

User Behavior Simulator

Git commit with doc: 71f1bc4

[HTML version](#)

1. Overview	1
2. Event Structure	1
3. Prerequisites	2
4. Usage	2
4.1. Kafka mode	2
5. Configuration	3

A tool to simulate different user behaviors for product viewing in an e-commerce environment, sending events to Snowplow collector or Kafka topic.

1. Overview

This simulator generates different patterns of product viewing behavior:

- ¥ Frequent views of the same product
- ¥ Long duration views
- ¥ Normal browsing patterns

There are two modes for sending events:

- ¥ Snowplow - Sends events directly to a Snowplow collector
 - ! All events are tracked using the Snowplow [Node.js Tracker](#) and sent to a local collector.
- ¥ Kafka - Sends events to a Kafka topic
 - ! Events are generated as JSON and sent to a Kafka topic.

2. Event Structure

The simulator (for Snowplow mode) generates Snowplow ecommerce events with the following schema:

- ¥ Event: `product_view`
- ¥ Schema: `iglu:com.snowplowanalytics.snowplow.ecommerce/product_view/jsonschema/1-0-0`
- ¥ Context: `Product` `information` using `iglu:com.snowplowanalytics.snowplow.ecommerce/product/jsonschema/1-0-0`

3. Prerequisites

1. Docker installed
2. Node.js installed
3. npm or yarn package manager
4. For Snowplow mode:
 - a. A running Snowplow collector (default: <http://localhost:9090>)
5. For Kafka (Redpanda) mode:
 - a. A running Kafka broker (default: localhost:19092)

4. Usage

4.1. Kafka mode

1. Step 1: Start the Redpanda services:

```
$ ./run.sh redpanda up
```

2. Step 2: Watch the logs for the [discounts-processor](#) service:

```
$ ./run.sh redpanda logs
```

3. Step 3: Run the simulator in [kafka](#) mode:

- a. To run the simulator in [long](#) mode, type:

```
$ ./run.sh long kafka
```

- b. To run the simulator in [frequent](#) mode, type:

```
$ ./run.sh frequent kafka
```

!

The [normal](#) mode is also available.

4. Step 4: Stop the Redpanda services:

```
$ ./run.sh redpanda down
```

5. Configuration

To configure aspects such as Snowplow collector endpoint, Kafka broker, and user behavior patterns, copy the file `config.sample.js` to `config.js` and edit it as needed.