**Real Time Data Streaming and Decision Making**

Opex Analytics

Table of Contents

[**Introduction** 3](#_Toc488938459)

[**Data** 4](#_Toc488938460)

[Data Sources 4](#_Toc488938461)

[Files Shared by Vendors 5](#_Toc488938462)

[gbfs.json 5](#_Toc488938463)

[system\_information.json 7](#_Toc488938464)

[station\_information.json 9](#_Toc488938465)

[station\_status.json 11](#_Toc488938466)

[**Technologies** 13](#_Toc488938467)

[Data Ingestion 13](#_Toc488938468)

[Data Processing 13](#_Toc488938469)

[**References** 14](#_Toc488938470)

**Introduction**

The aim of this project is to develop and prototype a real time data streaming and decision making solution. As a working example, we will use the bike sharing data. The first step towards achieving this is to develop and prototype a solution that reads real-time streaming data, stores it and presents it in a format ready for consumption by analytical processes. This is then followed by using the data streamed to build analytical solutions to make real time decisions. For this working example, the questions answered would be:

* Where are there are bike stock outs?
* Where will there be stockouts in the next x minutes.
* What is the probability there will be a bike available for me at station x in y minutes?
* Where should I send my trucks with extra bikes to restock?

**Data**

# Data Sources

The following is a table comparing the various vendors:

|  |  |  |  |
| --- | --- | --- | --- |
| **Vendor** | **Locations** | **Refresh Interval(s)** | **Historic Data**  **(since)** |
| Bike Town | Portland, OR | 60 | N.A. |
| Capital Bike Share | Washington, DC | 10 | 2010 Q4 |
| Citi Bike | New York, NY | 10 | 2013 Q3 |
| CoGo Bikes | Columbus, OH | 30 | N.A. |
| Divy Bikes | Chicago, IL | 30 | 2013 Q3 |
| Nice Ride | Minnesota, MN | 10 | 2010 June |
| The Hubway | Boston, MA | 10 | 2011-2013 |

All the above mentioned vendors publish data in General Bikeshare Feed Specification (GBFS) format. All files shared in the GBFS format are JSON files with utf-8 encoding. Every JSON file presented in this specification contains the same common header information at the top level of the JSON response object:

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Required** | **Defines** |
| last\_updated | Yes | Integer POSIX timestamp indicating the last time the data in this feed was updated. |
| ttl | Yes | Integer representing the number of seconds before the data is refreshed (0 if the data should always be refreshed). |
| data | Yes | JSON hash containing the data fields for this response. |

# Files Shared by Vendors

This specification defines the following files along with their associated content:

|  |  |  |
| --- | --- | --- |
| **File Name** | **Required** | **Defines** |
| gbfs.json | Optional | Auto-discovery file that links to all of the other files published by the system. This file is optional, but highly recommended. |
| system\_information.json | Yes | Describes the system including System operator, System location, year implemented, URLs, contact info, time zone |
| station\_information.json | Yes | Mostly static list of all stations, their capacities and locations |
| station\_status.json | Yes | Number of available bikes and docks at each station and station availability |
| free\_bike\_status.json | Optional | Describes bikes that are available in non station-based systems |
| system\_hours.json | Optional | Describes the hours of operation for the system |
| system\_calendar.json | Optional | Describes the days of operation for the system |
| system\_regions.json | Optional | Describes the regions the system is broken up into |
| system\_pricing\_plans.json | Optional | Describes the system pricing |
| system\_alerts.json | Optional | Describes current system alerts |

## gbfs.json

The following fields are attributes in the gbfs.json file:

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Required** | **Defines** |
| language | Yes | The language that all of the contained files will be published in. This language must match the value in the system\_information file. |
| -feeds | Yes | An array of all the feeds that are published by this auto-discovery file. |
| -name | Yes | Key identifying the type of feed this is (e.g. "system\_information", "station\_information"). |
| -url | Yes | Full URL for the feed. |

Example:



