Jitterbit

Americas Virtual Meetup #13

Interacting with Instant Messaging Applications

March 15, 2023

Agenda

Meetup Agenda

1. Welcome & Introductions
2. Meetup Format / Schedule
3. Outline of Today’s Session
4. Technical Presentation
5. Q&A / Feedback

Schedule

Previous Virtual Meetup available on Success Central:

- Advanced Resources >> Jitterbit Customer Workshops

Today's Agenda:

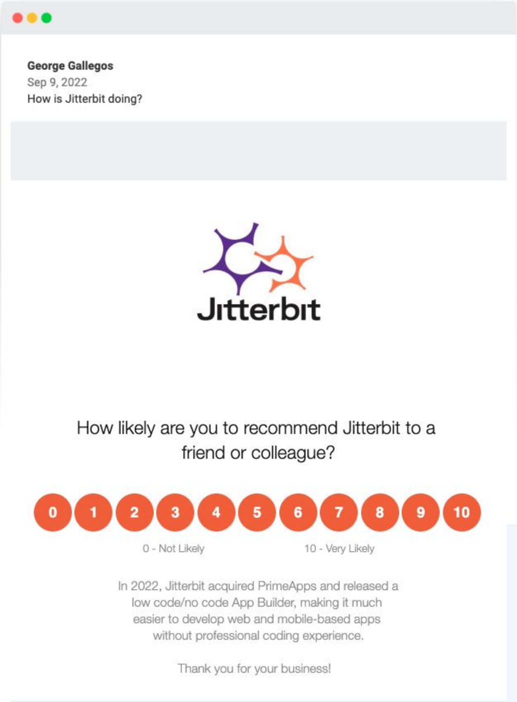
- Interacting with Instant Messaging solution

Event Schedule:

- Next: June 21, 2023
    - **What should we cover? What would be valuable?**
    - **Potential candidates: Message Queuing, LCAP**

Topics/Themes:

- Enablement, Best Practice & Top Tips
- Product Demos
- Cloud Studio & API's
- Real-time Dashboards
- System Monitoring



George Gallegos
Sep 9, 2022
How is Jitterbit doing?

How likely are you to recommend Jitterbit to a friend or colleague?

○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10

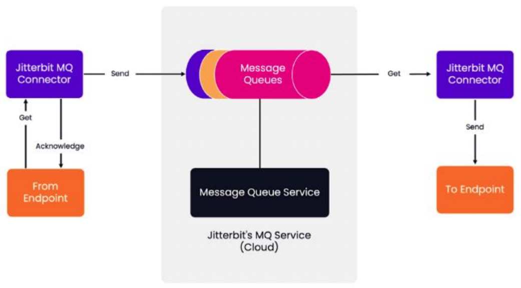
0 - Not Likely   10 - Very Likely

In 2022, Jitterbit acquired PrimeApps and released a low code/no code App Builder, making it much easier to develop web and mobile-based apps without professional coding experience.

Thank you for your business!

Jitterbit MQ

Jitterbit Message Queue (MQ) Service
Beta Program



What is Jitterbit MQ:

Jitterbit’s MQ Service is a cloud-based multi-tenant messaging queue service made available on the Harmony Platform. The service provides customers with the tools to manage their queues to support asynchronous processing, which helps to efficiently manage system workloads and resources.

Who is the Beta Program For:

We’re looking for a small group of Harmony customers and partners to participate in our Beta Program. Beta participants will get the opportunity to fully try the functionality and usability of Jitterbit’s Message Queue (MQ) Service capability, and provide feedback to our Product Management team to inform improvements in the offering.

Who to contact for nomination:

Please reach out to Tomydas Pall (tomydas.pall@jitterbit.com) for nomination or any other questions you may have on the Beta program. CC your CSM.

TODAY'S SPEAKERS

PROFILES

Jaime Delgado

SENIOR CUSTOMER SUCCESS ARCHITECT

Jaime has 15 years of experience in the cloud application and integration technology space helping organizations through digital transformation and automation of processes. Jaime's interests are in advising organizations maximize their use of APIs to drive innovation and value for their customers.

Roberto Espinosa

SENIOR CUSTOMER SUCCESS ARCHITECT

For over two decades Roberto has focused on customer and partner success through strategic IT recommendations and world-class solution delivery. He enjoys getting customers to transformative value.

