

[Download PDF Version](#)

## **TaxPro GST User Control Library**

---

**User Guide  
Ver 0.1 (Under Construction)**

## Table of contents

Contact Us .....	4
Overview .....	5
GST Filing Flow .....	6
Assemblies and Namespaces .....	7
Where to Start? .....	8
Support Multiple TaxPayer .....	9
URL Based APIs .....	13
TaxPro GSP API .....	15
Generate ASP Certificate .....	17
GetApiSetting .....	18
GetKey API .....	19
GetKey API Sequence Diagram .....	21
Sample C# Code .....	22
GST Pass Through API .....	23
GSP Error Codes .....	24
GSP Utility APIs .....	26
Public APIs .....	27
Search TaxPayer API .....	28
GST API .....	29
APISession .....	30
Auth API Methods .....	33
GSTR API Methods .....	34
Json Models .....	35
List of Controls .....	36
ToolBox Controls .....	37
ProgressPanel .....	38
SummaryPanel .....	40
SendSMS .....	41
GSTSession .....	42
Form3BOffsetLiability .....	43
GSTR Controls .....	44
GSTR1Upload .....	45
GSTR1Return .....	46
GSTR2Upload .....	47
GSTR2Return .....	48
GSTR1AReturn .....	49
GSTR1AUpload .....	50
Common Controls .....	51
EditAPISetting .....	52
APIAuthorization .....	53
Returns .....	54
ReturnStatus .....	55
AdvTaxPaidDetails .....	56
PartialITCDetails .....	57
TDSDetails .....	58
Ledgers .....	59
CashLedger .....	60
ITCLedger .....	61
LiabilityLedger .....	62
UtilizeCash .....	63
UtilizeITC .....	64

TaxProGST Core .....	65
GSTRTools .....	66
Database Models .....	67
Sample Project .....	68

# Contact Us

**Corporate Office:**

Chartered Information Systems Pvt. Ltd.  
"Chartered House",  
West of Lata Mangeshkar Musical Park,  
Bhandara Road, NAGPUR - 440 035 INDIA  
**Phone:** 0712 - 663 8888 (100 Lines), 9225245088  
**Fax:** 0712 - 663 8899  
**Email:** [support@taxprogs.co.in](mailto:support@taxprogs.co.in)

**Mumbai Office:**

Chartered Information Systems Pvt. Ltd.  
#110, 1st Floor, Building No. 3, Hari OM Plaza,  
M. G. Road, Near National Park, Borivali (E), MUMBAI -400 066 INDIA  
**Phone:** 022- 28955888 / 65228288, 9766491466  
**Email:** [taxpromumbai@taxpro.co.in](mailto:taxpromumbai@taxpro.co.in)

**Delhi Office:**

Chartered Information Systems Pvt. Ltd.  
#222-223, Durga Chambers, 1333-34, D.B. Gupta Road,  
Karol Bagh, NEW DELHI - 110 005 INDIA  
**Phone:** 011 - 45037177, 45020850, 9871265511 (Kapil)  
**Email:** [taxprodelhi@taxpro.co.in](mailto:taxprodelhi@taxpro.co.in)

## Kolkata Office

**Chartered Information Systems Pvt. Ltd.**  
Room No D-7, 3rd Floor, Beside Glendear House, 219/C  
Old China Bazar Street, KOLKATA - 700 001 INDIA  
**Phone:** 033 - 40078356  
**Email:** [taxprokolkata@taxpro.co.in](mailto:taxprokolkata@taxpro.co.in)

## Bengaluru Office

**Chartered Information Systems Pvt. Ltd.**  
#207-B, Wing A, Brigade Majestic,  
1st Main (Kalidas) Road, Gandhi Nagar,  
BENGALURU - 560 009 INDIA  
**Phone:** 080 - 40921639, 9972932213 (Raghav)  
**Email:** [taxprobangalore@taxpro.co.in](mailto:taxprobangalore@taxpro.co.in)

## Ahmedabad Office

**Chartered Information Systems Pvt. Ltd.**  
#B-35, 2nd Floor, Ajanta Commercial Center,  
Income Tax Circle, Ashram Road,  
AHMEDABAD - 380014.  
**Phone:** 079-40083529, +919723715239  
**Email:** [taxproahmedabad@taxpro.co.in](mailto:taxproahmedabad@taxpro.co.in), [dsc.ahmedabad@charteredinfo.com](mailto:dsc.ahmedabad@charteredinfo.com)

# Welcome to TaxPro GST User Control Library

Chartered Information Systems Private Limited is a leader in India's Taxation Products domain and GST is no different. *TaxPro GST User Control Library* will enable developers of modern Desktop Accounting and Billing Applications to make their applications GST compliant and keep it up to date as per changes in GST Act and Rules. It includes user control for API Access as well as general usability and tooling:

- Settings, Authorization,
- all GSTR Data Upload and Return Controls,
- controls for Return Filing Status
- GST ledgers viz. ITC Ledger, Cash Ledger, and Tax Liability Ledger
- Other Tooling controls for Progress, Summary, SMS, etc

The Library also has ready classes for API access, Database Models and Methods for Data movement at different levels.

The usages scenarios for simplest cases to most complex cases has been kept in mind while designing *TaxPro GST User Control Library*.

