

GST INVOICE GENERATION AND VERIFICATION TOOL

Venkatesh D N¹, Saleem², Prof. Anitha Sandeep³

¹Student, Dept. of CSE, R V College of Engineering, Bangalore

²Student, Dept. of CSE, R V College of Engineering, Bangalore

³Assistant Professor, Dept. of CSE, R V College of Engineering, Bangalore

Abstract - The Project **GST Invoice Generation and Verification Tool** builds a standalone application to generate GST invoices in the JSON format using the GST offline utility tool and verify the JSON files generated.

And to develop APIs to upload the JSON files to the Sandbox environment using handshaking protocol. So far, Analysis of JSON files is done through the production environment which comes with a cost hence to avoid it and test all the test cases development and implementation of the API is been carried out. After completion of the API, verification of the JSON file can be carried out and used for analysis.

Key Words: GST, API, App Key, JSON, Vayana GSP Sandbox Environment, Auth token, Spring Boot.

1. INTRODUCTION

The goods and services tax abbreviated as GST is an indirect tax levied on all goods and services except Petroleum crude, motor spirit (petrol), high-speed diesel, and natural gas and aviation turbine fuel. The Goods and Service Tax is paid to the state government and central government by the consumers and the businesses selling the goods and services. It is mandatory to pay GST, everyone who supplies goods and services of transaction value exceeding Rs 20 lakh in a financial year, (Rs 10 lakh for some special category states). GST replaced indirect taxes such as VAT, customs duty, Excise, CST, Service Tax, Entertainment Tax with a single tax called the Goods and Services Tax. Earlier tax system had cascading effects on the service tax and VAT. The GST allowed us to have Cascading benefits hence we can avail input tax credit paid on inputs.

1.1 GST Types

GST can be broadly divided into 3 types.

1.1.1 CGST

Central Goods and Service Tax levied by the central government on the Intra-state supply of goods and services.

1.1.2 SGST

State Goods and Service Tax levied by the state government on the Intra-state supply of goods and services.

1.1.3 IGST

Integrated Goods and Service Tax levied by the central government on Inter-state supply of goods and services.

1.2 GST Returns

GST return is a document that contains the details of sales, purchases, tax collected on sales and tax paid on purchases.

There are around 15 GST returns. But we have considered GSTR1, GSTR2 and GSTR3B.

1.2.1 GSTR1

GSTR1 is a return details of all outward supplies made during a transaction period. Again it has sub forms – B2B, B2CL, B2CS, EXP, CDNR and HSN.

1.2.2 GSTR2

GSTR2 is a return details of all inward supplies made during a transaction period. It is auto-populated once GSTR1 is filed from the consumer.

1.2.3 GSTRB

GSTR-3B is a summary of all outward supplies made, inwards supplies made, input tax credit claimed.

The design approach for handling large data download (for more than 10000 invoices) is to 'chunk' the data into manageable bites. This means, the call to a GET of B2B Returns invoices will return a token. The developer can then use the token to determine the number of chunks and subsequently download each chunk independently; hence it is difficult to analyze the large data. The existing system of analysis of GST invoices involves fetching the data from the production environment provided by the GST server, which is costly. Current GST invoice analyzer system does not have test cases for more than 10000 invoices. This project aims to overcome these problems hence improving the scalability of GST analysis for more than 10000 invoices.

2. OBJECTIVE

- To mimic the real world GST network production environment into the development environment.
- Build APIs to generate, upload and verify along with handshaking protocols.

- To provide test cases that has more than 10000 invoices for the analyzing system.

3. METHODOLOGY

- Using the raw data of invoices, populate the excel sheet automatically via a python program script.
- Using the GST offline tool to convert data in excel format into the JSON format.
- AES/ECB/PKCS5 Padding is the Cipher used for encryption and decryption.

- HMACSHA256 is a type of keyed hash algorithm that is constructed from the SHA-256 hash function and used as a Hash-based Message Authentication Code (HMAC).

- Develop an API to upload the JSON file to the Vayana GSP Sandbox environment using handshaking protocol.

4. TOOLS USED

4.1 SPRING BOOT

Spring boot is a java application framework that lets us build java spring applications. It abstracted the problems with the Spring like – multiple setup steps, multiple configuration steps and multiple build and deploy steps. It features convention over configuration concept. In this project we have used spring tool suite to write APIs.

4.2 VAYANA GSP SANDBOX ENVIRONEMNT

VayanaGSP will provide a pass-through API Gateway to enable ASPs to access the GSTN for the purpose of uploading purchase and sales invoices and filing returns. It will expose the exact same APIs as provided by GSTN and hence will always be in sync with the capabilities exposed by GSTN. As an ASP one needs to study the API and authentication documentation provided by GSTN to build the requests to VayanaGSP.

4.3 POSTMAN CANARY

In our project, Postman is used to handle HTTP send and receive requests, POST data to the Vayana GSP Sandbox server.

