Return Filing through API

A normal Tax payer has to file GSTR1, GSTR2 & GSTR3, GSTR1A every month to become GST complaint. For these details rules and flows are defined in draft rule and laws. These are yet to be finalized but generic workflow can be described as below. Rules talks about auto population of supplier data in GSTR2A and talks about acceptance and rejection in GSTR2A, to exclude/include invoices in GSTR2 return. From the API perspective we have made it similar. We have only kept GSTR2 API, to handle auto population, acceptance/rejection/modification/pending. Below are the scenarios for filing above returns through API.

Happy Scenarios:

Taxpayer (TP) is filing GSTR1 on or before time:

1. TP will use saveGSTR1 API to upload invoices.
2. Once finalized, TP will call submit GSTR1. Before submit GSTR1, TP can call getGSTR1Summary to know summarize details as it will freeze the position of Tax payers and now no more changes is possible in the return. These steps will do all the validation and once all validations are successful. Then later steps can be done in synchronous mode.
3. TP will call getGSTR1Summary.
4. TP will call fileGSTR1 with payload received from above payload and get ARN number is synchronous mode.

Taxpayer (TP) is filing his GSTR2 on or before time and all his Counter Parties (CP) have already filed GSTR1:

1. TP will call GSTR2-getB2B, getB2BA, getCDN, getCDNA, getISD, getTCS, getTDS using filter action_required='Y' to get auto populated data. This data will be equivalent to GSTR2A data mentioned in rules and forms.
2. TP will accept/reject this data using saveGSTR2 API. TP will also provide ITC details along with acceptance/rejection/Modification through this API. In case of accept/reject only checksum value has to be sent along with appropriate flag and ITC details. In case of Modify, complete invoice detail has to be sent along with ITC details.
3. If TP wants to keep any invoice pending, he can use pending flag for that. For implicit clarity, with every invoice, system expect a flag (Accept-A, Reject-R, Modified-M, Uploaded-U, Pending-P). In B2BA and CDNA only accept/rejection is allowed.
4. TP can also add missing invoices using saveGSTR2 API.
5. Once finalized, TP will call submit GSTR2. TP can call getGSTR2Summary before submit GSTR2 to know summarize details.
6. TP will call getGSTR2Summary. (Summary will have all the invoices accepted/rejected/pending/uploaded/Modified. But for liability calculation only accepted/uploaded/Modified invoices will be considered.)
7. TP will call fileGSTR2 with payload received from above payload and get ARN number is synchronous mode.

Taxpayer (TP) is filing his GSTR1A before filing of GSTR3:

Note: GSTR1A filing is optional and if it is not filed then all auto populated data will be auto populated in next month's GSTR1.

1. TP will call GSTR1A-getB2B, getB2BA, getCDN, getCDNA, using filter action_required='Y' to get auto populated data.
2. TP will accept/reject this data using saveGSTR1A API. In case of accept/reject only checksum value has to be sent along with appropriate flag.
3. If TP wants to keep any invoice pending, he can use pending flag for that. For implicit clarity, with every invoice, system expect a flag (Accept-A, Reject-R,Pending-P)
4. Once finalized, TP will call submit GSTR1A. TP can call getGSTR1ASummary before submit GSTR1A to know summarize details.
5. TP will call getGSTR1Summary. (Summary will have all the invoices accepted/rejected/pending/ Modified. But for liability calculation only accepted/modified invoices will be considered.)
6. TP will call fileGSTR1A with payload received from above payload and get ARN number is synchronous mode.

Taxpayer (TP) is filing his GSTR3:

1. TP will first call generateGSTR3 and will get a ref_id. This step will be in asynchronous mode and it will generate summary.
2. TP will than call get GSTR3 Details and would be able to view the GSTR3 summary which would be consolidated summary of GSTR1 and GSTR2

