

**PUBLICLY SHARED**

# pub.dev outage 2021-03-24

**Author:** jonasfj@, sigurdm@  
**Status:** final  
**Created:** 2021-03-30

Failure to ignore a 'quota exceeded issue' in a non-essential YouTube API integration caused an outage. This impacted the pub.dev public website, and the API used by the dart pub client.

**Service:** pub.dev

**Date-time:** 2021-03-25 01:50 UTC

**Duration:** ~3 hours

**Impact:** an estimate ~27k unique IPs is estimated to have experienced issues with the website or API.

## Timeline

- 2021-03-25 00:35 UTC: Email alert for quota exceeded error was sent by monitoring.
- 2021-03-25 00:43 UTC: Email alert for "quota exceeded error" was reviewed
  - Identified as originating from YouTube integration
  - Ignored because:
    - YouTube integration was designed to be non-essential, and
    - Service appeared to be unaffected.
  - Responding engineer goes to bed.
- 2021-03-25 01:10 UTC: Emails warning of new error in stack driver error reporting ticks in.
- 2021-03-25 01:10 UTC: 5xx errors starts spiking in load-balancer for pub.dev
- 2021-03-25 01:10 UTC: Alerts for API latency fires off (nobody notices).
- **2021-03-25 01:50 UTC: [pub.dev is down](#), website, APIs, all of it.**
- 2021-03-25 02:01 UTC: Engineers notice the issue in multiple chat rooms and start debugging.
- 2021-03-25 02:19 UTC: [dart-lang/pub-dev#4663](#) is filed.
- 2021-03-25 02:58 UTC: Attempts to reach main engineers working on this service fail, as they are sleeping.
- 2021-03-25 03:39 UTC: Traffic roll back to a previous, one week old deployment of pub.dev is initiated
- 2021-03-25 03:41 UTC: 100% of traffic is migrated to previous deployment of pub.dev
  - This does not resolve the situation, because YouTube integration had been introduced more than 7 days earlier .
- 2021-03-25 04:04 UTC: It is identified that:
  - All backend instances to be unhealthy in cloud console.
  - GCLB will not send traffic to unhealthy instances.
- 2021-03-25 04:04 UTC: Instances are identified to be restarting every ten minutes
- 2021-03-25 04:18 UTC: Started looking through the log explorer with all logs turned on.
- 2021-03-25 04:26 UTC: It is identified that we are hitting the quota limit for YouTube.
- 2021-03-25 04:37 UTC: It is identified that global top-level exceptions are accidentally logged as INFO.

- Source: An issue that should be addressed in the latest version of [package:appengine](#).
- 2021-03-25 04:39 UTC: Problem is identified: [app/lib/service/youtube/backend.dart#L35](#)
  - If this fails, nothing catches the exception, it bubbles up and restarts the server.
- 2021-03-25 04:45 UTC: YouTube API quota is increased.
- **2021-03-25 04:51 UTC: Reports that pub.dev is now alive-ish**
- 2021-03-25 11:45 UTC: A new version of pub.dev without Youtube integration is deployed.

## Impact

- 27k unique IPs estimated to have been impacted, based on requests in the same timespan on a similar day 1 week earlier.

## Root Cause

A recent deployment had introduced a new feature for displaying relevant Dart & Flutter videos at the bottom of the pub.dev landing page. The implementation of this used the YouTube API for getting a list of videos. When a non-critical YouTube API request to list videos threw an exception, pub.dev didn't log this exception with a warning. Instead the error propagated the global error handler for the serving isolate, which restarted the serving isolate. In turn this meant that issues with the YouTube API could take down pub.dev.

When implementing this, we did introduce retries around the API calls, but we overlooked that the place that initiated these calls would crash the isolate, if those errors propagated. Thus, we should have wrapped the retry logic with a catch-all that logged these non-critical errors, instead of allowing them to propagate. Going forward, we should probably establish better patterns for how to run non-critical background tasks, as we have done for some periodic background tasks already.

## Lessons Learned

### Things That Went Well

- We attempted to switch traffic back to the previous version of pub.dev, as outlined in [playbook](#).
  - In this instance this could not resolve the issue, as the YouTube integration wasn't new.
- We got an alert that the quota was exceeded.
  - Unfortunately, we then promptly decided to ignore this alert, as we believed this was a non-critical failure.

### Where we got lucky

- Engineers responding were able to find someone with access to the cloud project.
- The quota error logged as INFO was discovered.
- We were able to request a Youtube API quota increase through YouTube on-calls.

## What didn't work

- Response and coordination was rather improvised.
  - Engineers tasked with covering this service in other timezones had not been adequately trained, and training had not been maintained.
- We were unable to reach primary engineers working on this service.
- Exceptions thrown outside the context of a request were logged as INFO in older versions of package:appengine.
- Switching traffic to a previous version of pub.dev was not sufficient to resolve the issue.
- We did not notice that we were close to exceeding the daily YouTube API quota limit every day for several days prior to the outage.

## Action items

Immediate remediation:

- Disable YouTube API usage, until a proper fix is identified.
- Reduce YouTube API usage

Failure prevention:

- Better isolation of background tasks: Design a pattern for writing non-critical background tasks in pub.dev.
- Upgrade to the latest version of package:appengine, where we already fixed issues with log messages being dropped.
- Make pub.dev easily deployable without a running pub.dev

Process:

- Update pub-playbook troubleshooting section to reference dashboards
- Configure pager for primary engineers owning this service.
- Configure alerts for being *close* to quota limits.
- Formalize incident response capabilities across time zones.

## Future Mitigation

In addition to improving our handling of outages, we must invest in larger and more long term architectural changes to reduce the risk of outages occurring in the first place.

- Switch to serving the API necessary to do "pub get" directly from cloud storage via a CDN.
- Migrate to Cloud Run for faster deployment.