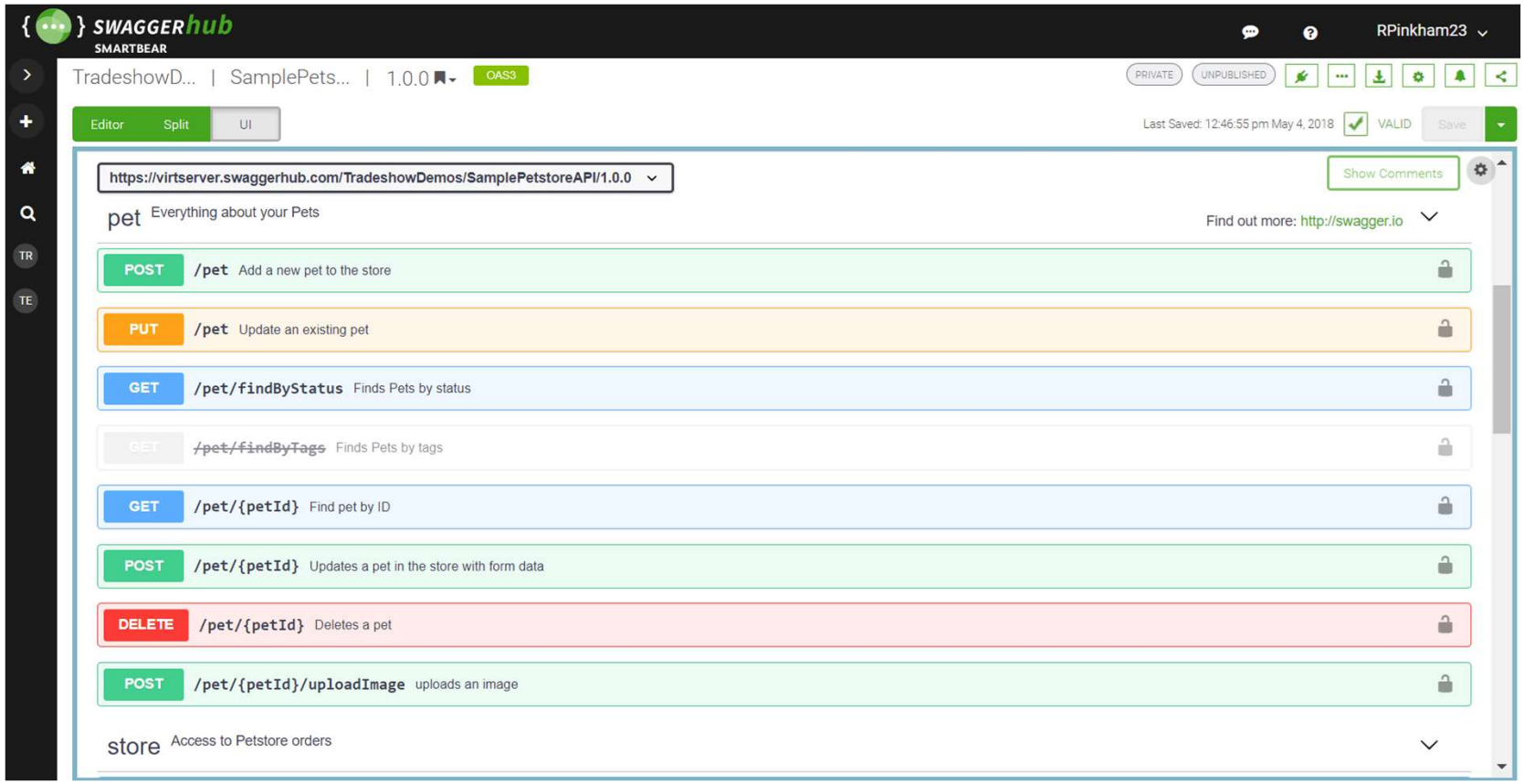
- **200 OK**
- **204 No Content:** в теле ответа нет сообщения.
- **400 Bad Request**: вопрос был сформирован неверно.
- **401 Unauthorized**: для совершения запроса нужна аутентификация. Информация передается через заголовок Authorization.
- **403 Forbidden**: сервер не открыл доступ к ресурсу.
- **404 Not Found:** означает, что ресурс не найден на сервере.
- **503 Service Unavailable**: это может случиться, если на сервере произошла ошибка или он перегружен. Обычно в этом случае, сервер не отвечает, а время, данное на ответ, истекает.