Mayank Bhasin

SENIOR CUSTOMER SUCCESS ARCHITECT

Mayank is an Enterprise Data Integration Architect by trade who has lead successful technology campaigns including software development and sales, business process improvements, data warehouse modernization on-prem and on-cloud, platform modernization on-prem and on-cloud, data engineering and streaming architectures, data governance, enterprise data cataloging, self-serve data-prep, and data science and advanced analytics; over 17 years of experience and focused mainly on healthcare, retail and utilities industries.

Interacting with Messaging solution

Learnings

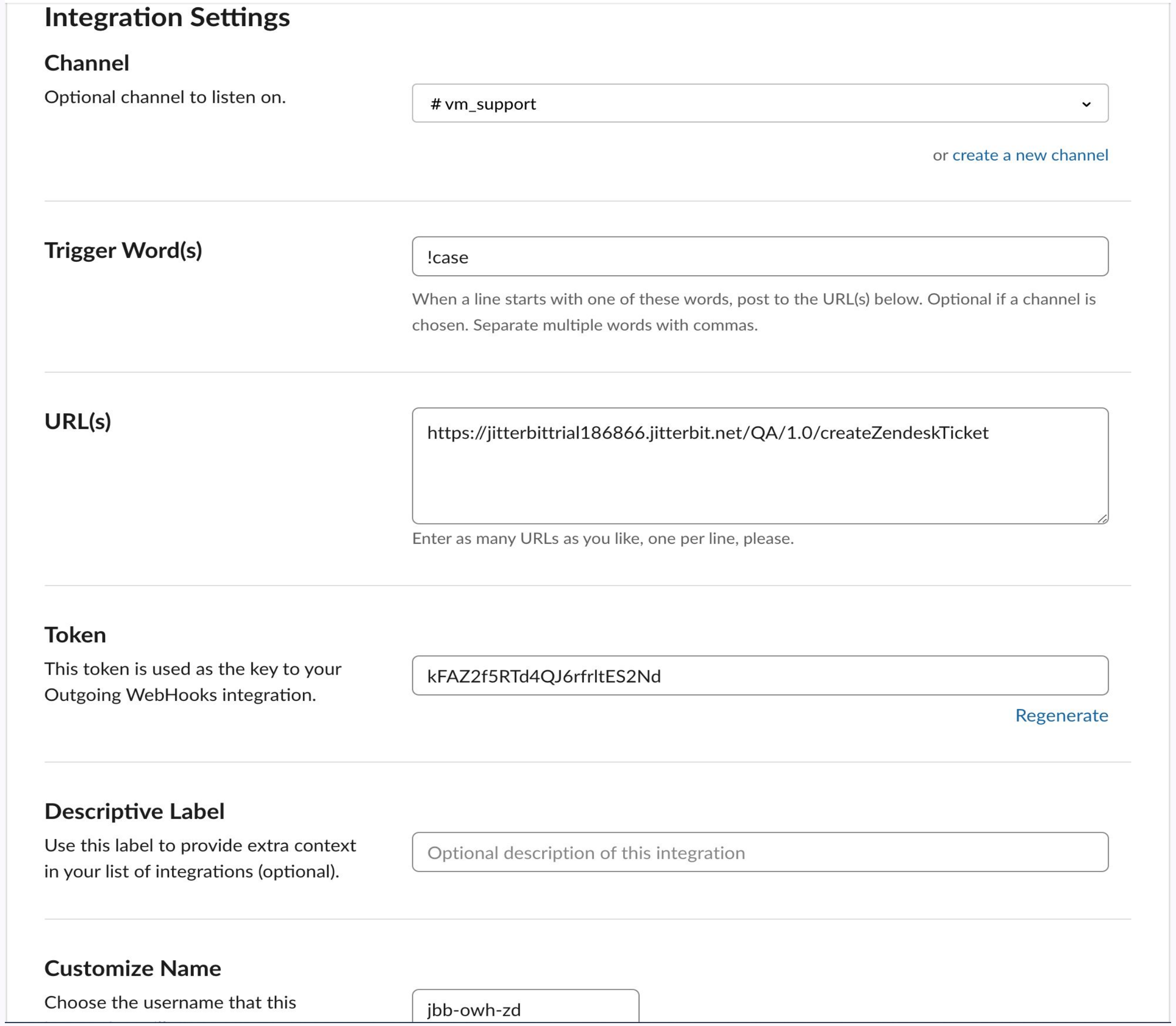
Interacting with an Instant Messaging Application
1. From Slack to Zendesk
2. Application Mashup Salesforce / Zendesk
3. Zendesk event notification to Slack

