



Sharding & IDs at Instagram



Instagram Engineering · Follow

Published in Instagram Engineering



Sign up to discover human stories that deepen your understanding of the world.

Free

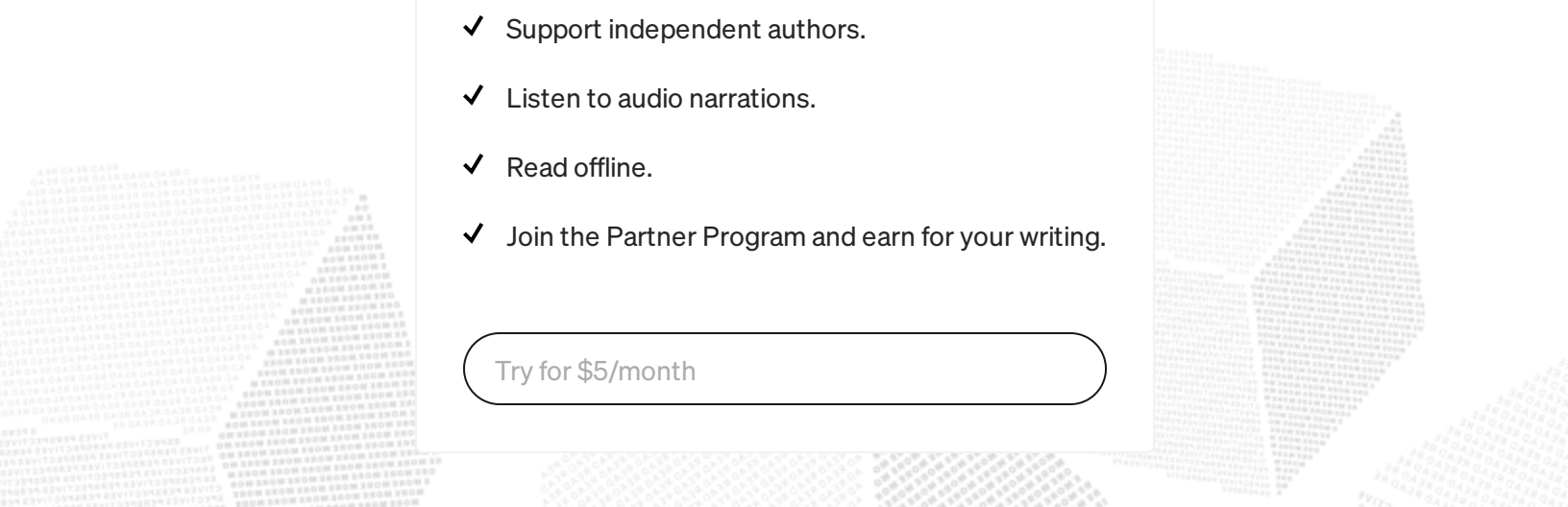
- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month



to-understand solutions that we trust.

Existing solutions

Many existing solutions to the ID generation problem exist; here are a few we considered:

Generate IDs in web application

This approach leaves ID generation entirely up to your application, and not up to the database at all. For example, MongoDB's ObjectId, which is 12 bytes long and encodes the timestamp as the first component. Another popular approach is to use UUIDs.



Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month

Uses the database's auto-incrementing abilities to enforce uniqueness. [Flickr uses this approach](#), but with two ticket DBs (one on odd numbers, the other on even) to avoid a single point of failure.

Pros:

1. DBs are well understood and have pretty predictable scaling factors

Cons:



Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month

- 41 bits for time in milliseconds (gives us 41 years of IDs with a custom epoch)
- 13 bits that represent the logical shard ID
- 10 bits that represent an auto-incrementing sequence, modulus 1024. This means we can generate 1024 IDs, per shard, per millisecond

Let's walk through an example: let's say it's September 9th, 2011, at 5:00pm and our 'epoch' begins on January 1st, 2011. There have been 1387263000 milliseconds since the beginning of



Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month

```
now_millis bigint;  
shard_id int := 5;  
BEGIN  
SELECT nextval('insta5.table_id_seq') %% 1024 INTO seq_id;  
  
SELECT FLOOR(EXTRACT(EPOCH FROM clock_timestamp()) * 1000) INTO  
now_millis;  
result := (now_millis - our_epoch) << 23;  
result := result | (shard_id <<10);  
result := result | (seq_id);  
END;  
$$ LANGUAGE PLPGSQL;
```



Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month



Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month





Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month



Static Analysis at Scale: An Instagram Story

How Instagram develops and uses linting and codemod tools based on LibCST to maintain a modern codebase at scale.

13 min read · Aug 15, 2019

👏 1.6K 💬 5



Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month



Recommended from Medium



Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month





Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month





Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month





Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month



Multi-Threading: Concurrency and Synchronization

A discussion on concurrency in multithreaded applications.

8 min read · 5 days ago

 161  5



Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month