HTTP/HTTPS:

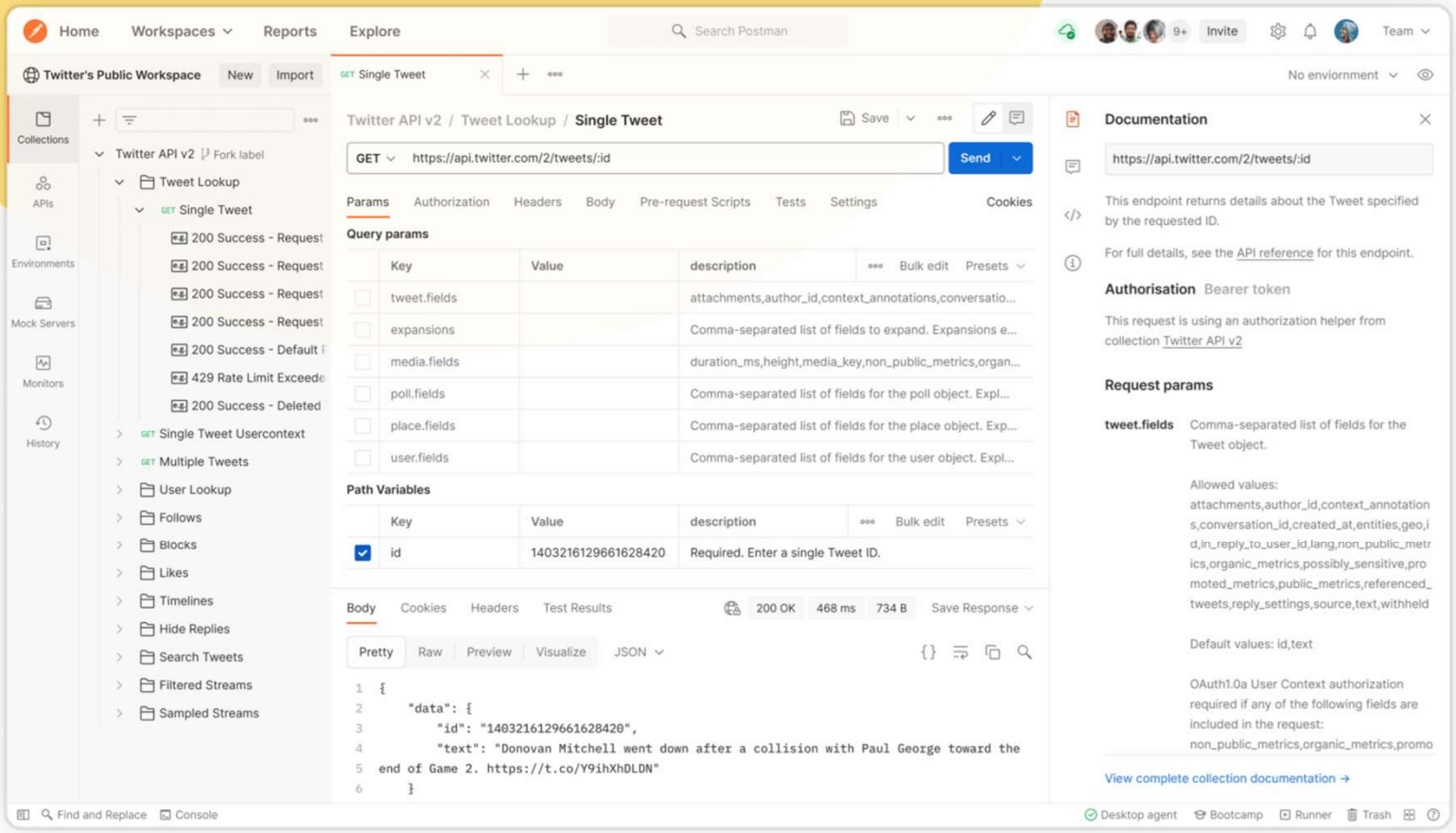
- HTTPS не является отдельным протоколом передачи данных, а представляет собой расширение протокола HTTP с надстройкой шифрования;
- передаваемые по протоколу HTTP данные не защищены, HTTPS обеспечивает конфиденциальность информации путем ее шифрования;
- HTTP использует порт 80, HTTPS — порт 443.