Use Case 1: Create Zendesk Ticket from Slack

Use Case 1 Demo: Create Zendesk Ticket from Slack

Create Zendesk Ticket from Slack

1. APIM
   a. Create a Jitterbit API URL to receive the message from Slack
2. Slack Channel
   a. Create a slack channel that the user can initiate a request from.
   b. Configure the slack channel
      i. From the Slack App Directory add Outgoing Webhooks* to the channel
      ii. Configure the Outgoing Webhook
         1. Trigger Word that will tell Slack to send a message
         2. URL - Target URL to deliver the message coming from Slack
         3. Optionally customize the name of the integration workflow



Integration Settings

Channel
Optional channel to listen on.
# vm_support ⌄
or create a new channel

Trigger Word(s)
!case
When a line starts with one of these words, post to the URL(s) below. Optional if a channel is chosen. Separate multiple words with commas.

URL(s)
https://jitterbittrial186866.jitterbit.net/QA/1.0/createZendeskTicket
Enter as many URLs as you like, one per line, please.

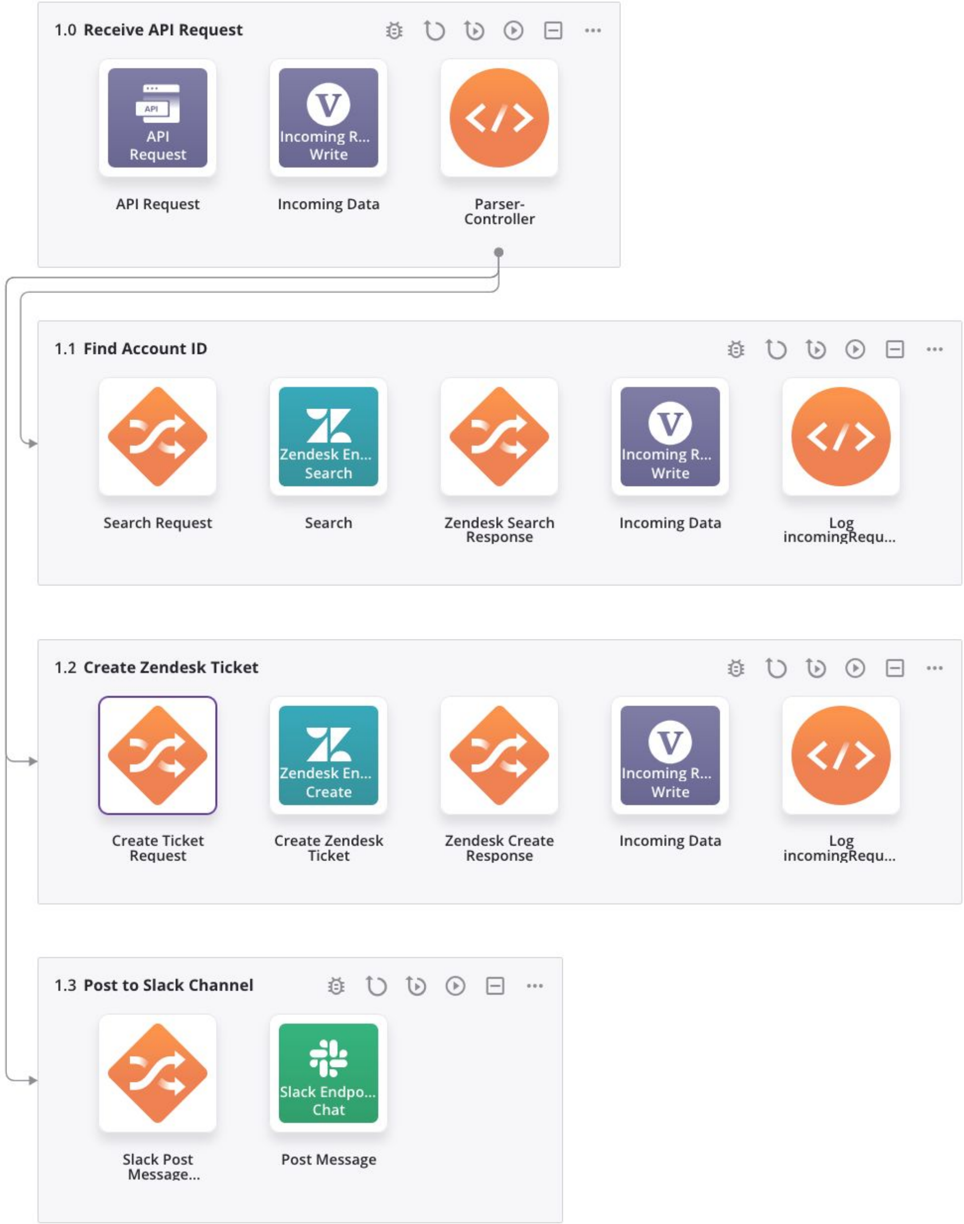
Token
This token is used as the key to your Outgoing WebHooks integration.
kFAZ2f5RTd4QJ6rfrltES2Nd
Regenerate