   Sections in GSTR3 PART A such as (1) Turn Over Details, (2) Outward Supplies, (3) Inward Supplies, (4) Input Tax Credit, (5) Addition and reduction of amount in output tax for mismatch and other reasons, (6) Total tax liability, (7) Credit of TDS and TCS would be auto populated from GSTR1 and GSTR2 forms. Interest Liability and Late fee section would be system calculated and would be shown.
3. TP will then call "submit liabilities and interest" API to submit interest liabilities by verifying the system calculated interest liabilities and will input the specific sections (1) Interest Liability Carry Forward (2) Delay in Payment of Tax (3) On account of ITC reversal (4) Other interest. This is a mandatory step to be followed.
   System will now post tax liability, interest liability and late fee in respective ledger of TP. This step will be asynchronous.
4. TP will call *getCashITCBalance* for retrieving the running cash balance, running ITC balance and Provisional Component (by tax period) of Running ITC Balance.
5. TP will then offset his liabilities by calling "Set off liability" through cash or ITC. Partial payment can be made only once and invalid filing can be only done once.
6. TP can submit refund claim using "Submit Refund Claim", if he has cleared his liabilities and have enough cash balance. This is an optional step.
7. TP will call getGSTR3Details with final flag as "Y", to get the payload of all section for GSTR3 to sign.
8. TP will call fileGSTR3 with signed payload received from above step and get ARN number in synchronous mode.

Taxpayer (TP) is filing his GSTR3B:

1. TP will first call Save GSTR3B data to save tax and ITC information along with interest detail.
2. TP will call Get GSTR3B Details to know tax payable, liability id, late fee, which will be now auto populates in tax payable section. As now there will be no Submit API so ITC Ledger API will only provide Last balance. To get the current balance, Application has to add current ITC amount into it.

   Example: Suppose in month June user has ITC balance of 100 Rs. In Month July, GSTR3B has 200 Rs. as ITC. So, Application will call "Get Cash ITC Balance". It will return 100 Rs. Application need to add 200 in to it. So currently month balance for user will be 300.
3. TP will now offset liabilities using "Offset Liability GSTR3B data" by putting tax payment details. Liability id could be "0" as liability will be posted after offset only.
4. TP will again call Get GSTR3B Details to get final return.
5. TP will then call File GSTR3B API with the payload received from above step.

Taxpayer (TP) is filing his GSTR4:

1. In normal scenario GSTR4 will be auto populated from GSTR1 but for next two quarters as notified it will not be auto populated. So, get B2B API will have no auto populated data.
2. In current version no TDS data will be auto populated, so it will not be part of save request.
3. TP will call save GSTR4 API (In current version all the sections can be saved together i.e. If Tax payer has data in B2B, B2BUR, CDNR, CDNUR, IMPS, TXOS, AT, TXPD sections then he has to fill all the sections and save. Individual section save is not available in this version.) If Tax payer want to update/delete any information, he has to send all data along with updated records, so essentially every save request will overwrite previous save request. In this version it is a slight departure from our API approach. Next versions it will be in line with other APIs.

   Save call will be asynchronous and TP will get ref_id back. TP can check status of save using "get Return Status" API as done for other APIs.
4. Once TP has finished saving all the data, he will call GSTR4 summary to get summary. He can verify the summary of saved records.
5. Once TP is satisfied with summary, he can call submit GSTR4 API. This API will freeze TP's position and he will not be able to change anything in return after this call. It will be an asynchronous call and TP will get a reference id. Status of the request can be checked using similar to save API.
6. After submit is successful, TP need to call "GSTR4 Summary" (this summary post submit will fetch both payable and paid liability details as well in case of "Set Off Liability") and get payable liability details.
7. After this TP need to call "Set Off Liability" and set off liability can be called only once for current quarter, so TP has to set off all his liabilities completely.
8. TP will call again get summary API.
9. TP will call file API using above summary.

Taxpayer (TP) is filing his GSTR6: (Updated Flow after delinking of GSTR6 with GSTR1 & GSTR5)

1. TP will call get B2B, B2BA, CDN & CDNA. (All GSTR1 data filed before delinking will be auto populate in GSTR6 of corresponding month and will be available through get GSTR6 APIs.)
2. If any data is available through above APIs
    - TP will verify the data and delete/Modify as per need.
    - TP will upload all the missing invoices if any.

