Act in Time

InfluxDB JS Client

Javascript client to InfluxDB v2 APIs

Direct write into InfluxDB

$ curl -i -X POST 'http://localhost:8086/api/v2/write?org=my-org&bucket=my-bucket' \

    -H 'Authorization: Token my-token' \

    -d '**cpu_load_short,host=server01,region=us-west value=0.64 1434055562000000000**'

Format: **line protocol**

measurement	tag_set	field_set	timestamp
cpu_load_short	,host=server01,region=us-west	value=0.64	1434055562000000000

InfluxDB Client Libraries

Access InfluxDB API from various programming languages

- **Arduino** – https://github.com/tobiasschuerg/InfluxDB-Client-for-Arduino
- **C#** – https://github.com/influxdata/influxdb-client-csharp
- **Dart** – https://github.com/influxdata/influxdb-client-dart
- **Go** – https://github.com/influxdata/influxdb-client-go
- **Java** – https://github.com/influxdata/influxdb-client-java
- **JavaScript** – https://github.com/influxdata/influxdb-client-js
- **PHP** – https://github.com/influxdata/influxdb-client-php
- **Python** – https://github.com/influxdata/influxdb-client-python
- **Ruby** – https://github.com/influxdata/influxdb-client-ruby
- **Swift** – https://github.com/influxdata/influxdb-client-swift

Docs – https://docs.influxdata.com/influxdb/cloud/tools/client-libraries

JavaScript client library

Two main packages (for node.js):

- @influxdata/**influxdb-client**
  - **Querying** data using the Flux language
  - **Writing** data (batched, automatic retries)
  - Web Browser, Deno: @influxdata/**influxdb-client-browser**
- @influxdata/**influxdb-client-apis**
  - **Manage** sources, buckets, authorizations, tasks, …
  - **Health** check
- Examples: https://github.com/influxdata/influxdb-client-js/tree/master/examples
- API documentation: https://influxdata.github.io/influxdb-client-js

Write data

Smart write logic implemented

- Non-blocking
- Buffering
- Retry

```javascript
const {InfluxDB, Point, HttpError} = require('@influxdata/influxdb-client')
const {url, token, org, bucket} = require('./env')
const {hostname} = require('os')

// create a write API
const writeAPI = new InfluxDB({url, token}).getWriteApi(org, bucket, 'ns')
writeApi.useDefaultTags({host: hostname()})

// write point with the current timestamp
const point1 = new Point('temperature')
  .tag('example', 'write.ts')
  .floatField('value', 20 + Math.round(100 * Math.random()) / 10)
writeAPI.writePoint(point1)

// flush data, close API
writeAPI.close()
```

Query data

```
import {InfluxDB, FluxTableMetaData} from '@influxdata/influxdb-client'
import {url, token, org} from './env'

const queryApi = new InfluxDB({url, token}).getQueryApi(org)
const fluxQuery = 'from(bucket:"my-bucket") |> range(start: 0) |> filter(fn: (r) => r._measurement == "temperature")'

// Execute query and receive table metadata and rows
queryApi.queryRows(fluxQuery, {
  next(row: string[], tableMeta: FluxTableMetaData) {
    const o = tableMeta.toObject(row)
    console.log( `${o._time} ${o._measurement} in '${o.location}' (${o.example}): ${o._field}=${o._value}` )
  },
  error(error: Error) {
    console.error(error)
    console.log('\nFinished ERROR')
  },
  complete() {
    console.log('\nFinished SUCCESS')
  },
})
```

Create Bucket

All management APIs are documented in
https://influxdata.github.io/influxdb-client-js/influxdb-client-apis.html

```javascript
const {InfluxDB, HttpError} = require('@influxdata/influxdb-client')
const {OrgsAPI, BucketsAPI} = require('@influxdata/influxdb-client-apis)
const {url, org, token} = require('./env')
const influxDB = new InfluxDB({url, token})

const name = 'example-bucket'
const orgsAPI = new OrgsAPI(influxDB)
//Get organizations
const organizations = await orgsAPI.getOrgs({org})
const orgID = organizations.orgs[0].id

// get bucket - if exists, delete it
const bucketsAPI = new BucketsAPI(influxDB)
const buckets = await bucketsAPI.getBuckets({orgID, name})
if (buckets && buckets.buckets && buckets.buckets.length) {
  const bucketID = buckets.buckets[0].id
  await bucketsAPI.deleteBucketsID({bucketID})
}
// creates a bucket, entity properties are specified in the "body" property
const bucket = await bucketsAPI.postBuckets({body: {orgID, name}})
```

Documentation

More information:

https://influxdata.github.io/influxdb-client-js/