Descriptive Label
Use this label to provide extra context in your list of integrations (optional).
Optional description of this integration

Customize Name
Choose the username that this
jbb-owh-zd

Create Zendesk Ticket from Slack

3. Configure Cloud Studio Project
   a. Set up an operation to receive the Slack Message
      i. Parses the Slack Message to get the Account/Customer Name and Ticket Description
      ii. Controls the flow of execution for ticket creation
   b. Find Account ID
      i. Searches Zendesk for the Account ID associated with the Account/Customer Name received from Slack. If not found, an error response is returned to Slack
   c. Create Zendesk Ticket
      i. Creates ticket in Zendesk under aforementioned Account ID passing Ticket Description from Slack
   d. Post Ticket ID and URL to Slack



Level of Effort (LOE)

Steps	Activity	Effort
Step a (i)	Research Zendesk Documentation	45 minutes
Step (b)	Build Custom API using Jitterbit APIM	1 minute
Step (c)	Create and Configure Slack Channel	5 minutes
Step (d)	Build Jitterbit Cloud Studio Workflow	3 Hours
Step (e)	Testing	30 minutes
	TOTAL	**4.35 hours**

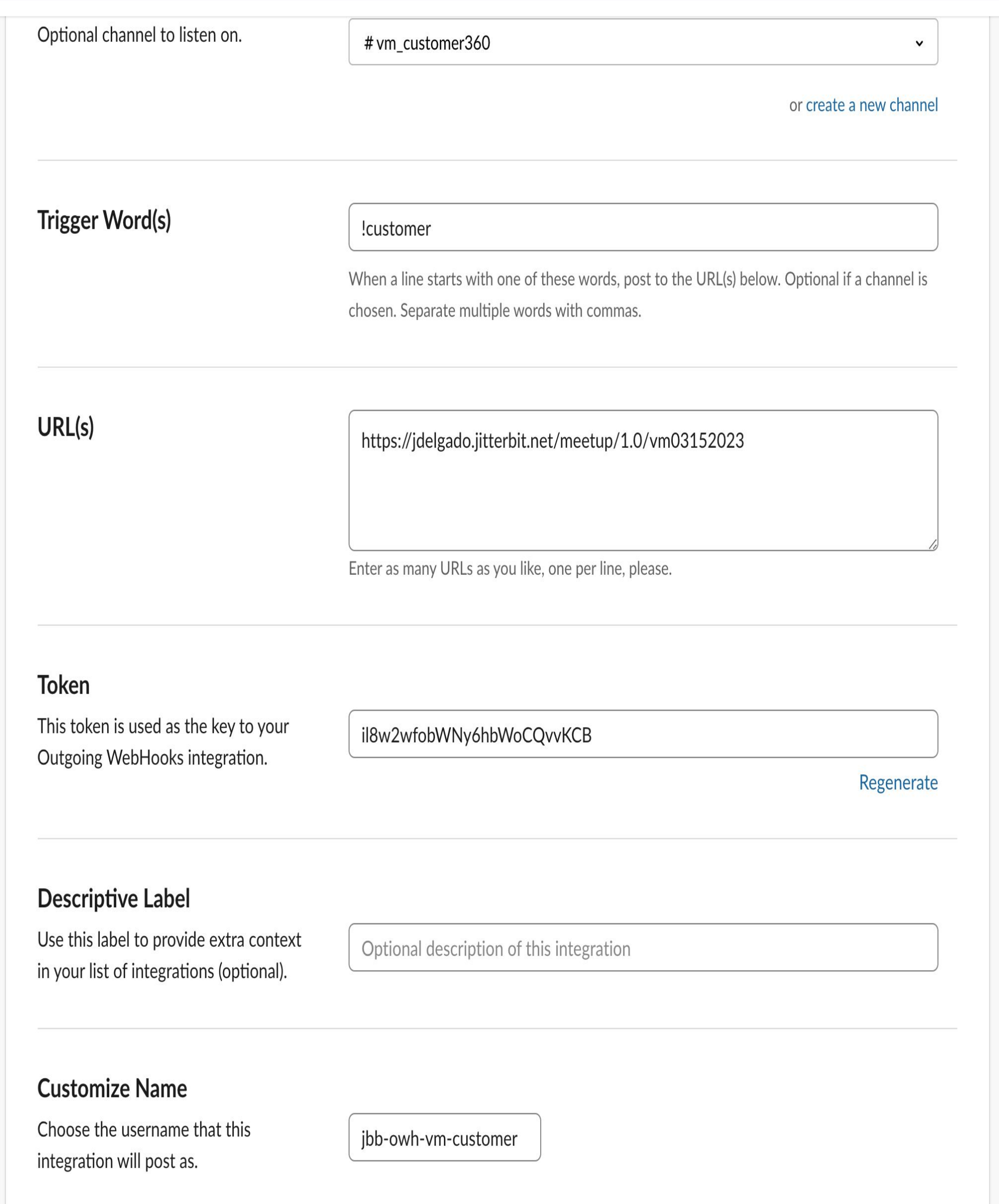
PS: The LOE for a second use case with similar approach as above would be lower as compared to above as a lot of the time in the first use case is consumed while reading through the endpoint documentation.

