

# MoMo SMS Data Processor — Report

## 1. Introduction to API security

This report documents the development, testing, and validation of a REST API for processing MoMo SMS transactions. The API allows clients to securely interact with transaction data and demonstrates efficient search operations using Python data structures.

The API uses **Basic Authentication** to restrict access. While suitable for a prototype:

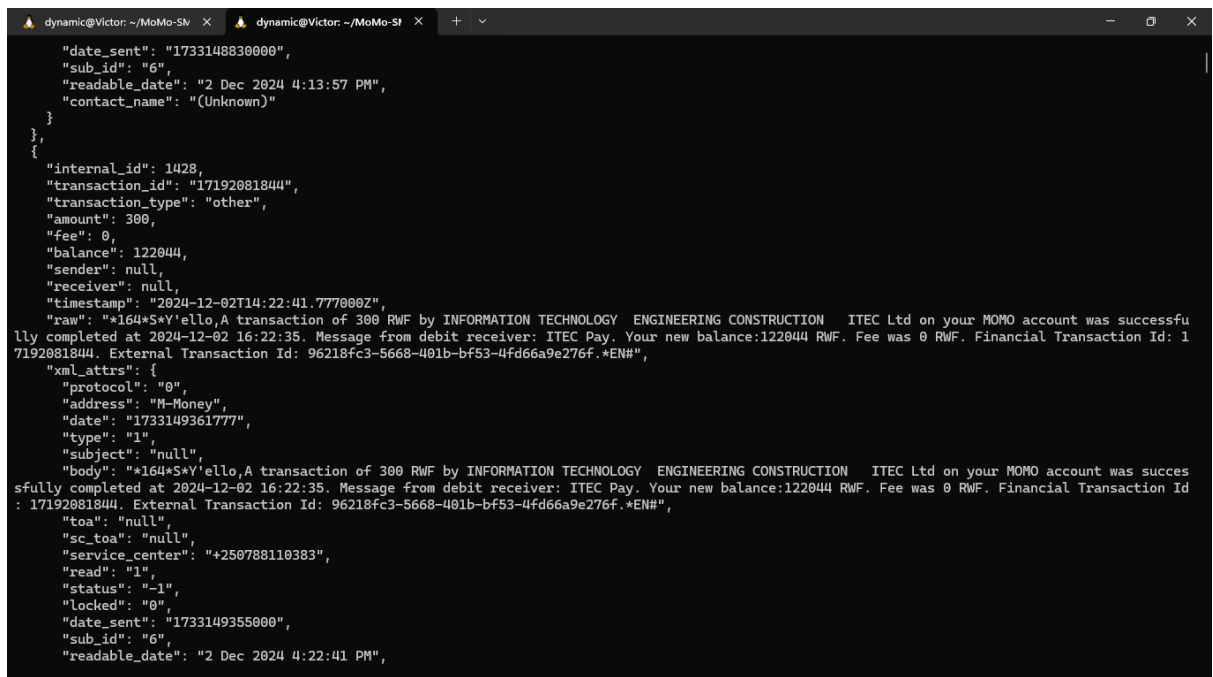
- Credentials are sent in every request (even if Base64-encoded).
- Weakness: Susceptible to interception over plain HTTP.

**Recommended improvements for production:**

- **HTTPS** for encrypted communication.
- **JWT (JSON Web Tokens)** or **OAuth2** for token-based, stateless authentication.
- Rate limiting and input validation to prevent misuse.

## 2. Documentation of endpoints

- GET /transactions — all transactions



```
"date_sent": "1733148830000",
"sub_id": "6",
"readable_date": "2 Dec 2024 4:13:57 PM",
"contact_name": "(Unknown)"
},
{
  "internal_id": 1428,
  "transaction_id": "17192081844",
  "transaction_type": "other",
  "amount": 300,
  "fee": 0,
  "balance": 122044,
  "sender": null,
  "receiver": null,
  "timestamp": "2024-12-02T14:22:41.777000Z",
  "raw": "*164*S*Y'ello,A transaction of 300 RWF by INFORMATION TECHNOLOGY ENGINEERING CONSTRUCTION ITEC Ltd on your MOMO account was successfully completed at 2024-12-02 16:22:35. Message from debit receiver: ITEC Pay. Your new balance:122044 RWF. Fee was 0 RWF. Financial Transaction Id: 17192081844. External Transaction Id: 96218fc3-5668-401b-bf53-4fd66a9e276f.*EN#",
  "xml_attrs": {
    "protocol": "0",
    "address": "M-Money",
    "date": "1733149361777",
    "type": "1",
    "subject": "null",
    "body": "*164*S*Y'ello,A transaction of 300 RWF by INFORMATION TECHNOLOGY ENGINEERING CONSTRUCTION ITEC Ltd on your MOMO account was successfully completed at 2024-12-02 16:22:35. Message from debit receiver: ITEC Pay. Your new balance:122044 RWF. Fee was 0 RWF. Financial Transaction Id : 17192081844. External Transaction Id: 96218fc3-5668-401b-bf53-4fd66a9e276f.*EN#",
    "toa": "null",
    "sc_toa": "null",
    "service_center": "+250788110383",
    "read": "1",
    "status": "-1",
    "locked": "0",
    "date_sent": "1733149355000",
    "sub_id": "6",
    "readable_date": "2 Dec 2024 4:22:41 PM",
```

