Guide: Email fallback

Email fallback is a feature that notifies conversation participants of new messages via email if receipts do not read the new messages within a certain amount of time after the messages are sent. This document describes how to build this functionality into your application.

Setup

1. Create an account with an email sender, such as SendGrid or Mailchimp, and note necessary API keys.
2. Ensure your application server can validate Layer webhooks.
3. Register a Webhook for `message.sent` and `message.read` events.
4. Ensure your system or application has a scheduling system to execute code at some point in the future.

Implementation

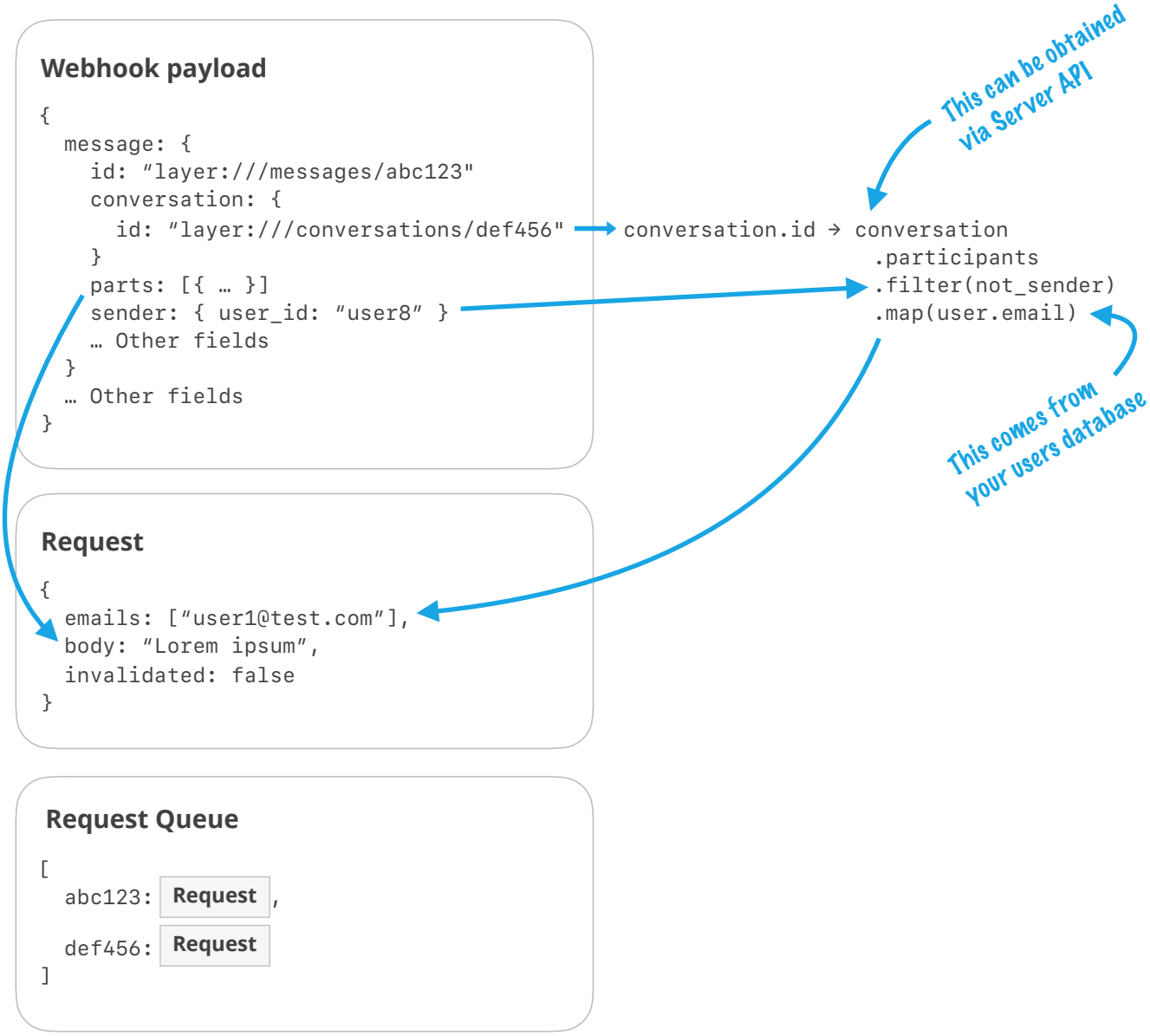
When your application server receives a `message.sent` webhook payload:

5. Acknowledge the request
6. Create a **Request** object (see next section)
7. Enqueue Request into scheduling system for some amount of time in the future. Typically, apps schedule fallback notifications to trigger 10 – 30 minutes after a message is sent. Make sure you have a reference to the enqueued Request, typically via some type of **request ID**. The UUID of the message (included in the webhook payload) would be a good choice for request ID.
8. At some point in the future, the request will be executed — unless it has been invalidated (see below). Executing the request means using the email sender API to trigger an email. This can be done in a separate module in your application code, a separate (micro)service, or a serverless setup.

When your application receives a `message.read` webhook payload:

9. Invalidate the corresponding Request object. Using the message UUID as a request ID makes this simple.

Pseudocode



Requests can be invalidated from the queue:
`queue.invalidate("abc123")`

Resources

- Webhooks library (Node)
- SendGrid integration library (Node)