Use Case 2 Demo: Slack to Mashup (Salesforce, Zendesk)

Slack to Mashup (Salesforce, Zendesk)

Application Mashup Salesforce / Zendesk

1. APIM
   a. Create a Jitterbit API URL to receive the message from Slack
2. Slack Channel
   a. Create a slack channel that the user can initiate a request from.
   b. Configure the slack channel
      i. From the Slack App Directory add Outgoing Webhooks to the channel
      ii. Configure the Outgoing Webhook
         1. Trigger Word that will tell Slack to send a message
         2. URL - Target URL to deliver the message coming from Slack
         3. Optionally customize the name of the integration workflow



Optional channel to listen on. # vm_customer360

or create a new channel

Trigger Word(s)

!customer

When a line starts with one of these words, post to the URL(s) below. Optional if a channel is chosen. Separate multiple words with commas.

URL(s)

https://jdelgado.jitterbit.net/meetup/1.0/vm03152023

Enter as many URLs as you like, one per line, please.

Token

This token is used as the key to your Outgoing WebHooks integration.

il8w2wfobWNy6hbWoCQvvKCB

Regenerate

Descriptive Label

Use this label to provide extra context in your list of integrations (optional).

Optional description of this integration

Customize Name

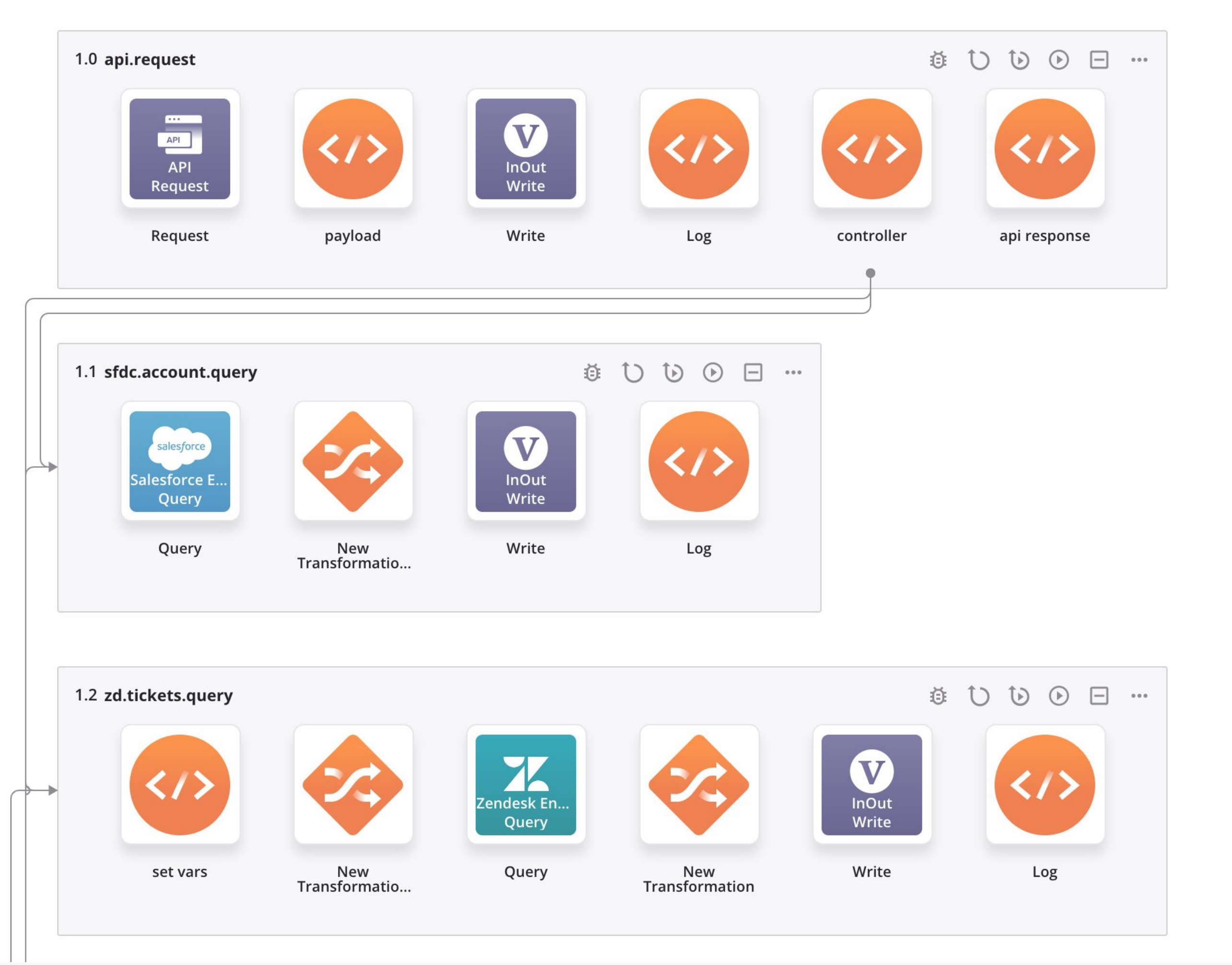
Choose the username that this integration will post as.

jbb-owh-vm-customer

Slack to Mashup (Salesforce, Zendesk)

Application Mashup Salesforce / Zendesk

3. Configure Cloud Studio Project
   a. Configure an operation to receive the Slack Message
