Batching

send multiple messages with a single request

benefits

- increased throughput
  less overhead for HTTP headers, more efficient use of threads and connections

- decreased costs
  the fewer requests we make, the less money we pay for using services with a pay-per-request pricing model

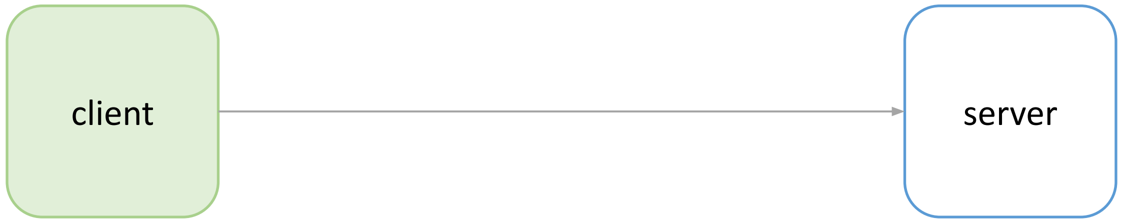
- less chance of being throttled
  services often protect themselves from being abused by clients that send too many requests in a short period of time

Batching

batching introduces complexity on both the client and server side

accumulate messages in a buffer and flush the buffer based on time or size

messages are processed one by one and partial failures are possible



can be hard to implement and configure correctly

handling partial failures can be tricky

Batching

how the server handles batch requests

- treat the entire request as a single atomic unit
  request succeeds only when all nested operations complete successfully

this approach is more common in practice

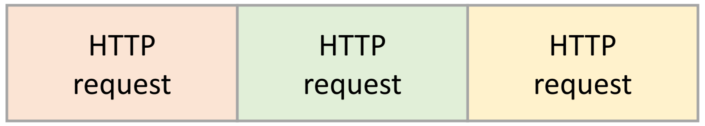
- treat each nested operation independently and report back failures for each individual operation
  service processing the request tries to make as much progress as possible

Batching

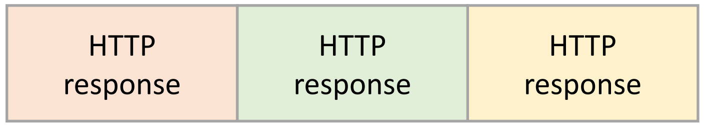
batch request format

set of *n* **requests** batched together

standard HTTP request



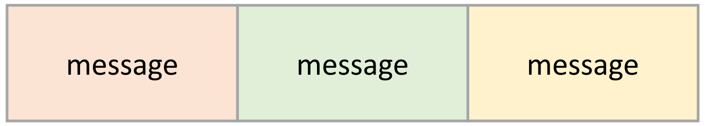
standard HTTP response



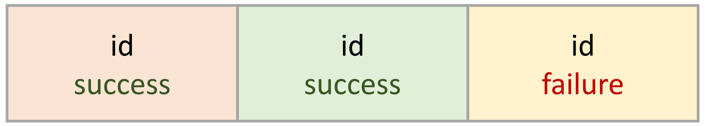
example
Google Drive batch API
https://developers.google.com/drive/api/v3/batch

list of *n* **resources** batched together

standard HTTP request



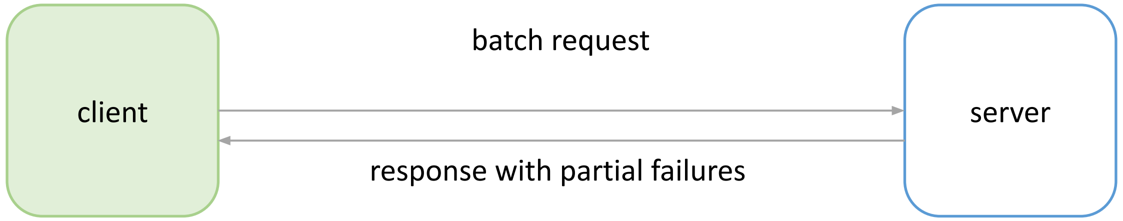
standard HTTP response



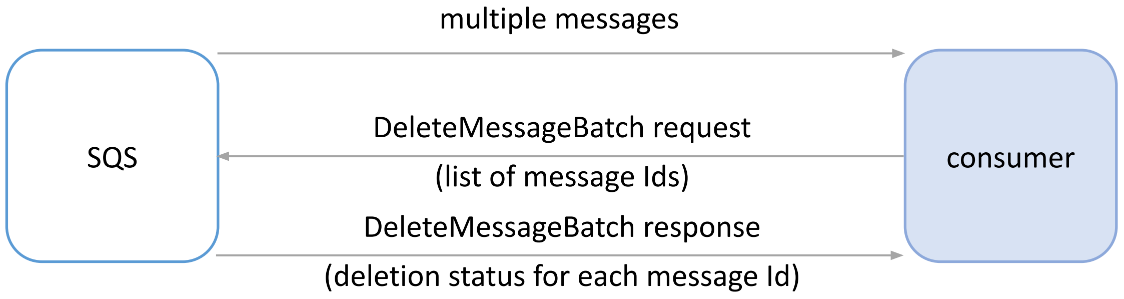
example
AWS SQS batch API
https://docs.aws.amazon.com/AWSSimpleQueueService /latest/APIReference/API_SendMessageBatch.html

Batching

how the client handles failed nested operations



- retry the entire batch request
- retry each failed operation individually
- create another batch request containing only failed individual operations


multiple messages
SQS
consumer
DeleteMessageBatch request
(list of message Ids)
DeleteMessageBatch response
(deletion status for each message Id)