Postman & Swagger:

Swagger:


SWAGGERhub
SMARTBEAR
RPinkham23
TradeshowD... | SamplePets... | 1.0.0
OAS3
PRIVATE
UNPUBLISHED
Editor
Split
UI
Last Saved: 12:46:55 pm May 4, 2018
VALID
Save
https://virtserver.swaggerhub.com/TradeshowDemos/SamplePetstoreAPI/1.0.0
Show Comments
pet
Everything about your Pets
Find out more: http://swagger.io
POST /pet Add a new pet to the store
PUT /pet Update an existing pet
GET /pet/findByStatus Finds Pets by status
GET /pet/findByTags Finds Pets by tags
GET /pet/{petId} Find pet by ID
POST /pet/{petId} Updates a pet in the store with form data
DELETE /pet/{petId} Deletes a pet
POST /pet/{petId}/uploadImage uploads an image
store
Access to Petstore orders

Postman:


Home
Workspaces
Reports
Explore
Search Postman
Invite
Team
Twitter's Public Workspace
New
Import
Single Tweet
No enviornment
Collections
APIs
Environments
Mock Servers
Monitors
History
Twitter API v2
Fork label
Tweet Lookup
Single Tweet
200 Success - Request
200 Success - Request
200 Success - Request
200 Success - Request
200 Success - Default
429 Rate Limit Exceede
200 Success - Deleted
Single Tweet Usercontext
Multiple Tweets
User Lookup
Follows
Blocks
Likes
Timelines
Hide Replies
Search Tweets
Filtered Streams
Sampled Streams
Twitter API v2 / Tweet Lookup / Single Tweet
Save
GET
https://api.twitter.com/2/tweets/:id
Send
Params
Authorization
Headers
Body
Pre-request Scripts
Tests
Settings
Cookies
Query params
Key
Value
description
Bulk edit
Presets
tweet.fields
attachments,author_id,context_annotations,conversatio...
expansions
Comma-separated list of fields to expand. Expansions e...
media.fields
duration_ms,height,media_key,non_public_metrics,organ...
poll.fields
Comma-separated list of fields for the poll object. Expl...
place.fields
Comma-separated list of fields for the place object. Exp...
user.fields
Comma-separated list of fields for the user object. Expl...
Path Variables
id
1403216129661628420
Required. Enter a single Tweet ID.
Body
Cookies
Headers
Test Results
200 OK
468 ms
734 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"data": {
"id": "1403216129661628420",
"text": "Donovan Mitchell went down after a collision with Paul George toward the end of Game 2. https://t.co/Y9ihXhDLDN"
}
Documentation
https://api.twitter.com/2/tweets/:id
This endpoint returns details about the Tweet specified by the requested ID.
For full details, see the API reference for this endpoint.
Authorisation
Bearer token
This request is using an authorization helper from collection Twitter API v2
Request params
tweet.fields
Comma-separated list of fields for the Tweet object.
Allowed values: attachments,author_id,context_annotations,conversation_id,created_at,entities,geo,id,in_reply_to_user_id,lang,non_public_metrics,organic_metrics,possibly_sensitive,promoted_metrics,public_metrics,referenced_tweets,reply_settings,source,text,withheld
Default values: id,text
OAuth1.0a User Context authorization required if any of the following fields are included in the request: non_public_metrics,organic_metrics,promo
View complete collection documentation →
Find and Replace
Console
Desktop agent
Bootcamp
Runner
Trash