      i. This operation reads the Slack Message to get the Account/Customer Name
   b. Salesforce Account Query
      i. This operation retrieves the account details for the Account/Customer Name passed from the Slack Channel
   c. Zendesk Ticket Query
      i. This operation retrieves the number of unresolved tickets for this Account/Customer



Level of Effort (LOE)

Steps	Activity	Effort
Step a (i)	Research Zendesk Documentation	45 minutes
Step (b)	Build Custom API using Jitterbit APIM	1 minutes
Step (c)	Create and Configure Slack Channel	5 minutes
Step (d)	Build Jitterbit Cloud Studio Workflow	3 Hours
Step (e)	Testing	30 minutes
	TOTAL	**4 hours**

PS: The LOE for a second use case with similar approach as above would be lower as compared to above as a lot of the time in the first use case is consumed while reading through the endpoint documentation.

Use Case 3: Zendesk ticket notification into a Slack channel

Use Case 3: Zendesk ticket notification into a Slack channel

1. Use case demonstration
   a. Change the status of a ticket in Zendesk and submit
      i. Open
      ii. Pending
      iii. Solved
      iv. Create New
   b. Get a new Zendesk notification in the Slack channel with the following ticket details
      i. Account Name
      ii. Ticket ID
      iii. Ticket Priority
      iv. Ticket Assignee Name
      v. Organization Name
      vi. Ticket Requester Name
      vii. Ticket Description
      viii. Ticket Due Date
      ix. Ticket Status
      x. Ticket Title
      xi. New Comment
      xii. Ticket URL