- GET /transactions/{id} — a single transaction by ID

```
(dynamic@Victor)-[~/MoMo-SMS-data_processor]
$ curl -u admin:1234 http://localhost:8000/transactions/2
{
  "internal_id": 2,
  "transaction_id": "73214484437",
  "transaction_type": "payment",
  "amount": 1000,
  "fee": 0,
  "balance": 1000,
  "sender": null,
  "receiver": {
    "name": "Jane Smith 12845 has been completed at 2024-05-10 16"
  },
  "timestamp": "2024-05-10T14:31:46.754000Z",
  "raw": "TxId: 73214484437. Your payment of 1,000 RWF to Jane Smith 12845 has been completed at 2024-05-10 16:31:39. Your new balance: 1,000 RWF. Fee was 0 RWF.Kanda*182*16# wiyandikishe muri poromosiyo ya BivaMoMotima, ugire amahirwe yo gutsindira ibihembo bishimishije.",
  "xml_attrs": {
    "protocol": "0",
    "address": "M-Money",
    "date": "1715351506754",
    "type": "1",
    "subject": "null",
    "body": "TxId: 73214484437. Your payment of 1,000 RWF to Jane Smith 12845 has been completed at 2024-05-10 16:31:39. Your new balance: 1,000 RWF. Fee was 0 RWF.Kanda*182*16# wiyandikishe muri poromosiyo ya BivaMoMotima, ugire amahirwe yo gutsindira ibihembo bishimishije.",
    "toa": "null",
    "sc_toa": "null",
    "service_center": "+250788110381",
    "read": "1",
    "status": "-1",
    "locked": "0",
    "date_sent": "1715351498000",
    "sub_id": "6",
    "readable_date": "10 May 2024 4:31:46 PM",
    "contact_name": "(Unknown)"
  }
}
```

- POST /transactions — add a new transaction

```
(dynamic@Victor)-[~/MoMo-SMS-data_processor]
$ curl -u admin:1234 -X POST http://localhost:8000/transactions \
-H "Content-Type: application/json" \
-d '{"transaction_type": "deposit", "amount": 10000, "sender": {"name": "Gift"}, "timestamp": "2024-10-08T09:00:00Z"}'
{"message": "Transaction added successfully", "id": 1692}
```

- PUT /transactions/{id} — update a transaction

```
(dynamic@Victor)-[~/MoMo-SMS-data_processor]
$ curl -u admin:1234 -X PUT http://localhost:8000/transactions/2 \
-H "Content-Type: application/json" \
-d '{"amount": 15000}'
{"message": "Transaction 2 updated successfully"}
```

- DELETE /transactions/{id} — remove a transaction

```
(dynamic@Victor)-[~/MoMo-SMS-data_processor]
$ curl -u admin:1234 -X DELETE http://localhost:8000/transactions/5
{"message": "Transaction 5 deleted successfully"}
```

### 3. DSA Comparison — Search Efficiency

#### Setup

Transactions parsed from modified\_sms\_v2.xml (1691 records). Two search methods were compared:

1. **Linear Search:** Sequential scan of the list.
2. **Dictionary Lookup:** Use transaction IDs as keys in a Python dictionary (hash table).

## Reflection:

- Linear search time grows with dataset size ( $O(n)$ ).
- Dictionary lookup is nearly constant ( $O(1)$ ) thanks to hash tables.
- Alternative data structures for efficient search:
  - Binary search on a sorted list ( $O(\log n)$ )
  - B-Trees or database indexes
  - Hash tables for in-memory lookup

```
(dynamicⓈVictor)~[~/MoMo-SMS-data_processor]
$ python3 dsa/search.py
Loaded 1691 transactions

--- Benchmark Results (20 lookups) ---
Linear Search Total Time: 0.00192189 sec
Dictionary Lookup Total Time: 0.00001311 sec
Average Linear Search: 0.0000960946 sec per lookup
Average Dict Lookup : 0.000006557 sec per lookup

Reflection:
- Linear Search:  $O(n)$ , grows with dataset size.
- Dict Lookup:  $O(1)$  on average, independent of dataset size.
- Dict lookup is faster because Python dicts use hash tables.
- Alternatives: binary search on sorted data ( $O(\log n)$ ), B-Trees, database indexes.
```

## 4. Reflection on Basic Auth limitations

Basic Authentication provides a simple way to secure API endpoints by requiring a username and password for each request. It is easy to implement, test, and integrate with tools like `curl` or Postman, making it suitable for prototypes and small internal systems.

However, for production environments, stronger methods are preferred. Using HTTPS ensures that credentials are transmitted securely, while token-based authentication (JWT) or OAuth2 provides better control, session management, and scalability. These approaches enhance security without adding much complexity to client-server communication.