## Platform

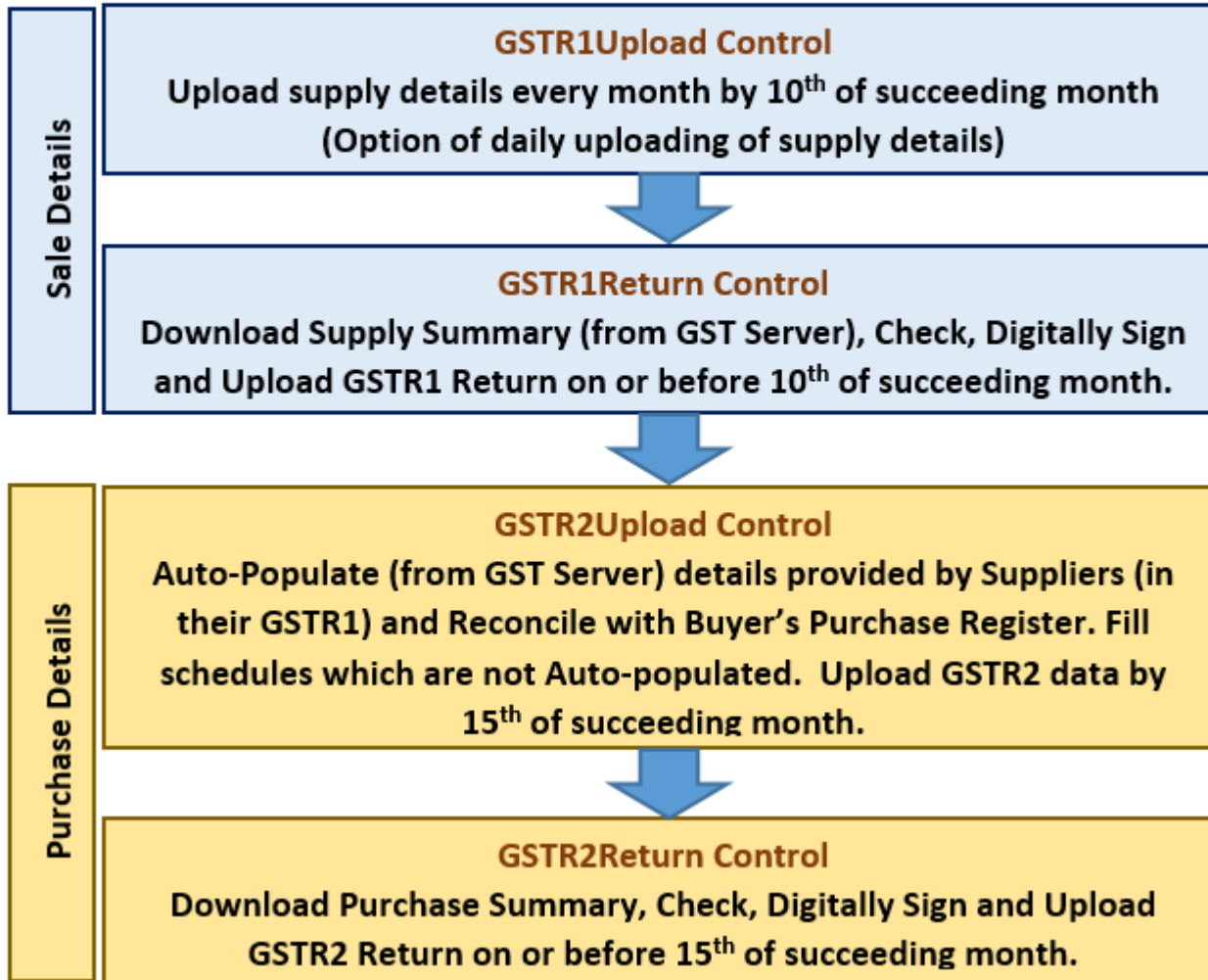
TaxProGST UCL requires .NET Framework 4.6 or latter.

## How to best read this user guide

- The best way to get started with GST implementation is to read topic [Where to Start?](#)
- Now that you are familiar with the basics of using the components, the next step would be to start integrating the component into your application. A good starting point is to refer to the code examples in the [Sample Projects](#).
- After integrated the component into application using one of the code examples as a starting point, it is likely that you may want additional information on specific features and API. The best option is to search the specific topic using the search box that is available at the top of the user guide.
- Another valuable resource is the API reference that provides detailed information on the object hierarchy as well as the settings available on every object.

## GST Return Filing Flowchart

(Assuming all returns are filed within Due Date)



Chartered Information Systems Private Limited © Chartered Information Systems Private Limited