Zendesk ticket notification into a Slack channel…. *contd.*

1. Use case approach

   a. Zendesk

      i. Webhook

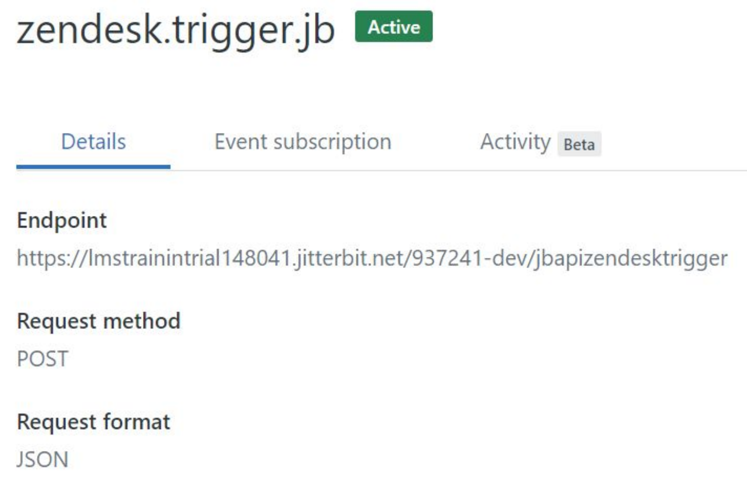
         1. Build a Webhook in Zendesk that will communicate with Jitterbit custom API
         2. Provide URL from Jitterbit APIM into the Webhook (this step will be performed after step (d))

      ii. Trigger

         1. Build a trigger in Zendesk
         2. Provide a condition / business rule in the trigger based on the use case requirement
            a. Run the trigger when the status of the ticket changes
         3. Configure action to be taken whenever the above condition or rule is true
            a. Notify webhook (built above) to communicate with Jitterbit custom API (see steps (c, d))
         4. Provide JSON body that needs to be sent to the Jitterbit API

   b. Slack

      i. Create a new channel to receive notifications from Zendesk
      ii. Capture channel ID to be used in the API

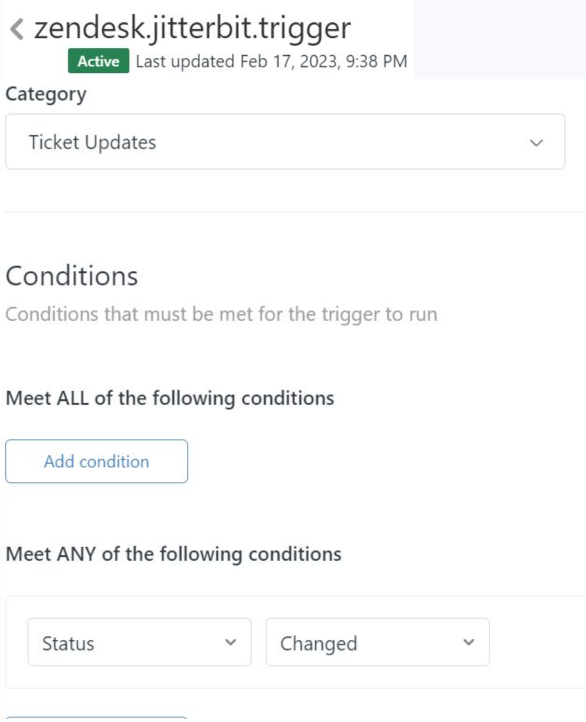


zendesk.trigger.jb Active

Endpoint
https://lmstrainintrial148041.jitterbit.net/937241-dev/jbapizendesktrigger

Request method
POST

Request format
JSON



zendesk.jitterbit.trigger
Active Last updated Feb 17, 2023, 9:38 PM

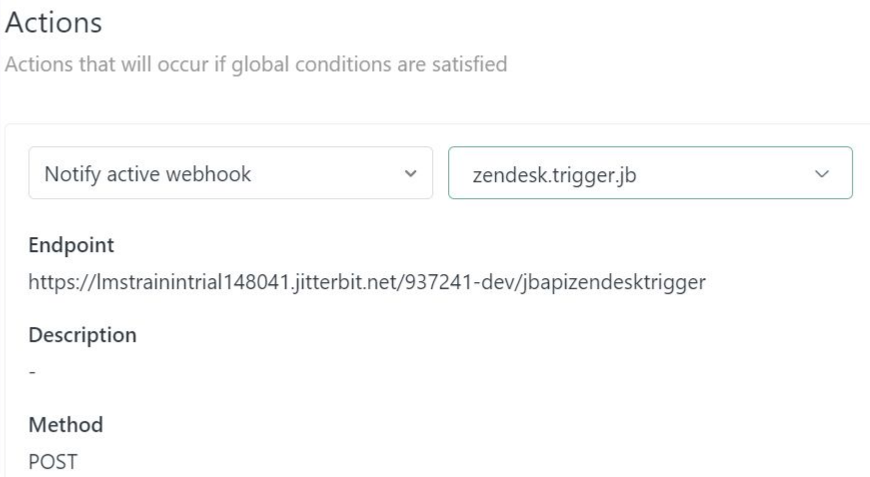
Category
Ticket Updates

Conditions
Conditions that must be met for the trigger to run

Meet ALL of the following conditions

Meet ANY of the following conditions

Status Changed



Actions
Actions that will occur if global conditions are satisfied

Notify active webhook zendesk.trigger.jb

Endpoint
https://lmstrainintrial148041.jitterbit.net/937241-dev/jbapizendesktrigger

