









High-level Design of Twitter

Understand the high-level design of the Twitter service.

We'll cover the following

- ^
- User-system interaction
- API design
 - Post Tweet
 - Like or dislike Tweet
 - Reply to Tweet
 - Search Tweet
 - Response
 - View home_timeline
 - Follow the account
 - Retweet a Tweet

User-system interaction

Let's begin with the high-level design of our Twitter system. We'll initially highlight and discuss the building blocks, as well as other components, in the context of the Twitter problem briefly. Later on, we'll dive deep into a few components in this chapter.





Twitter components



- **Users** post Tweets delivered to the server through the load balancer. Then, the system stores it in persistent storage.
- **DNS** provides the specified IP address to the end user to start communication with the requested service.
- **CDN** is situated near the users to provide requested data with low latency. When users search for a specified term or tag, the system first searches in the CDN proxy servers containing the most frequently requested content.
- **Load balancer** chooses the operational application server based on traffic load on the available servers and the user requests.
- **Storage system** represents the various types of storage (SQL-based and NoSQL-based) in the above illustration. We'll discuss significant storage systems later in this chapter.
- Application servers provide various services and have business logic to orchestrate between different components to meet our functional requirements.

We have detailed chapters on DNS, CDN, specified storage systems (Databases, Key-value store, Blob store), and Load balancers in our building blocks section. We'll focus on further details specific to the Twitter service in the coming lessons. Let's first understand the service API.

API design

This section will focus on designing various APIs regarding the functionalities we are providing. We learn how users request various services through APIs. We'll only concentrate on the significant parameters of the APIs that are relevant to our design. Although the front-end server c call another API or add more parameters in the API received from the end

users to fulfill the given request, we consider all relevant arguments specified for the particular request in a single API. Let's develop APIs for each of the following features:



- Post Tweet
- Like or dislike Tweet
- Reply to Tweet
- Search Tweet
- View user or home timeline
- Follow or unfollow the account
- Retweet a Tweet

Post Tweet

The POST method is used to send the Tweet to the server from the user through the /postTweet API.



Let's discuss a few of the parameters:

Parameter	Description
user_id	It indicates the unique ID of the user who posted the Tweet.
access_type	It tells us whether the Tweet is protected (that is, only visible to follo public.
tweet_type	It indicates whether the Tweet is text-based, video-clip based, im ? consisting of different types.
content	It specifies the Tweet's actual content (text).
tweet_length	It represents the text length in the Tweet. In the case of video, it duration and size of a video.

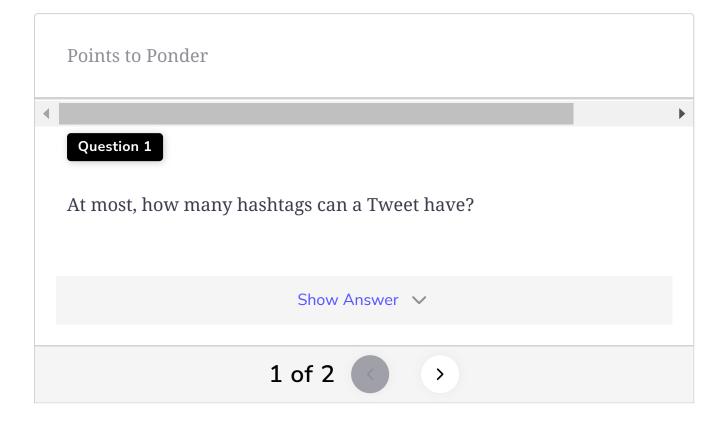
media_field

It specifies the type of media (image, video, GIF, and so on) delivered Tweet.



The rest of the parameters are self-explanatory.

Note: Twitter uses the **Snowflake** service to generate unique IDs for Tweets. We have a detailed chapter (Sequencer) that explains this service.

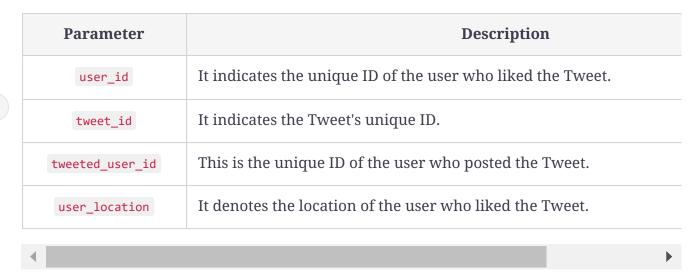


Like or dislike Tweet

The /likeTweet API is used when users like public Tweets.