Accept/Reject Changes

Aut  
Invo  
In

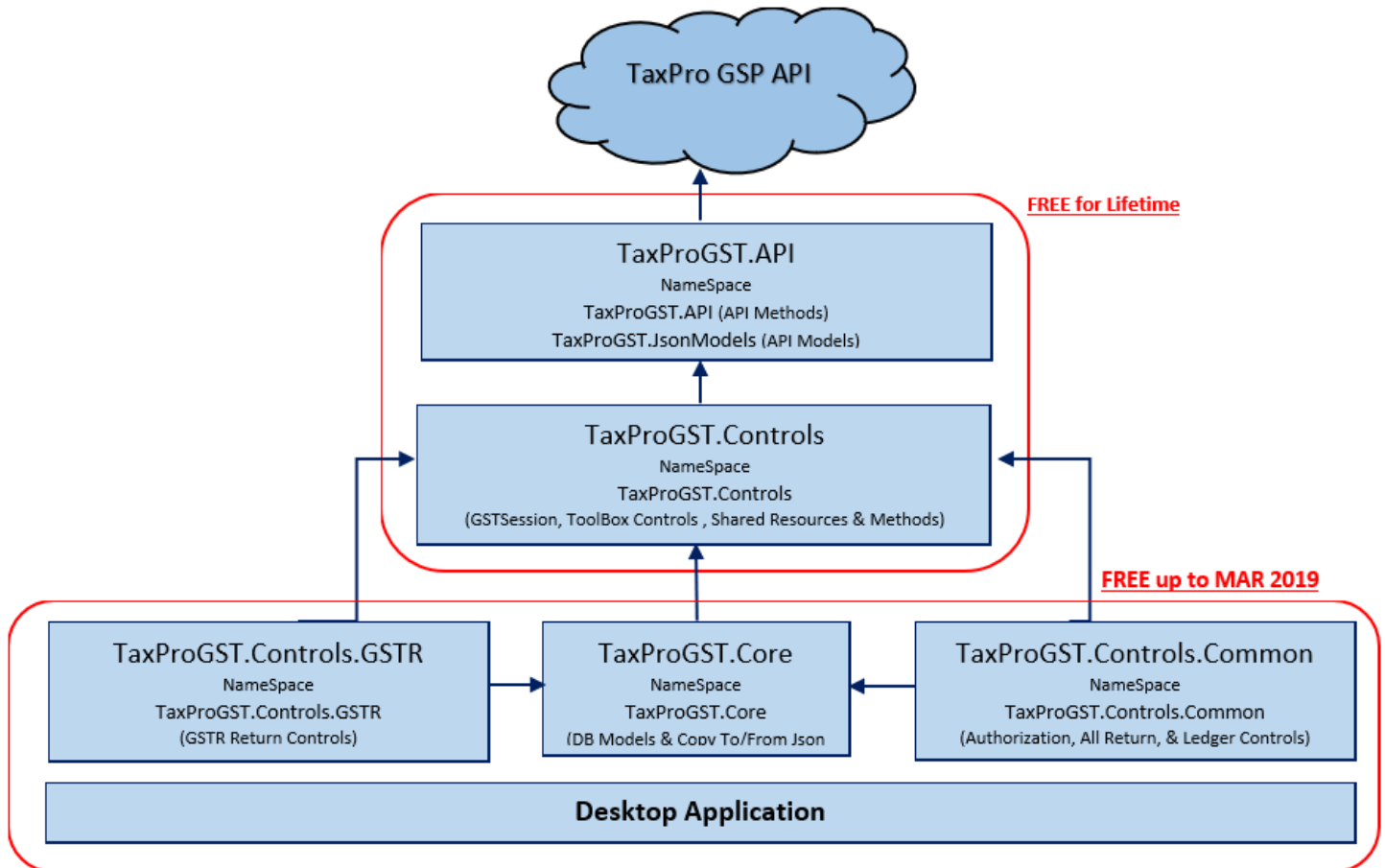
Do

F

## Assemblies and Namespaces

### FREE TaxPro GST API Windows .NET Library

#### TaxProGST Assembly Dependencies.



## Where to Start?

TaxPro GST User Control Library is developed to free developer from nitty-gritties of GST Invoice upload process and return filing. Software developer, using, TaxPro GST User Control Library should only supply connection string to GSTR controls. GSTR controls uses Microsoft Entity Framework to Save or Read data from database.

Thus good point to start developing using TaxPro GST User Control Library is Database Models present in assembly (and namespace) TaxProGST.Core This class library has classes like Gstr1, Gstr1Return, Gstr1A, etc. Please refer to [TaxProGST Core > DBModels](#) section for detailed list of such classes. These Models reflects database schema to support respective GSTR controls. These models helps you build your database schema along with primary keys and foreign keys and relations. You can use these classes (types) to build DbContext as shown below code sample. The complete DbContext classes required for GSTR forms are available as open source on GitHub as Sample Projects.

DbContext class sample for GSTR1 Return:

```
public class GSTR1Context : DbContext
{
    public DbSet<Gstr1Return> dbGstr1Return { get; set; }
    public DbSet<Gstr1Return.Gstr1Summary> dbGstr1RetSummary { get; set; }
    public DbSet<Gstr1Return.Gstr1Summary.SectionSummary> dbGstr1RetSecSummary { get; set; }
    public DbSet<Gstr1Return.Gstr1Summary.SectionSummary.CPSummary> dbGstr1RetCPSummary { get; set; }
    public DbSet<Gstr1Return.Gstr1Summary.SectionSummary.StateCodeSummary> dbGstr1RetSCSummary { get;
set; }

    public GSTR1Context() : base(nameOrConnectionString: "cnTPGST") {}
}
```

Once you implement all DbContext in your project, it remains to create your database tables. You may use the scripts available in DBUpdates folder or EntityFramework migrations to create database tables easily.

**Second thing to watch for is [GSTSession](#).** Using this class, TaxPro GST User Control Library provides out-of-box support for handling API Session for single GSTN by storing API Settings and API LoginDetails to Json configuration file in current directory from where application exe runs. If you need to implement support for multiple GSTIN (multiple TaxPayers) API Sessions, you need to write wrapper class which is discussed in topic [ToolBox Controls > GSTSession](#). GSTSession is derived from base class [APISession](#). Also sample wrapper class (with TaxPayerID dependency injection) is provided in GidHub Sample Projects.

Good Luck! Get going...!!!