Description
-

Method
POST

Zendesk ticket notification into a Slack channel…. *contd.*

1. Use case approach…. *contd.*

c. Jitterbit Cloud Studio

  i. Workflow

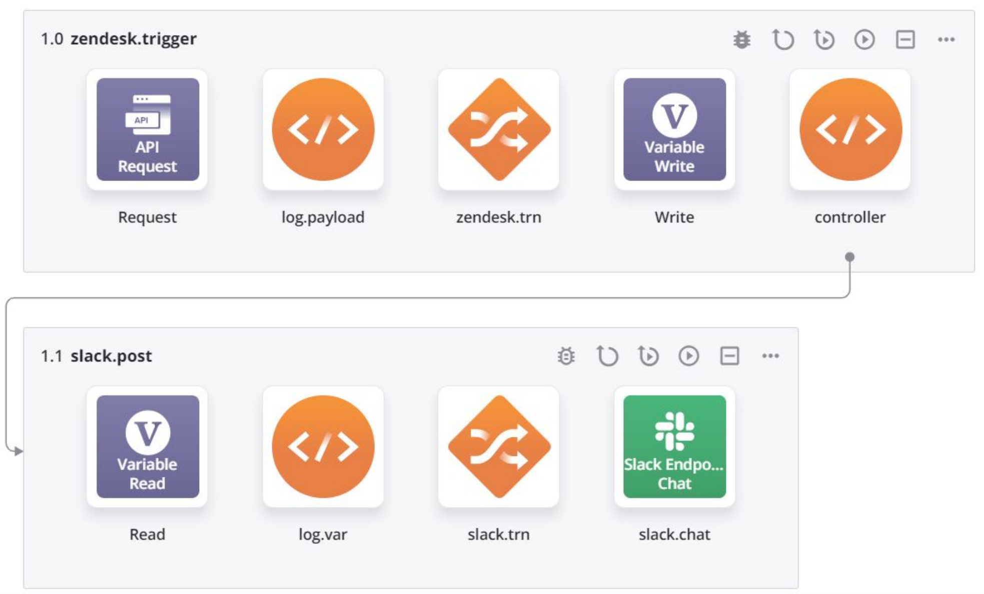
    1. Build a Workflow to receive API request from the Webhook built in Zendesk
    2. Provide Slack channel ID and perform transformation (if needed) and write the information into the Slack channel using out-of-the-box Jitterbit Slack connector

      a. Need Slack Bot user access token to establish connection with the Slack Workspace

d. Jitterbit APIM

  i. Use Jitterbit APIM to expose the Workflow built in Cloud Studio above as a custom API (Method: POST) to communicate with Zendesk Webhook

  ii. Capture the API URL and provide it in the Zendesk Webhook built in step a (i)



Zendesk ticket notification into a Slack channel*…. contd.*

1. Level of Effort (LOE)

Steps	Activity	Effort
Step a (i)	Build Zendesk Webhook	30 minutes
Step a (ii)	Build Zendesk Trigger	1 hour
Step (b)	Configure Slack Channel	15 minutes
Step (c)	Build Jitterbit Cloud Studio Workflow	2 hours
Step (d)	Build Custom API using Jitterbit APIM	15 minutes
	TOTAL	**4 hours**

PS: The LOE for a second use case with similar approach as above would be lower as compared to above as a lot of the time in the first use case is consumed while reading through the endpoint documentation.

Other Ways to Get Help from Jitterbit

Interested in any of the following ⇒ Please reach out to your CSM:

- Learn more about our Solutions (Integration, Automation, App Building, & EDI)
- Best Practices session with Customer Success Architect (CSA)
- Health Check review with CSA
- Architecture assessment with CSA
- Working session with CSA for Jitterbit examples and/or development activities
- CSA office hours every Wednesday @ 10am Pacific Time
- Customer workshops or customized training with CSA or TAM (customized to your use cases)

- **Contact:**
  - https://success.jitterbit.com/
  - Unsure who your CSM is ⇒ email Success@jitterbit.com or phil@jitterbit.com

Jitterbit

QUESTIONS & FEEDBACK

Rock your workflow!