

Trayio Implementation and Sales Engineer Candidate Test

High Level Requirements: The integration should take in a JSON webhook as input (from Salesforce) and propagate that information through the pipeline designed. The Slack API is given the “country” entry from the webhook to determine the appropriate slack channel to post the message to.

Overview: See datapath/flowchart. I have designed two datapaths (one with Amazon SQS for bonus assignment)

API Methods:

- **chat.postMessage:** Sends a message to a channel
- **chat.update:** In case the client wants to change an existing message, they can make use of the chat.update method rather than sending a brand new message (if we’re dealing with a message limit this is important to consider)
- **api.test:** Checks API calling code. This is to avoid any errors.
- **conversations.archive:** If there are messages that are sent to the slack channel that are some form of *anomaly* or significance, then the client should be able to save the conversation.
- **chat.delete:** Deletes a message. Useful if there is a message that you want the entire channel to disregard.
- **reactions.add:** Adds a reaction to an item. Adds a personal touch to a message to show that you have seen the message. Also allows users to display if they have positive or negative feelings towards the message.
- **reactions.list:** Lists all the reactions by a user. This is important because the client may want to know who acknowledged the message and who did not.

Other requirements: 1) We need a database to store the messages for several reasons. It is a form of caching in case we need to go back into the previously sent messages. Doing so gives us confidence in our internal security, in case a message contained confidential information we will know what kind of information was messaged to the slack channel. Lastly, these webhooks may be important in the future for a variety of use cases, e.g managing efficiency, etc. 2) An additional API is required to dispatch the required information to Slack API from Salesforce and to store the webhooks in the database. 3) According to Slack, Slack Channels need to enable webhooks in order to receive them.

Follow Up Questions:

1. Does this spec of the slack integration meet your expectations?
2. Are there any API limitations we need to be made aware of?
3. Are the API methods sufficient for your company’s purposes or do you need more?
4. We would like to schedule a tentative date for a User Acceptance Test (UAT) so we can show you full functionality of the integration upon completion. When would you like that to happen?

Bonus Assignment Detail:

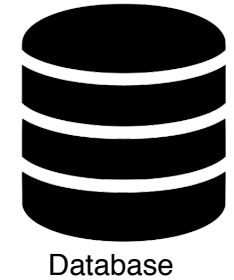
I designed a separate datapath that includes Amazon SQS. This allows for the Slack limitation of 50 messages per hour to be met. The Salesforce webhook first gets stored in the SQS queue and then there is a signal that determines if the Slack API is ready to receive messages (50 message restriction)—if the Slack API is ready, the message will be POSTed to the additional API I included. From there, the rest of the datapath is the same as the original. The signal

mentioned before is a response outputted by the Slack API when it has received a max number of messages (HTTP 429 “Too Many Requests” error). I placed a conditional boolean (see red box) that takes this signal as input and then outputs 0 or 1 accordingly.

INPUT

```
{
  "id": "006580000BwzJt",
  "name": "Sample02",
  "status": "Qualification",
  "account": "Andrew Test",
  "close_date": "2017-03-16",
  "type": "New Business",
  "country": "UK", // can also be US or EU
  "owner_name": "Andrew",
  "owner_email": "andrew@tray.io",
  "amount": 4500
}
```

Extra API that propagates two sets of important information to Slack API and to DB. The box on the right is content that will appear in the slack channel

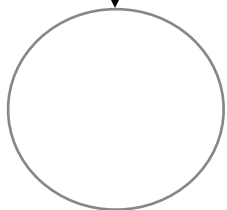


Database

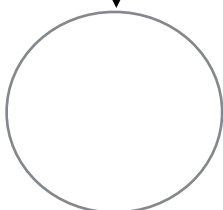
```
{
  "country": "UK"
}
```

```
{
  "name": "Sample02",
  "account": "Andrew Test",
  "close_date": "2017-03-16",
  "type": "New Business",
  "owner_name": "Andrew",
  "amount": 4500
}
```

SLACK API uses "country" to choose which #channel to notify. In this case, #sales-UK is selected.



#sales-US



#sales-EU



#sales-UK



Salesforce APP: 2:47 PM

Opportunity created with stage "Qualification"

Opportunity Name	Account Name
Sample02	ANDREW TEST
Close Date	Amount
2017-03-16	\$4500
Type	Owner
New Business	Andrew

Sent via tray.io

OUTPUT

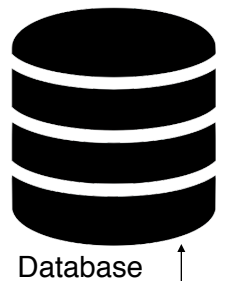
INPUT

```
{
  "id": "006580000BwzJt",
  "name": "Sample02",
  "status": "Qualification",
  "account": "Andrew Test",
  "close_date": "2017-03-16",
  "type": "New Business",
  "country": "UK", // can also be US or EU
  "owner_name": "Andrew",
  "owner_email": "andrew@tray.io",
  "amount": 4500
}
```

Don't delete top message from queue until
Slack API is ready to receive



Is Slack API
ready to
receive?



Database

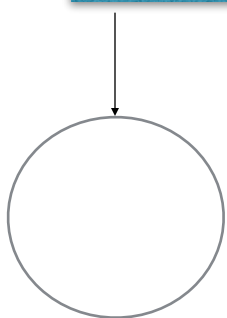
Extra API that propagates two sets of
important information to Slack API
and to DB. The box on the right is
content that will appear in the slack
channel

If HTTP 429 Too Many Req == True:
Output 0 to halt propagation
If HTTP 429 Too Many Req == False:
Output 1 to resume propagation

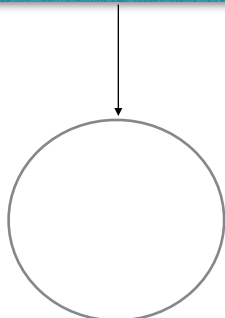
SLACK API uses "country" to choose which
#channel to notify. In this case, #sales-UK
is selected.

```
{
  "country": "UK"
}
```

```
{
  "name": "Sample02",
  "account": "Andrew Test",
  "close_date": "2017-03-16",
  "type": "New Business",
  "owner_name": "Andrew",
  "amount": 4500
}
```



#sales-US



#sales-EU



#sales-UK



Salesforce APP: 2:47 PM

Opportunity created with stage "Qualification"

Opportunity Name
Sample02
Close Date
2017-03-16
Type
New Business
Sent via tray.io

Account Name
ANDREW TEST
Amount
\$4500
Owner
Andrew

OUTPUT