

# Returns API Overview

## About the Returns API Catalog

Returns API refers to a collection of APIs that enable interaction with the GST System for the purpose of filing of various return forms. In the current release, the API catalog includes APIs for:

1. Filing of GSTR1
2. Filing of GSTR2
3. Filing of GSTR3
4. Some Common utility APIs that cut-across all Return forms

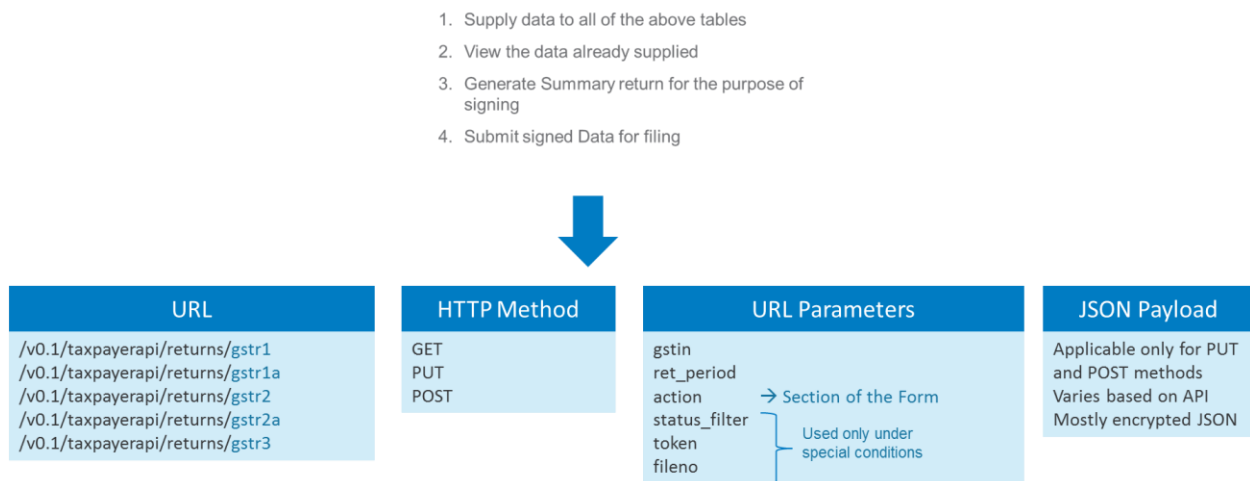
The following sections provide a brief overview of the above sets of API for the first time reader.

## About API Model for Return Forms

The objective of the GSTR1 APIs is to enable the following distinct set of interactions:

1. Supply data to each section of the form
2. View the data already uploaded (for purpose of reviewing)
3. Generate Summary return for the purpose of signing
4. Submit digitally signed form for final filing

The design shall accomplish all of these through APIs which generally have a 4 part construct:



While the URL of the API determines the Form on which the system must operate on, the URL parameters **establish the context** such as GSTIN ID, Return period etc. The URL parameters also establish the **scope**

---

**of the API** - like for example the **action** parameter shall determine the section of the form on which the API must operate on. The API model also includes some API specific parameters such as filter to be applied to **further narrow down the scope** of the API.

The following section describe the model for each of the above mentioned interactions.

### Step 1: Supply data to each section of the form

This task is accomplished using the 'Save' API for the corresponding form. The design of the Save API allows the developer to **submit data for all the sections** of the form in a single JSON format. For example, in the case of GSTR1, the Save API will accept a single JSON which supports a data structure for all the sections like B2B invoices, B2C invoices, B2C summary, Exports, Credit/Debit Notes etc.

However it is important to note that while the API is *capable* of processing all the sections of the form in a single call, each of the section themselves are *optional*. This means, the user can use the API to only supply data for one section of the form like B2B invoices alone in a single call.

#### *Handling multiple calls:*

Developer can make multiple calls to the Save API, each time submitting data only for one section of the form. Developer may even supply partial data for each section in each call. For example, submission of B2B invoices, can be done in batches of 50 or 100 invoices at a time.

The API behavior will be the store the **UNION** of each such call, appending the payload of each call with the already existing data. For e.g. the user may upload 10 invoices in the first call and a second call with a different set of 12 invoices. Resulting state will be a set of 22 invoices stored against the return – provided of course the invoices submitted were non-intersecting. See following section for

#### *Handling submission of overlapping data across call:*

While the general approach is to persist the UNION of data across calls, the UNION approach is only till a certain level of granularity. Data below this level of granularity will be replaced instead of appended. For e.g. in the case of B2B invoices, the granularity will be at the Invoice level indicated by the Invoice Number. Any subsequent call for the same Invoice Number will REPLACE the already existing data instead of APPENDING.

For instance, if the first call had a certain invoice with 2 line items and subsequent call carried 3 different line items, then the second call will simply REPLACE the earlier invoice instead of appending. The level of granularity for each section of the form may vary. The following table lists the level of granularity for each of the sections of GSTR1.