The parameters above are also used in the <code>/dislikeTweet</code> API when users dislike others' Tweets.

Reply to Tweet

The /replyTweet API is used when users reply to public Tweets.

The reply type and reply length parameters are the same as tweet type and

Search Tweet

tweet_length respectively.

When the user searches any keyword in the home timeline, the GET method is used. The following is the /searchTweet API:

searchTweet(user_id, search_term, max_result, exclude, media_field, expansions, so
rt_order, next_token, user_location)

Some new parameters introduced in this case are:

Parameter Description



	riight lotal Boolgit at Times. Clothering mouthin bytesin Boolgit lines from the Engineers a managero
search_term	It is a string containing the search keyword or phrase.
max_result	It is the number of Tweets returned per response page. By default, to response is 10.
exclude	It specifies what to exclude from the returned Tweets, that is, replies The maximum limit on returned Tweets is 3200, but when we exclude maximum limit is reduced to 800 Tweets.
media_field	It specifies the media (image, video, GIF) delivered in each returned
expansions	It enables us to request additional data objects in the returned Twee mentioned user, referenced Tweet, attached media, attached places on.
sort_order	It specifies the order in which Tweets are returned. By default, it wirecent Tweets first.
next_token	It is used to get the next page of results. For instance, if max_result is Tweets, and the result set contains 200 Tweets, then the value of nex pulled from the response to request the next page containing the fol Tweets. The last result (page) will not have a next_token.

Response

Let's look at a sample response in JSON format. The **id** is the user's unique ID who posted the Tweet and the **text** is the Tweet's content. The **result_count** is the count of the returned Tweet, which we set in the max_result in the request. Here, we're displaying the default fields only.

்்ு Click to see response in JSON



 $T_{T_{i}}$



>

Note: Twitter performs various types of searches. The following are two of them:

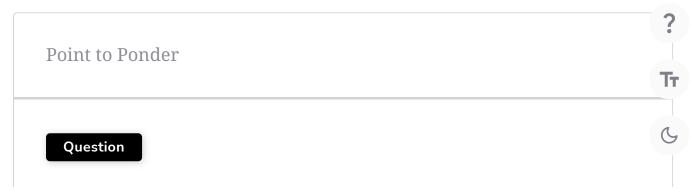
- One search type returns the result of the last seven days, which all registered users usually use.
- The other type returns all matching results on all Tweets ever posted (remind that service does not delete a posted Tweet). Indeed, matches can contain the first Tweet on Twitter. This search is usually used for academic research.

View home_timeline

The GET method is suitable when users view their home timelines through the /viewHome timeline API.

In the /viewUser_timeline API, we'll exclude the user_location to get the user timeline.

The max_rsult parameter determines the number of tweets a client application can show the user. The server sends the max_result number of tweets in each response. Further, the server will also send a paginated list_of_followers to reduce the client latency.

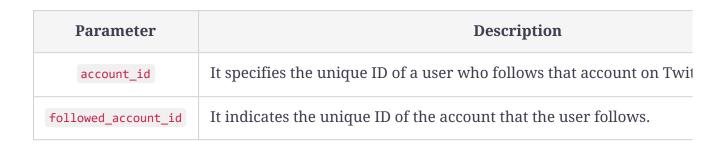


Which parameter in the viewHome_timeline method is the most relevant when deciding which promoted ads (Tweets) to be returned in response?

Show Answer V

Follow the account

The /followAccount API is used when users follow someone's account on Twitter.



The /unfollowAccount API will use the same parameters when a user unfollows someone's account on Twitter.

Retweet a Tweet

When a registered user Retweets (re-posts) someone's Tweet on Twitter, th ? following /retweet API is called:

The same parameters will be required in the /undoRetweet API when users undo a Retweet of someone's Tweet.





Next \rightarrow

Requirements of Twitter's Design

Detailed Design of Twitter



✓ Mark as Completed