## system\_information.json

The following fields are all attributes within the main "data" object for this feed.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Required** | **Defines** |
| system\_id | Yes | ID field - identifier for this bike share system. This should be globally unique. In addition, this value is intended to remain the same over the life of the system |
| language | Yes | An IETF language tag indicating the language that will be used throughout the rest of the files. This is a string that defines a single language tag only. |
| name | Yes | Full name of the system to be displayed to customers |
| short\_name | Optional | Optional abbreviation for a system |
| operator | Optional | Name of the operator of the system |
| url | Optional | The URL of the bike share system. The value is a fully qualified URL that includes http:// or https://, and any special characters in the URL must be correctly escaped. |
| purchase\_url | Optional | A fully qualified URL where a customer can purchase a membership or learn more about purchasing memberships |
| start\_date | Optional | String in the form YYYY-MM-DD representing the date that the system began operations |
| phone\_number | Optional | A single voice telephone number for the specified system. This field is a string value that presents the telephone number as typical for the system's service area. |
| email | Optional | A single contact email address for customers to address questions about the system |
| timezone | Yes | The time zone where the system is located. |
| license\_url | Optional | A fully qualified URL of a page that defines the license terms for the GBFS data for this system, as well as any other license terms the system would like to define (including the use of corporate trademarks, etc) |

Example:



## station\_information.json

All stations contained in this list are considered public (ie, can be shown on a map for public use). If there are private stations (such as Capital Bikeshare’s White House station) these should not be exposed here and their status should not be included in station\_status.json.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Required** | **Defines** |
| stations | Yes | Array that contains one object per station in the system as defined below |
| - station\_id | Yes | Unique identifier of a station. |
| - name | Yes | Public name of the station |
| - short\_name | No | Short name or other type of identifier, as used by the data publisher |
| - lat | Yes | The latitude of station. |
| - lon | Yes | The longitude of station. |
| - address | Optional | Valid street number and name where station is located. |
| - cross\_street | Optional | Cross street of where the station is located. |
| - region\_id | Optional | ID of the region where station is located. |
| - post\_code | Optional | Postal code where station is located |
| - rental\_methods | Optional | Array of enumerables containing the payment methods accepted at this station. |
| - capacity | Optional | Number of total docking points installed at this station, both available and unavailable |

Example:



## station\_status.json

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Required** | **Defines** |
| stations | Yes | Array that contains one object per station in the system as defined below. |
| - station\_id | Yes | Unique identifier of a station. |
| - num\_bikes\_available | Yes | Number of bikes available for rental. |
| - num\_bikes\_disabled | Optional | Number of disabled bikes at the station. |
| - num\_docks\_available | Yes | Number of docks accepting bike returns. |
| - num\_docks\_disabled | Optional | Number of empty but disabled dock points at the station. This value remains as part of the spec as it is possibly useful during development |
| - is\_installed | Yes | 1/0 boolean - is the station currently on the street. |
| - is\_renting | Yes | 1/0 boolean - is the station currently renting bikes. |
| - is\_returning | Yes | 1/0 boolean - is the station accepting bike. |
| - last\_reported | Yes | Integer POSIX timestamp indicating the last time this station reported its status to the backend |

Example:



**Technologies**

# Data Ingestion

|  |  |  |
| --- | --- | --- |
| **Technology** | **Pros** | **Cons** |
| Kafka | Highly available, redundant, scalable, one-to-many messaging. | Relatively new, lack of commercial support, no built in connectors to Hadoop products. |
| Flume | Stable, well-established, natively supported by Hadoop. | One-to-one messaging, not redundant. |

# Data Processing

|  |  |  |
| --- | --- | --- |
| **Technology** | **Pros** | **Cons** |
| Storm | Very low latency, well-established. | No guarantee that data will be processed only once. |
| Spark-streaming | Data is processed reliably, interfaces seamlessly with Spark. | Micro-batching instead of true streams, moderate latency. |
| Flink | Very low latency, data is processed reliably. | No commercial support. |

# Data Storage

|  |  |  |
| --- | --- | --- |
| **Technology** | **Pros** | **Cons** |
| Cassandra | Availability, easy maintenance, good with large scale. | No multiple secondary indexes, no dynamic querying on columns, slow on-the fly aggregations. |
| MongoDB | Flexible, highly available, good for real time systems. | No foreign key. |
| Hbase | Optimized for read, strict consistency, fast r/w with scalability. | Only range based scan, not good for data aggregation. |

**References**

* <https://github.com/BetaNYC/Bike-Share-Data-Best-Practices/wiki/Bike-Share-Data-Systems>
* <https://www.motivateco.com/use-our-data/>
* https://resources.zaloni.com/blog/top-streaming-technologies-for-data-lakes-and-real-time-data