   If there is no data available
    - TP can call GSTR6A get API to get Auto Populated data from GSTR1.
    - TP can use this data to prepare payload for GSTR6 and upload it in to GSTR6 using save GSTR6 API.
      Or
      TP can completely ignore auto populated data and upload its own invoice data.
3. TP can distribute eligible/ineligible ITC credit in ISD details section of save GSTR6. TP can also include data of "Table 8" of GSTR6 form in this section. As ITC credit of different heads can be distributed in different head like IGST can be distributed as CGST, so there is different attribute like iamti (IGST distributed as IGST), iamts (IGST distributed as SGST) etc. In "ITC Details" section of getITC API all IGST, CGST, SGST will be clubbed and total value will be auto populated.
   Note: - In case of distribution, If ISD state code is same as Unit State code then Allowed amounts are iamti, iamtc, camtc, iamts, samts and cess. If ISD state code is different from Unit State code then Allowed amounts are iamti, camti, samti and cess. Table 4 of GSTR6 will be auto populated. 4a data will come from table 3 and table 6a. 4b and 4c data will come from table 5 and table8.
4. For amendment B2BA, CDNA, ISDA section of save API can be used.
5. TP can now call get Summary API and verify the details.
6. Once TP is satisfied with the summary, he can "Submit". After "Submit", GSTR6 will be freeze and no further update will be possible. But if on Submit TP gets an error message that "Total ITC received in 4(a) should be equal to the total ITC Distributed in 4(b)+4(c) at individual tax/cess level.", then TP can call "Get ITC" API and can distribute eligible/ineligible ITC credit in ISD details section of save GSTR6.
7. Note: getITC API will only return correct total value after submit. Before that it will not return correct ITC value. Idea is that GSP/ASP will build validation in his application to inform taxpayer if there is some issue in distribution. So, in GSP application Taxpayer should have a view to get total "ITC" available for distribution and "Total Distributed" amount at any time.
8. After "Submit" TP can call get Summary or get Late Fee API to know about "Late Fee" and debit id.
9. TP can call "Offset Late fee" API to set off any late fee.
10. TP can call "Get Summary" API.
11. TP can call "File GSTR6"

Note: After delinking there will be no auto population either from GSTR6 to GSTR1 or GSTR1 to GSTR6 (Before Delinking GSTR1 file data will be available as auto populated for ease of Taxpayer). Auto populated data from GSTR1 will be available through GSTR6A API. GSTR6A auto populated invoices can be uploaded in GSTR6. GSTR6 Tax payer will be able to upload all rejected, pending invoice of previous periods (before delinking) in current period. All previously uploaded or accepted/Modified invoices /Credit/Debit note cannot be uploaded in B2B and CDN section of next month.

Taxpayer (TP) is filing ITC04:

1. TP will call save ITC04 API to save records.
2. TP will call get Summary ITC04 once all records have been saved.
3. TP can change any record after calling the get Summary API also. Records cannot be edited after File.
4. TP will call file ITC04 API to file.
5. TP can call “get Invoices” API anytime to know the details of records saved by TP.

Taxpayer (TP) is filing ITC03:

1. There are 2 sections 4A/4B in ITC03. TP, who has opted for composite scheme can fill for both 4A/4B sections and regular TP can only fill for 4B section.
2. TP will use Save ITC03 API to upload the invoices.
3. After that TP can call Get ITC03 API to verify all uploaded invoices.
4. Before Offset API call, TP can call “Close API” to clear all uploaded data if required. This can be used in case TP wanted to withdraw it declaration.
5. TP will call Offset ITC03 Liability API to set off all his liabilities completely.
6. TP will call again Get summary API.
7. TP will call file API using above summary.

Taxpayer (NRTP – Non-Resident Tax Payers) is filing GSTR5:

1. TP will call Save GSTR5 API (for all sections B2B, B2BCL, B2BCS, CDN, CDN- Unregistered User, IMPG,) to upload invoices.
2. Once finalized, TP will call submit GSTR5. Before submit GSTR5, TP can call Get GSTR5 Summary to know summarize details as it will freeze the position of Tax payers and now no more changes is possible in the return. These steps will do all the validation and once all validations are successful. Then later steps can be done in synchronous mode.
3. After submit is successful, TP need to call “GSTR5 Summary” (this summary post submit will fetch both payable and paid liability details as well in case of “Set Off Liability”) and get payable liability details.
4. TP also get the payable liability details from “Get Tax Liab” API after submit GSTR5.
5. After this TP need to call “Offset Liability” and this API can be called any number of times till TP has to set off all his liabilities completely.
6. TP will call again get summary API.
7. TP will call file API using above summary.

Taxpayer (TDS TP) is filing GSTR7:

1. TP will call save GSTR7 API to save records.
2. TP will call Get TDS once all records have been saved.
3. TP will call Get Checksum once all records have been saved to know the checksum against each saved record (necessary for deletion of any record if needed).
4. For amendment TDSA section of save API can be used.
5. TP can now call Get Summary API and verify the details.

6. Once TP is satisfied with the summary, he can call “Proceed to File” API. After “Proceed to File”, GSTR7 will be freeze and no further update will be possible.