Section	Granularity	Remarks
B2B	Invoice Number	Subsequent call will replace all the items within the invoice.
B2C Large	Invoice Number	
B2C Small	State	All the HSNs within the State will be replaced

Credit/Debit Note	Credit Note No	Mostly 1 CN/DN will correspond to 1 invoice.
Nil Rated	None	Subsequent call will replace all 4 records
Exports	Invoice Number	Subsequent call will replace all the items within the invoice.
AT Liability	Document No	All the HSNs within the Document No will be replaced
AT Paid	Invoice Number	Most cases 1 Invoice will be 1 Document No. But if there are more, all invoices will be replaced
eCommerce	GSTIN of eCom + Nature of Supply (Inter Vs Intra)	All the merchant IDs / HSNs for that eCom GSTIN will be replaced
HSN Summary	Nature of Supply	All the HSN with in a nature of supply will be replaced

#### *Asynchronous processing:*

GST system is expected to handle not only large volume of data but also high concurrency. Design best practices dictate that such systems be designed to handle requests asynchronously. It should be noted that asynchronous processing design is not to save on any compute or other infrastructure. In fact, in most cases asynchronous processing increases the infrastructure requirement. However the benefit of such asynchronous design increases availability and stability manifold.

This means the call to Save API will return only a token instead of the actual status. The developer is expected to call-back for the status or error lists after some time. The time gap between submissions on availability of status may vary based on the size of the data and also the concurrency. An optimal strategy for call backs for status will be published and refined in due course.

#### **Step 2: View the data already uploaded (for purpose of reviewing)**

This interaction to download already uploaded data can be accomplished using a set of GET APIs. Unlike the POST API uploading the data, the GET APIs will work at each section level to enable the developer to only download the sections he/she is interested in. This is accomplished by specifying the Section in the action parameter.

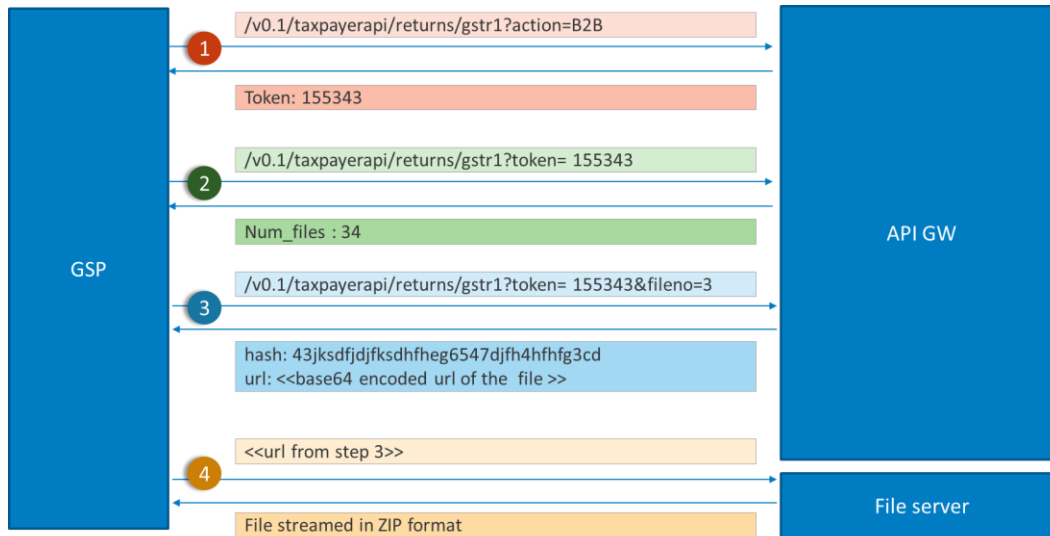
Further, some sections will also support additional filters to further refine the scope of the data retrieved. For example, the APIs will support download of only actionable invoices or invoices for a specific counter party etc.

#### *Handling large data sets:*

Some of the sections of the form can potentially carry a large data set – for example, lakhs of B2B invoices. Such large volumes of data cannot possibly be returned in a single call. The design approach for handling large data download is to ‘chunk’ the data into manageable bites. This means, the call to a GET of B2B

invoices will return a token. The developer can then use the token to determine the number of chunks and subsequently download each chunk independently.

The following figure shows the interaction for handling large volume of data.



### Step 3: Generate Summary return for the purpose of signing

The Returns in GST system need to digitally signed before submission. The Business process also dictates that the GST system make necessary provision for periodic upload of data. This means, that the Tax Payer would have already uploaded all the necessary data before filing of returns. Signing the data would then mean downloading all the invoices again and resubmitting it with digital signature. This obviously is a significant and avoidable overhead.

The solution therefore is to enable Tax Payer to download a summary generated based on already uploaded data and asking the user to submit the signed summary instead of the entire data. For example, the if the Tax Payer has submitted 1000 invoices across 10 dealers, the summary would be generated containing not 1000 rows – but just 10 rows – one for each counter party.

The above concept is illustrated in the figure below.

