

Events

QBench Events Messaging v1.0 Documentation

Overview

QBench provides API users the ability to take action based on an event that occurs in the application. Third party applications can listen for events such as a sample being updated to programmatically fetch or update data. There are endless possibilities for how this functionality can be used, below are only a few basic examples:

- Fetch data about an order to prepare and invoice for downstream accounting systems.
- Listen for a test being completed to fetch result data to perform analysis
- Listen for tests being completed to render a complex report using third party software and reattach the report
- Aggregate data for report or emailing business metrics
- Many, many, more

This guide describes the underlying architecture, how to register a listener, the notification structure, and the event types.

Built on Amazon's Simple Notification Service (SNS)

The QBench notifications API uses AWS Simple Notification Service (SNS) to send notifications to listeners. SNS already offers various options to bind listeners of which QBench supports two: HTTP(S) endpoint and AWS SQS.

Registering a Listener

There are two types of listeners that QBench developers can utilize:

1. HTTP(S) endpoint
This option allows you to specify an HTTP URL that can receive a POST request when the notification is fired. Contact our development team at support@qbench.net with your request and we will assist you in registering your endpoint. AWS provides detailed documentation on how to receive messages and confirm subscriptions here: <http://docs.aws.amazon.com/sns/latest/dg/SendMessageToHttp.html>
2. AWS Simple Queue Service
With this option you are required to have your own AWS account. Contact our development team at support@qbench.net with your request and we will assist you in subscribing to the SNS topic. Details on subscribing your topic can be found here: <http://docs.aws.amazon.com/sns/latest/dg/SendMessageToSQS.cross.account.html>

Subscribing to your SNS Topic

These are the instructions on how to subscribe to your QBench instance's SNS Topic:

1. Create an AWS Account (preferably business account)
 - There is a free tier option for the account
2. On the Dashboard, you'll want to type in "Lambda" to the Search Services box
3. Click the "Create function" button at the top right
4. Fill out the information and click "Create function" at the bottom right
5. At the top right you'll notice an ARN that you can copy
6. Send that ARN to support@qbench.net and we can grant access to your account so that you can subscribe the Lambda function to the SNS topic
7. Once we've set up your permissions, click the "+ Add trigger" button
8. Select SNS as the Trigger
9. In the SNS topic box, you'll be able to see:
Arn:aws:sns:us-east-1:689663267667:qbench-events-prod-{your instance}
 and click Add

Notification Structure

The notification will be in JSON format with the following fields:

Notification Fields

Name	Description
qbn_type	This is the type of notification (i.e. entity_updated).
qbn_entity	This will specify which data type the notification is about.
qbn_id	This will be the ID of the entity in QBench the notification is about.
qbn_addl	This field may contain additional information for special notifications.

Event Types

QBench sends notifications on data updates and status changes. The table below describes the type of notifications.

Notification Types

Name	Description	Entity Type
entity-created	When a data object has been created in	order, sample, test,

	QBench.	customer, contact, assay, source, panel, qcdata
entity-updated	When a data object has been updated in QBench.	order, sample, test, customer, contact, assay, source, panel, qcdata
entity-deleted	When a data object has been deleted in QBench.	order, sample, test, customer, contact, assay, source, panel, qcdata
entity-removed NOT IMPLEMENTED	When an entity has been disassociated from another entity. Will include additional fields: "removed_type" and "removed_id". Note: this is not the same as deleted.	order, sample, test, customer, contact, assay, source, panel, qcdata
status-changed	When the QBench status of an entity changes. Includes additional field "new_status".	order, test
all-tests-completed NOT IMPLEMENTED	When all tests on a sample are completed	sample
assay-condition-met NOT IMPLEMENTED	When conditions for an assay are met. Will include additional field "assay-conditions".	test
attachment-uploaded	When any attachment is uploaded in QBench	order, sample, test, source, customer, contact, assay, panel, batch, document, log entry, project, issue, supplier, equipment, record, location, invoice, portal documentation, help center, inventory item, reports, print docs, generic attachments
attachment-deleted	When any attachment is deleted in QBench	order, sample, test, source, customer, contact, assay, panel, batch, document, log entry, project, issue, supplier, equipment, record, location, invoice, portal documentation, help center, inventory item, reports, print docs, generic attachments
custom	A custom notification that can be manually sent from within QBench.	test, sample

	Includes additional field "message" that is the name of the custom notification.	
--	--	--

Example messages

```
// basic entity update
{
  "qbn_type": "entity-updated",
  "qbn_entity": "order",
  "qbn_id": "1235584"
}

// status update
{
  "qbn_type": "status-changed",
  "qbn_entity": "test",
  "qbn_id": "55448",
  "qbn_addl": {
    "new_status": "COMPLETED",
  }
}

// attachment deleted
{
  'qbn_type': 'attachment-deleted',
  'qbn_data_type': 'order',
  'attachment_id': 1,
  'asset_id': 2
}

// attachment uploaded
{
  "qbn_type": "attachment-uploaded",
  "qbn_data_type": "sample",
  "attachment_id": 1,
  "asset_id": 2,
  "key": "QBench/1/2.png",
  "params": {
    "object_id": 200,
    "data_type": "sample",
    "upload_type": "Sample"
  }
}
```

Revision History

Date	By	Changes
2017-09-02	Joel Clark	Initial
2019-06-14	Kyle Phan	Added information related to attachments