   Note: - “Proceed to File “should be immediately followed by “File GSTR7” as after 24 hrs. liability get altered and will affect the summary also i.e., after 24hrs TP has to again call Get Summary API and then “Proceed to File” API.
7. After “Proceed to File” TP can call Get Summary to know the offset liability details.
8. TP can call “File GSTR7” API to set off liability if any and file GSTR7.

   Note: - For GSTR7 TCS Users, additional header details would be there as mandatory attributes at the time of save, proceed to file and file API calls as mentioned on **https://developer.gst.gov.in/apiportal/taxpayer/returns.**

Taxpayer (TCS TP) is filing GSTR8:

1. TP will call save GSTR8 API to save records.
2. TP will call Get TCS once all records have been saved.
3. TP will call Get Checksum once all records have been saved to know the checksum against each saved record (necessary for deletion of any record if needed).
4. For amendment TCSA section of save API can be used.
5. TP can now call Get Summary API and verify the details.
6. Once TP is satisfied with the summary, he can call “Proceed to File” API. After “Proceed to File”, GSTR8 will be freeze and no further update will be possible.

   Note: - “Proceed to File “should be immediately followed by “File GSTR8” as after 24 hrs. liability get altered and will affect the summary also i.e., after 24hrs TP has to again call Get Summary API and then “Proceed to File” API.
7. After “Proceed to File” TP can call Get Summary to know the offset liability details.
8. TP can call “File GSTR8” API to set off liability if any and file GSTR8.

   Note: - For GSTR8 TCS Users, additional header details would be there as mandatory attributes at the time of save, proceed to file and file API calls as mentioned on **https://developer.gst.gov.in/apiportal/taxpayer/returns.**

Taxpayer (Counter TP receiving TDS- TCS Credit) is filing GSTR2X:

1. TP will call Get TDS-TCS Credit Received Details API to get all auto populated data from TDS/TCS users.
2. TP can only Accept/Reject auto populated invoice in Save TDS-TCS Credit Received Details API.
3. TP can now call Get TDS-TCS Credit Received Details API and verify the details of Acceptation/Rejection of TDS/TCS/TDSA/TCSA invoices.
4. Once TP is satisfied with the summary, he can call “Proceed to File” API. After “Proceed to File”, GSTR2X will be freeze and no further update will be possible.

   Note: - “Proceed to File “should be immediately followed by “File GSTR2X” as after 24 hrs. liability get altered and will affect the summary also i.e., after 24hrs TP has to again call Get Summary API and then “Proceed to File” API.
5. After “Proceed to File” TP can call Get Summary.
6. TP can call “File GSTR2X” API

Note: - For GSTR2X Users, additional header details would be there as mandatory attributes at the time of save, proceed to file and file API calls as mentioned on **https://developer.gst.gov.in/apiportal/taxpayer/returns.**

Non-Happy Scenarios:

Taxpayer (TP) is filing GSTR1 after filing by all Counter Parties:

1. TP will call GSTR1-getB2B, getB2BA, getCDN, getCDNA using filter action_required='Y' to get auto populated data. This data will be equivalent to GSTR1A data mentioned in rules and forms.
2. TP will delete all earlier uploaded invoices, which has come now as auto populated (as buyer has uploaded those) using above APIs.
3. TP will accept/reject this data using saveGSTR1 API. In case of accept/reject only checksum value has to be sent along with appropriate flag.
4. If TP wants to keep any invoice pending, he can use pending flag for that. For implicit clarity, with every invoice, system expect a flag (Accept-A, Reject-R, Pending-P). In B2BA and CDNA only accept/rejection is allowed.
5. TP can also add missing invoices using saveGSTR1 API.
6. Once uploaded, TP can call getGSTR1Summary anytime to know summarize details.
7. Once finalized, TP will call submit GSTR1.
8. TP will call getGSTR1Summary. (Summary will have all the invoices accepted/rejected/pending/uploaded. But for liability calculation only accepted/uploaded/Modified invoices will be considered.)
9. TP will call fileGSTR1 with payload received from above payload and get ARN number is synchronous mode.