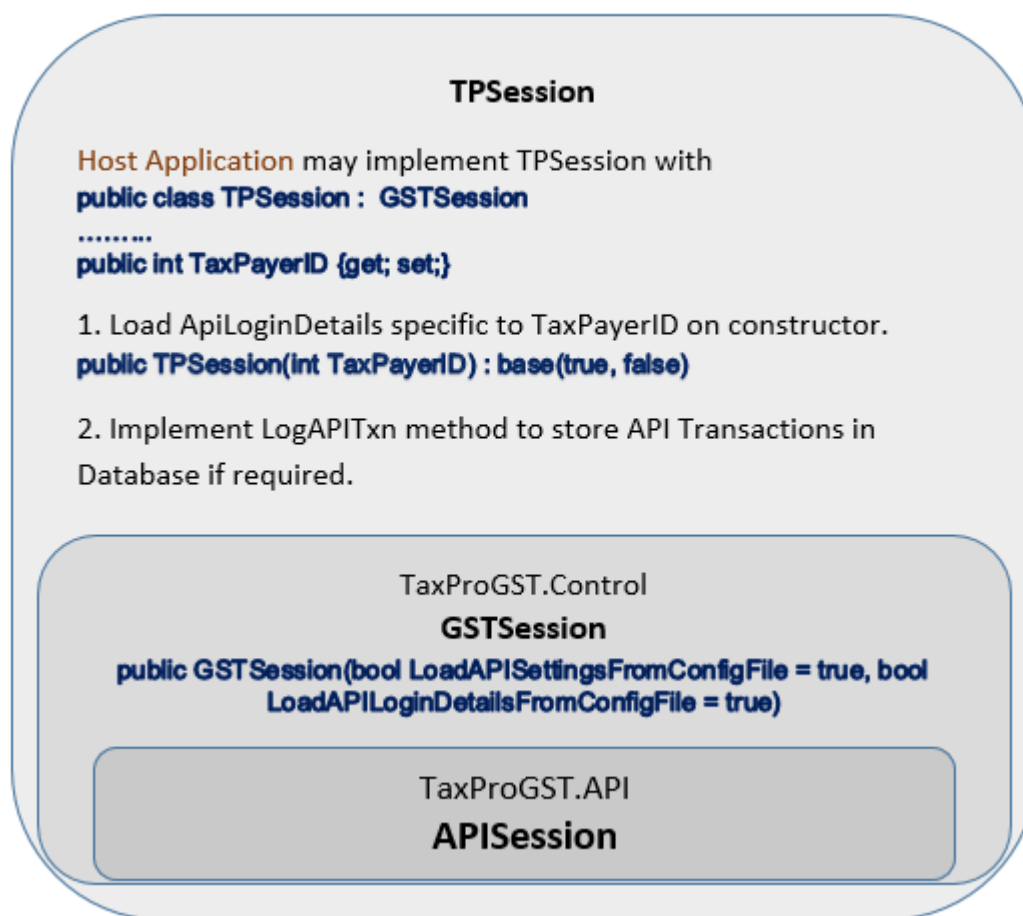


## Support Multiple TaxPayer

TaxPro GST User Control Library provides support for single TaxPayer ApiSetting and ApiLoginDetails store in default Json files, out-of-box. GstSession object would be internally created by reading these settings and there is no need to set GstSession property of API Enabled control.

If host application needs to support multiple TaxPayer, it needs to store ApiLoginDetails for multiple TaxPayer and needs to implement TPSession (creating different session object for each TaxPayer). Same logic may be implemented for all GSTR controls. Using TPSession name is not compulsory. Developer may use any convenient name and assign this object to GstSession property of all API Enabled controls (in host form/page constructor.)

### Understand APISession inheritance to support multiple taxpayers.



Sample C# Code:

```
using System;
using RestSharp;
using TaxProGST.API;
using TaxProGST.Controls;
using System.Reflection;

namespace TaxProGST
{
    public class TPSession : GSTSession
    {
        public int TaxPayerId { get; set; }
    }
}
```

```

public string TaxPayerName { get; set; } //If required

/// <summary>
/// Constructor for use in case of Non TaxPayer Specific API
/// </summary>
public TPSession() : base(true, false)
{
    if (ApiSetting.ASPName?.ToUpper() == "GSTN_SANDBOX")
        UserIP = "123.123.123.123";
    else
        UserIP = "DeskTopApp";
}

/// <summary>
/// Constructor for TaxPayerID dependencyt injection in GSTSession
/// </summary>
/// <param name="TPID"></param>
public TPSession(int TPID) : base(true, false)
{
    LoadLoginSettings(TPID);

    if (ApiSetting.ASPName?.ToUpper() == "GSTN_SANDBOX")
        UserIP = "123.123.123.123";
    else
        UserIP = "DeskTopApp";
    StartingApiTxn += CheckForLic;
    //RefreshSessionCompleted += StoreRefreshedSession; //
    Use for TaxPro GST GetKey Completion if you want to store ASP-GSP Session details
    RefreshAuthTokenCompleted += StoreRefreshedAuth;
}

private void LoadLoginSettings(int tpID)
{
    //Query various properties (as below) for ApiLoginDetails from database for Company tpID
    ApiLoginDetails = new APILoginDetails();
    ApiLoginDetails.AppKey = "AppKey for tpID";
    ApiLoginDetails.AuthToken = "AuthToken for tpID";
    ApiLoginDetails.GstPortalUserID = "GSTPortalUserName for tpID";
    ApiLoginDetails.SessionEK = "SessionEK for tpID";
    ApiLoginDetails.GSTIN = "GSTIN for tpID";
    //ApiLoginDetails.TokenExp = TokenExp.HasValue ? (DateTime)TokenExp : DateTime.Now;
    //TaxPayerName = CoName;
}

private void CheckForLic(object sender, EventArgs e)
{
    //if(!LicenseVersion)
    //{
    //    CancelApiMsg = "API Transactions not activated in Preview / Demo Version.";
    //    CancleApiTxn = true;
    //}
}

public override void AddApiHeaders(RestRequest request)
{
    request.AddHeader("appver", "TPDEMO" + Assembly.GetExecutingAssembly().GetName().Version.ToString());
}

```

```

        base.AddApiHeaders(request);
    }

    public override void AddGspApiHeaders(RestRequest request)
    {
        request.AddHeader("appver", "TPDEMO" + Assembly.GetExecutingAssembly().GetName().Version.ToString());
        base.AddGspApiHeaders(request);
    }

    private async void StoreRefreshedAuth(object sender, EventArgs e)
    {
        //Update Refresh AuthToken, SEK, TokenExp, AppKey to APILoginDetails for TaxPayer TPID
        //AuthToken = ApiLoginDetails.AuthToken;
        //SEK = ApiLoginDetails.SessionEK;
        //TokenExp = ApiLoginDetails.TokenExp;
        //AppKey = ApiLoginDetails.AppKey;
        //SaveChanges();
    }

    public async override void LogAPITxn(APITxnLogArgs e)
    {
        try
        {
            //using(MyDbContext db = new MyDbContext())
            //{
            //    APITxnItem al = db.dbApiTxnLog.Create();
            //    al.TPClientId = TaxPayerId;
            //    al.TPClientName = TaxPayerName;
            //    al.TxnID = TxnID;
            //    al.IsSuccess = e.IsSuccess;
            //    al.FilingPeriod = e.FilingPeriod;
            //    al.FormNo = e.FormNo;
            //    al.ApiAction = e.ApiAction;
            //    al.ApiTitle = e.ApiTitle;
            //    al.OutcomeMsg = e.OutcomeMsg;
            //    if(e.IsSuccess)
            //        al.Info1 = e.Info1;
            //    else
            //        al.ErrCode = e.ErrCode;
            //    al.Info2 = e.Info2;
            //    al.TxnDateTime = e.TxnDateTime;
            //    al.ASPName = ApiSetting.ASPName;
            //    al.AppUserName = strOperatorID;
            //    db.dbApiTxnLog.Add(al);
            //    await db.SaveChangesAsync();

            //    if(e.IsSuccess && e.ApiAction == "RETSUBMIT")
            //    {
            //        //StoreRetAckNo
            //    }
            //}

        }
        catch(Exception)
        {

        }
    }
}

```

}  
}  
}

## URL Based API

To make some of the GST Public APIs and GSP APIs more easy to consume for Developers, we have provided few of GET APIs through simple GET URL.