5. IMPLEMENTATION

The Implementation of the GST Invoice Generation and Verification tool can be achieved in two steps. They are

- JSON Generation
- API Implementation.

5.1 JSON GENERATION

In this Step, programming scripts are written to populate Excel sheets of size 10000 records each corresponding to each form of GSTR1, GSTR2A, GSTR 3B in Python programming language. These Excel Sheets are converted into the JSON Files, by uploading them into the GST Offline utility tool[3]. This step is necessary since the GST Sandbox Environment accepts only JSON Files. These JSON files are compared and validated against the corresponding JSON schema provided by the GSTN. The complete process is shown in Figure 1.

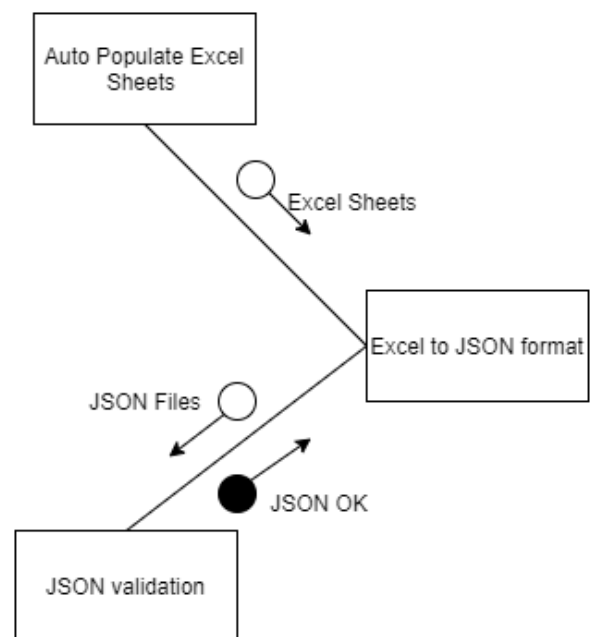


Fig -1: JSON Generation

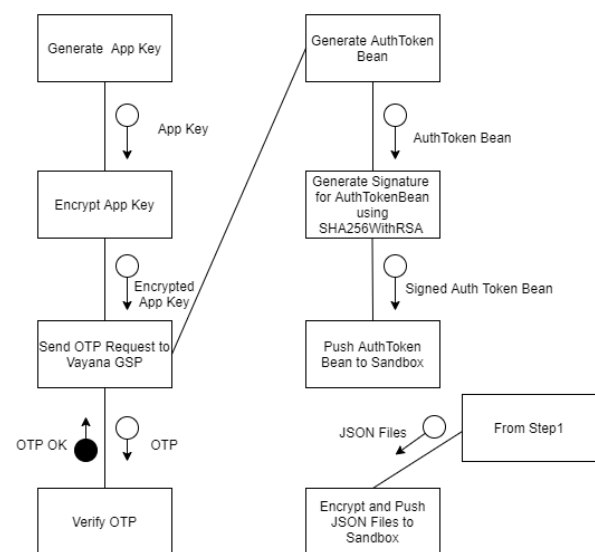


Fig -2: API Implementation

5.2 API IMPLEMENTATION

In this step, App Key is generated using a Regular Expression pattern. Encrypt it with Public Key that can be generated using the "SHA256WithRSA" algorithm

. An HTTP Request is made for the OTP to Vayana GSP Sandbox Environment [4] where payload includes the Encrypted App Key. The received OTP is sent for the OTP Verification. Auth Token is generated using Customer ID, Client Id, Transaction Id, Time Stamp, GSTIN, API Action as given below.

```
auth_token=v2.0:<cust_id>:<client_id>:<txn_id>:<timestamp>:<gstin>:<api_action>.
```

It is signed with a base64 encoded signature using the private key given by vayana. It is sent to Sandbox to create a session between the client and the GSTN. This complete process is to establish a connection with the Sandbox. In the next step, SHA256withRSA Encrypted JSON files are sent to the Sandbox for further processing. The complete process is given in figure 2.

6. CONCLUSION

We will be able to optimize the analysis and work on all the test cases as the sandbox environment allows us to download large data in a single file. API Calls to the Sandbox environment is free of cost since it is a testing environment. We can generate test cases for large dataset using program script.

ACKNOWLEDGEMENT

Prof. Anitha Sandeep, Assistant Professor, Department of Computer Science and Engineering, R.V. College of Engineering, Bangalore.

REFERENCES

- [1] <https://developer.gst.gov.in/apiportal/> [GST Developer portal]
- [2] <https://cleartax.in/s/gstr-1> [GST Concepts]
- [3] <https://www.gst.gov.in/download/returns> [GST Returns Offline Tool]
- [4] <https://vayana.com/gst/gsp/> [Vayana GSP Sandbox Environment]

BIOGRAPHIES



Venkatesh D.N is a Final year Computer Science and Engineering Student at R V College of Engineering.



Saleem is a Final year Computer Science and Engineering Student at R V College of Engineering.