Taxpayer (TP) is filing his GSTR2 on or before time and all his Counter Parties (CP) have not already filed GSTR1:

1. TP will call GSTR2-getB2B, getB2BA, getCDN, getCDNA, getISD, getTCS, getTDS using filter action_required='Y' to get auto populated data. This data will be equivalent to GSTR2A data mentioned in rules and forms. This data will contain both "uploaded but not Submitted(UNS)" and "Uploaded and submitted data(US)". This data can be identified by seeing the "Counter Party filing status attribute against ctin in above get API's output".
2. TP will accept/reject/Modify and provide ITC details for UF data using saveGSTR2 API as explained in above scenario and upload selected UNS data as missing invoices. In UNS case complete invoice data along with ITC details need to be uploaded.
3. If TP wants to keep any invoice pending, he can use pending flag for that. For implicit clarity, with every invoice, system expect a flag (Accept-A, Reject-R, Modified-M, Uploaded-U, Pending-P). In B2BA and CDNA only accept/rejection is allowed.
4. TP can also add more missing invoices, which he did not get as UNS data, using saveGSTR2 API.
5. Once uploaded, TP can call getGSTR2Summary anytime to know summarize details.
6. Once finalized, TP will call submit GSTR2.

7. TP will call getGSTR2Summary. (Summary will have all the invoices accepted/rejected/pending/uploaded/Modified. But for liability calculation only accepted/uploaded/Modified invoices will be considered.)
8. TP will call fileGSTR2 with payload received from above payload and get ARN number is synchronous mode.

Taxpayer (TP) is resubmitting his invalid GSTR3:

1. TP will call get GSTR3 Details and would be able to view the GSTR3 summary which would be consolidated summary of GSTR1 and GSTR2

   Sections in GSTR3 PART A such as (1) Turn Over Details, (2) Outward Supplies, (3) Inward Supplies, (4) Input Tax Credit, (5) Addition and reduction of amount in output tax for mismatch and other reasons, (6) Total tax liability, (7) Credit of TDS and TCS would be auto populated from GSTR1 and GSTR2 forms. Interest Liability and Late fee section would be system calculated and would be shown.
2. TP will call *getCashITCBalance* for retrieving the running cash balance, running ITC balance and Provisional Component (by tax period) of Running ITC Balance.
3. TP will then offset his liabilities by calling "Set off liability" through cash or ITC.
4. TP can submit refund claim using "Submit Refund Claim", if he has cleared his liabilities and have enough cash balance. This is an optional step.
5. TP will call getGSTR3Details with final flag as "Y", to get the payload of all section for GSTR3 to sign.
6. TP will call fileGSTR3 with signed payload received from above step and get ARN number in synchronous mode.

Some Typical Scenarios:

1. If TP files GSTR1 and its CP makes changes or add any invoices then it will be auto populated in TP's GSTR1A for that month only if TP has not filed GSTR3 of that month otherwise in next month's GSTR1.
2. Similarly, if TP has filed GSTR2 and CP has added/modified/rejected any invoice, then it will be auto populated in the next month's GSTR2 of TP.
3. GSTR1A filing is optional and can be filed before GSTR3. If GSTR1A is filed then its impact will be available in GSTR3 of same month. If GSTR1A is not filed, then all auto populated invoices will go in the next month GSTR1 of TP.
4. Supplier and buyer can upload same invoice, it will happen in late filing scenarios. But at the stage of submit GSTR1/submit GSTR2, whosoever will Submit first, his version will be accepted and other will get a duplicate error on submit, such uploaded invoices would need to be deleted and counterparty invoice accepted/rejected before resubmitting return. In case of normal scenario, if buyer gets an auto populated invoice, and he try to save the same invoice details, system will not accept it as save API stage itself.
5. If in an invoice there are five items having same HSN code and it is uploaded in GST system. GST system will not club it. It will remain 5 items and available to counter party as it is.

6. Rejection/Acceptance/Modification will happen on invoice level. So line item wise Rejection/Acceptance/Modification is not possible. In case of mismatch in line items attribute value like Taxable Value, then one change that line item and mark invoice as “M” and save complete invoice data.
7. Till 10th, all supplier uploaded data will be available to receiver for information but receiver will not be able to save those invoices in its GSTR2 with any action. After 10th also, all data will be available through get calls along with flags whether counter party has filed return or not. If cfs flag has "Y" value then receiver has to take accept/reject/modify action and save GSTR2 otherwise all uploaded data by seller, if it is valid, should be saved by Buyer. This buyer uploaded data will only visible to "Seller" if "Buyer" file his GSTR2. Otherwise not. Hope this will clarify.
8. For taking action in GSTR2, one should use “checksum” received through get calls. Buyer application will not be required to generate their own “checksum” for it. Buyer application for the purpose of automatic reconciliation can generate “checksum” on its purchase register data (algorithm to generate it is already shared). But for taking action it is not at all mandatory.