Below is the details for integrating **GSP APIs** and **Search GST TaxPayer API** with **TaxPro GSP** through URL based API. Other GST API are TaxPayer specific and requires end-to-end TaxPayer authorized and encrypted session to GSTN servers and hence its not possible to provide those API in this simple form.

### Steps:

1. [Register as ASP](#) with TaxPro GSP.
2. Invoke below URL (API) to GET the required data by passing ASPID, ASP Password along with GSTIN.

#### 1. Search GST TaxPayer Public API:

[https://api.taxprodsp.co.in/commonapi/v1.0/search?aspid=\*\*ASPID\*\*&password=\*\*ASPACPassword\*\*&Action=TP&Gstin=\*\*GSTINOfTaxPayerToSearch\*\*](https://api.taxprodsp.co.in/commonapi/v1.0/search?aspid=<b>ASPID</b>&password=<b>ASPACPassword</b>&Action=TP&Gstin=<b>GSTINOfTaxPayerToSearch</b>)

#### 2. Get API Balance:

[https://api.taxprodsp.co.in/aspapi/v1.0/getapibalance?aspid=\*\*ASPID\*\*&password=\*\*ASPACPassword\*\*](https://api.taxprodsp.co.in/aspapi/v1.0/getapibalance?aspid=<b>ASPID</b>&password=<b>ASPACPassword</b>)

#### 3. Get API Settings:

[https://api.taxprodsp.co.in/aspapi/v1.0/getapisetting?aspid=\*\*ASPID\*\*](https://api.taxprodsp.co.in/aspapi/v1.0/getapisetting?aspid=<b>ASPID</b>)

#### 4. Daily API Transaction Log:

[https://api.taxprodsp.co.in/aspapi/v1.0/dailytxnlog?aspid=\*\*ASPID\*\*&password=\*\*ASPACPassword\*\*&TxnDt=\*\*DateOfLogsToBeDownloaded\*\*](https://api.taxprodsp.co.in/aspapi/v1.0/dailytxnlog?aspid=<b>ASPID</b>&password=<b>ASPACPassword</b>&TxnDt=<b>DateOfLogsToBeDownloaded</b>)

Provide value of url query paramater in above urls - marked as **BOLD**.

### Parameters & Values

1. **aspid=ASPID** (Registered ASPID Received from TaxPro GSP)
2. **password=ASPACPassword** (given by you at the time of Registration)
3. **Gstin=GSTINOfTaxPayerToSearch** – GSTIN to be searched
4. **TxnDt=DateOfLogsToBeDownloaded** – format DDMMYY (01Jan17)

Response to the Search GST TaxPayer API Call will be as below

Search TaxPayer		
Details	Respective Key (Item)	Response Sample
Registration Date	rgdt:	01/07/2017
Company Type	ctb:	Private Limited Company/ Limited Company..
Constitution of Business	sts:	Active/ Inactive
Legal Name	lgnm:	ABC Corporation LTD.
State Jurisdiction	stj:	Jalandhar
GSTIN	gstin:	11AAAAA2222Q1Z2
Nature of Business Activity	nba:	Recipient of Goods or Services/Wholesale Bus



# TaxPro GSP API

TaxPro GSP Website	https://www.taxprogsp.co.in
TaxPro GSP API	Base Url: https://api.taxprogsp.co.in Samples same as below.
TaxPro GSP API Sandbox (for testing)	Base Url: http://testapi.taxprogsp.co.in Sample Urls GSP API: http://testapi.taxprogsp.co.in/aspapi/v1.0/getkey GST API: http://testapi.taxprogsp.co.in/taxpayerapi/v0.2/authenticate GST API: http://testapi.taxprogsp.co.in/taxpayerapi/v0.3/returns GST API: http://testapi.taxprogsp.co.in/taxpayerapi/v0.2/ledgers

## Steps to Access GST API:

GST Server is not directly accessible over Internet but would be accessible only through authorized GST Suidha Providers (GSP). Thus TaxPayer or Application Service Provider (ASP) who wishes to access GST API must create account with GSP and follow steps to gain access to GSP API.

1. [Generate ASP Certificate](#): Generate Private and Public Key for Encryption.
2. [Request Account](#) on GSP: Request ASP Account from TaxPro GSP. (Wait for Ac Approval.) (White label ASP Also available.)
3. [Download API Settings](#), and you are ready for GST API Access.

Above steps are trivial using TaxPro Products like TaxPro GST Desktop or Web Application, and TaxProGST.API library for .NET Framework 4.6

## FOR ADVANCED USAGE:

### API Access Scenarios using TaxPro UCL:

#### 1. Connect to GSTN SandBox directly:

- a. Use GSTN Sandbox Url e.g. http://devapi.gstsystem.co.in/taxpayerapi/vN.N/xxxxxx
- b. Put "GSTN\_SANDBOX" in ASP Name (in ApiSession.ApiSetting.ASPName)
- c. Put ASP 'ClientID' given for GSTN Sandbox access in ApiSession.ApiSetting.GSTNClientID and ClientSecret in ApiSession.ApiSetting.GSTNClientSecret
- d. This may be achieved by editing ApiSetting.json if you are using Default APISession arrangement provided by UCL. Or through ApiSettings Table of Database.

#### 2. Connect to TaxPro SandBox directly:

- a. Use TaxPro GSP API Sandbox Urls (as given in table above)
- b. Put "SANDBOX" word in ASP Name (in ApiSession.ApiSetting.ASPName)
- c. This would use TaxPro GSTN Sandbox Credentials.
- d. If you want to use your ASP 'ClientID' given for GSTN Sandbox access, put the same in ApiSession.ApiSetting.GSTNClientID and ClientSecret in ApiSession.ApiSetting.GSTNClientSecret
- e. This may be achieved by editing ApiSetting.json if you are using Default APISession arrangement provided by UCL. Or through ApiSettings Table of Database.

#### 3. Connect to Production using TaxPro ASPID:

- a. Use TaxPro GSP production Urls (as given in table above)
- b. TaxPro GSP Production Urls may be downloaded using API Call GETAPISETTING
- c. Keep ApiSession.ApiSetting.GSTNClientID and ApiSession.ApiSetting.GSTNClientSecret Empty - usages TaxPro Credentials on GSTN

**4. Connect to Production through TaxPro GSP using GSTN provided Credentials to GSTN registered ASP (thru TaxPro GSP):**

- a. Use TaxPro GSP production Urls (as given in table above)
- b. TaxPro GSP Production Urls may be downloaded using API Call GETAPISETTING
- c. Put ClientID in ApiSession.ApiSetting.GSTNClientID and ClientSecret in ApiSession.ApiSetting.GSTNClientSecret
- d. Alternatively Override AddHeaders method in APISession to add Production Credentials and then call base method. (Ref: [Sample C# Code](#) section)

-----



# Generate ASP Certificate

## Steps to be followed by ASP:

1. Download "Openssl" from internet (search on Google) and Extract in one folder.
2. Use Windows Command Prompt to execute below steps in openssl folder.
3. Execute below command to generate private key file **AspDsc.key** and Public Key file **AspDsc.cer**  
**openssl req -x509 -nodes -sha256 -days 3650 -newkey rsa:2048 -keyout AspDsc.key -out AspDsc.cer -config openssl.cnf**  
This setp will ask for Inputs like Country (IN), State, Co. Name, e-Mail etc.  
**Make sure you put "GST" in OU (Orginational Unit)**
4. Execute below command to create .pfx file.  
**openssl pkcs12 -export -in AspDsc.cer -inkey AspDsc.key -CSP "Microsoft Enhanced RSA and AES Cryptographic Provider" -out AspDsc.pfx -password pass:taxpro1234**
4. Update ASP Profile. Share Public Key .cer file with GSP for verification of Encrypted ASPSecretKey in your API Calls.
5. Copy the .pfx file generated in above step to folder where TaxProGST.API.dll is installed (Application Folder)
6. Asp may need to access API from multiple computers on LAN using Desktop application, hence support for .pfx DSC is given. Liability of protecting Private Key lies solely on ASP.

## GetApiSetting API

Assembly: TaxProGST.API

GetApiSetting may be **Optionally used** by ASP to get present API settings for ASP Application

**HTTP Method: GET**

**Headers:**

Property	Description	Data Type
aspid	AspUserId provided by GSP.	String

**Response: GetApiSetting API response returns below properties from type APISetting**

Property	Description	Data Type
ASPName	For User Friendly name only. Not referred or used by GSP System.	String
IsActive	ASP may use this in case multiple settings are stored at ASP end.	String
IsDefault	ASP may use this in case multiple settings are stored at ASP end.	Integer
AspWebsite	ASP Website address or GSP's ASP Portal address. Not used by API system.	String
UrlAuth	GST Authentication API access end point @ GSP Server. e.g. http://api.taxprovsp.co.in/taxpayerapi/v0.2/authenticate	String
AuthVer	Active version of Authentication API on GSP Server.	String
UrlReturn	GST Returns API access end point @ GSP Server. e.g. http://api.taxprovsp.co.in/taxpayerapi/v0.2/returns	String
ReturnVer	Active version of Returns API on GSP Server.	String
UrlLedger	GST Authentication API access end point @ GSP Server. http://api.taxprovsp.co.in/taxpayerapi/v0.2/ledgers	String
LedgerVer	Active version of Ledgers API on GSP Server.	String
UrlAspUtil	GSP - ASP APIs Url http://api.taxprovsp.co.in/aspapi/v0.2	String

**GetApiSetting API Call Method (applies only if you are using TaxProGST.API.GspAPI Class)**

API (Method)	Method in TaxPro Lib (GspAPI class)	Parameter Type	Description
GetApiSetting (POST)	GetApiSetting	<ul style="list-style-type: none"> <li><a href="#">APISession</a></li> </ul>	This method is used to download APISetting from TaxPro GSP Server.

## GetKey API

In very simple words; to prevent ASP from transmitting AspSecretKey (AspPassword) openly on internet, there is need to establish a same encryption key at ASP & GSP end. ASP will then use the key (AspEK) to encrypt AspSecretKey and GSP will use the same key to decrypt AspSecretKey before authenticating ASP. Thus GetKey API will establish AspEK encryption key, identified by AspSessionID, periodically. Encryption used is AES/ECB/PKCS7.

### Steps to be followed by ASP Application to establish and maintain ASP - GSP Session:

1. Call **GetKey API** to establish ASP - GSP Authentication
2. Call GST APIs  
(Check validity of ASP - GSP session before every call or alternatively have timer to refresh ASP - GSP session before it expires.)
3. Use **GetKey API** to Refresh ASP - GSP session when it is Expired.

**GetKey API for GSP / ASP Authentication** assumes that ASP has registered with GSP and completed below steps:

1. Enrolled with GSP to **obtain credentials AspID** (ASPUserName or ASPUserID) and **AspPassword** (AspSecretKey).
2. **Self Generated DSC**; PrivateKey and PublicKey pair for signing GetKey API request **and registered PublicKey with GSP.**

### Steps for ASP Software Integrators:

**Note: TaxProGST Desktop Application and TaxProGST.API library has inbuilt functionality for GetKey API and ASP - GSP Session is automatically checked for Expiry (and refreshed automatically) before call to any GST API. Details of implementation is given at the end of this page.**

HTTP Method: POST

## Headers

GetKey API Call should have below Headers

Property	Description	Data Type
aspid	AspUserId provided by GSP.	String
txn	Unique number which will be sent back in response as it is for ASP to map it back to original request.	String

## Body

GetKey API request Body is of Type ReqGSPSession

Property	Description	Data Type
timestamp	.NET DateTime Format "ddMMyyyyHHmmssFFFFFF"	String
signed_content	RSA SHA256 HASH of Concatenated string <"aspid" + "timestamp"> should be signed using ASP PrivateKey (.pfx file as discussed at top of this page). Signing Algorithm: RSA SHA256 <a href="#">Sample C# Code to DSC Sign</a> in given in this guide. <b>GspAPI.RefreshGSPSessionAsync</b> method in lib TaxProGST.API does this transparently.	String

## Response

### GetKey API response is of type RespGSPSession

Property	Description	Data Type
enc_key	The enc_key in GetKey response will be encrypted by GSP using ASPPassword (Password issued to ASP by GSP at the time of ASP registration). Decrypt enc_key and store in ApiSetting.AspKey.  Encrypt ASPPassword using ApiSetting.AspKey in subsequent GSTN API Calls. The Encrypted ASPPassword APISession header. Algorithm: AES/ECB/PKCS7 <b>APISession.StartApiTxn</b> method in lib TaxProGST.API does this transparently.	String
session_id	Session ID. This will be passed in the subsequent calls made by ASP. GSP System will decrypt aspid in subsequent calls to GST API, using enc_key for the session-id in header of subsequent calls.	String
validity_min	Validity of ASP- GSP Session in minutes. Session should be renewed after this period by call to GetKey API. This will provide new SessionID and AspEK in session_id and enc_key response variable respectively.	Integer
txn	Unique number which will be sent back in response as it is for ASP to map it back to original request.	String
message	Error Message if any.	String
status_cd	status_cd:"0" ( "0" is for failure) status_cd:"1" ( "1" is for success )	String

### GetKey API Call Method (applies only if you are using TaxProGST.API.GspAPI Class)

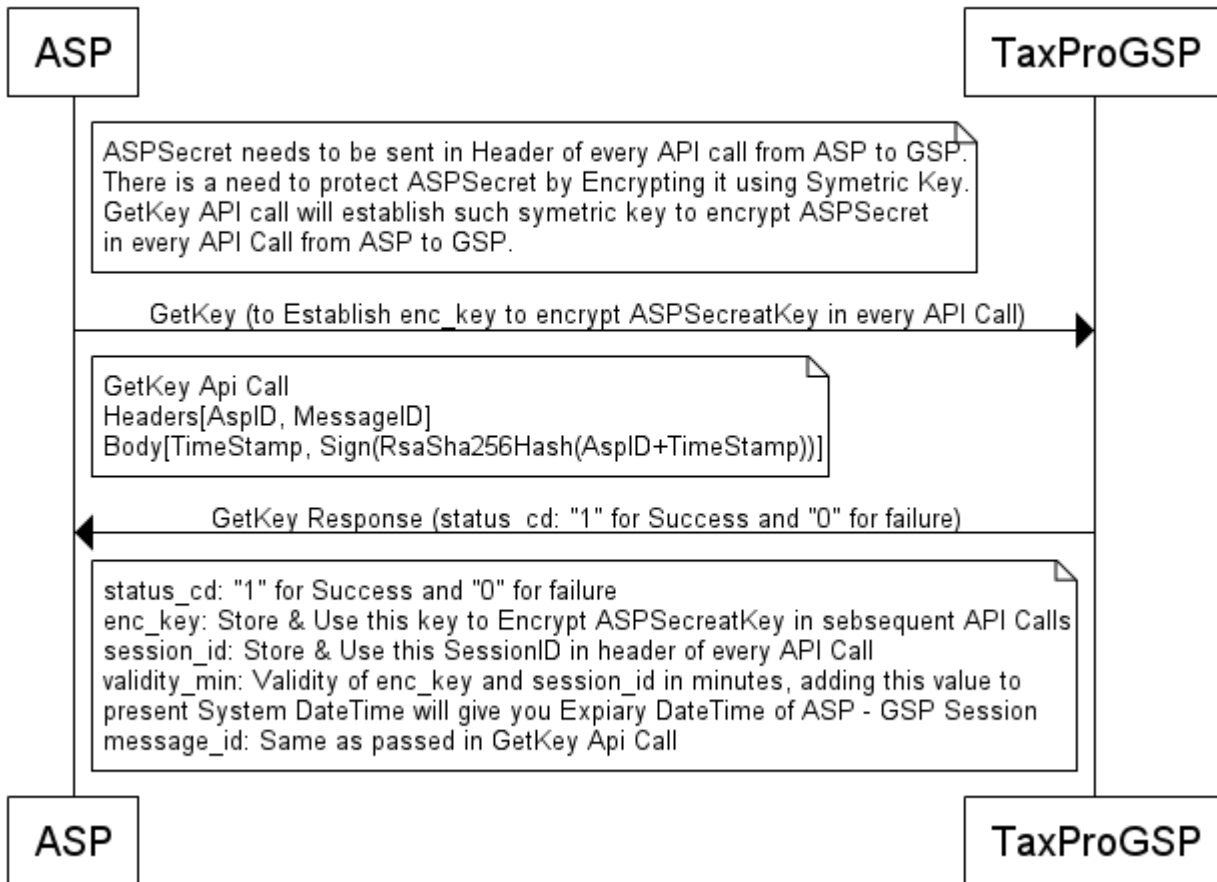
API (Method)	Method in TaxPro Lib (GspAPI class)	Parameter Type	Description
GetKey (POST)	RefreshGSPSessionAsync	• <a href="#">APISession</a>	This method is used to Get & Refresh GSP session asynchronously. This will also establish GSP Session Encryption Key: ApiSetting.AspEK used to encrypt GSP Security Key stored in: ApiSetting.AspPassword  This method is automatically called from APISession at the beginning of StartAPITxn method, in case GSPSession is found to be expired and explicit call is not required.

### Events

	RefreshSessionCompleted	updates instance of APISession on which StartAPITxn method is invoked	Use this event to store updated AspSessionID, AspKey, AspSessionExp in database.
--	-------------------------	---	--

# GetKey API Sequence Diagram

## GetKey API



Chartered Information Systems Private Limited

## C# Sample to DSC Sign AspID + TimeStamp

The below method and other Cryptographic methods are available in FREE library **TaxProGST.API** in **TPCrypto** class

```
//Sign String with Digital Signature
// PfxFile (full .pfx file path) may also be replaced with byte[] of pfxfile
/// <returns>Digital Signature Hash</returns>
public static string DSCSignForSession(string ContentToSign, string PfxFile, string DSCPassword)
{
    try
    {
        X509Store store = new X509Store(StoreLocation.CurrentUser);
        store.Open(new OpenFlags());
        X509Certificate2 AspCert = null;
        X509Certificate2Collection certificates = new X509Certificate2Collection();
        if (File.Exists(PfxFile))
            certificates.Import(PfxFile, DSCPassword, X509KeyStorageFlags.PersistKeySet);
        if (certificates.Count == 0)
            return "Certificate not found";

        foreach (X509Certificate2 cert in certificates)
        {
            foreach (string cerProp in cert.Subject.Split(','))
            {
                if(cerProp.Trim().StartsWith("OU") && cerProp.Split('=')
[1].ToUpper().Contains("GST"))
                {
                    AspCert = cert;
                    break;
                }
            }
        }
        if (AspCert != null)
        {
            RSACryptoServiceProvider rsaEncryptor = (RSACryptoServiceProvider)AspCert.PrivateKey;
            byte[] CalcHash = (new SHA256Managed()).ComputeHash(Encoding.UTF8.GetBytes(ContentToSign));
            return (Convert.ToBase64String(rsaEncryptor.SignHash(CalcHash, CryptoConfig.MapNameToOID("SHA256"))));
        }
        else
            return "Certificate not found";
    }
    catch(Exception ex)
    {
        return ex.Message;
    }
}
```

# GST Pass Through API

These details are given here for Documentation. Headers are added to all API calls using API methods if you are using TaxProGST.API library.

## GSP Headers in Every API Calls

Header	Description
aspid	AspUserName or AspUserID provided by GSP.
asp-secret	AspPassword or AspSecretKey provided by GSP for pass through API access, Encrypted using SessionEK (enc_key)
session-id	SessionID. ASP may create multiple Sessions.
txn	Unique Transaction ID even if calling GSP Service API. (Requires only once, which you are already passing as requirement for GST API.)
appver	Application Name and Version calling API. (Optional but Recommended)

## GST Headers in Every API Calls (in addition to GSP Headers above)

Header	Description
username	GSTN Portal UserName (UserID) of TaxPayer.
gstn	GSTIN of TaxPayer.
state-cd	State code of TaxPayer. (First 2 digits of GSTIN).
auth-token	TaxPayer Authorization token obtained from GST Server.
ret_prd	Return Period (Required for Return API Calls)
ip-usr	IP of device from which API is being invoked.
txn	Unique Transaction ID.
Content-Type	application/json

It is recommended to add Header 'appver' to every request which would help you debug API calls if required, with our help.

Override method AddApiHeaders in APISession to add required headers to every request.

```

public override void AddApiHeaders(RestRequest request)
{
    request.AddHeader("appver", "MyApp" +
Assembly.GetExecutingAssembly().GetName().Version.ToString());
    base.AddApiHeaders(request);
}

public override void AddGspApiHeaders(RestRequest request)
{
    request.AddHeader("appver", "MyApp" +
Assembly.GetExecutingAssembly().GetName().Version.ToString());
    base.AddGspApiHeaders(request);
}

```

## GSP Error Code

**Error Response:** Same model as GST error response (as below present in lib TaxProGST.API)

```
class RespErrPI
{
    public string status_cd { get; set; }    //"1" for Success, "0" for Error
    public Error error { get; set; }
    public class Error
    {
        public string error_cd { get; set; }
        public string message { get; set; }
    }
}
```

### TaxPro GSP Error Codes:

Error Code	Description	Error Domain / Area
GSP001	Missing Parameter 'aspid' in Request Header.	Request Validation
GSP001A	Invalid Length of 'aspid' in Request Header.	Request Validation
GSP002	Missing Parameter 'asp-secret' in Request Header.	Request Validation
GSP003	Missing Parameter 'session-id' in Request Header.	Request Validation
GSP004	Missing Parameter 'username' in Request Header.	Request Validation
GSP005	Missing Parameter 'GSTIN' in Request Header.	Request Validation
GSP005A	Invalid Parameter 'GSTIN' in Request Header.	Request Validation
GSP006	Missing Parameter 'txn' in Request Header.	Request Validation
GSP007	Missing Parameter 'ip-usr' in Request Header.	Request Validation
GSP008	Missing Parameter 'auth-token' in Request Header.	Request Validation
GSP009	Missing Parameter 'state-cd' in Request Header.	Request Validation
GSP009A	Invalid Len Parameter 'state-cd' in Request Header.	Request Validation
GSP009B	Invalid value in 'state-cd' in Request Header.	Request Validation
GSP011	Invalid Parameter 'aspid' or 'message-id'.	GetKey API
GSP012	Error: AspID not registered. Check or Register AspID.	GetKey API
GSP013	Error: Asp DSC not Registered. Please Register ASP DSC.	GetKey API
GSP014	Invalid Signature. Asp DSC Could not be Verified.	GetKey API
GSP014A	Invalid ASP Secret Key, please verify.	GetKey API
GSP016	Error: No Value Set for GSP Mandatory Paramater.	GSP Authorization
GSP017	Error Decrypting AspSecreatKey.	GSP Authorization
GSP018	Invalid ASPSecKey (Password).	GSP Authorization
GSP019	Insufficient ASP Credit. Please contact GSP.	GSP Authorization
GSP020	Invalid AspID no matching ASP found.	GSP Authorization
GSP020A	Error: Invalid ASP Password. (Used for ASPAPI using AspID & Pwd)	GSP Authorization



GSP021	IP Not White listed with GSP.	GSP Authorization
GSP022	Error: Session Expired. Call GetKey before API Request.	GSP Authorization
GSP023	Error: Invalid SessionID.	GSP Authorization
GSP024	Error: ASP Account Not Active. Contact Provider	GSP Authorization
GSP024A	Error: API Call allowed only for ASP Account Type.	
GSP025	Exceeded requests per Second (TPS). Retry with some Delay.	GSP Request Handler
GSP026	Failed to Create Response.	GSP Request Handler
GSP027	Network Error.	Communication Failure
GSP028	Transient Error, Failed to get ASP Settings.	GetApiSetting API
GSP030	"Error Establishing Connectivity with GST Server..."	--
GSP031	'Internal Server Error' response from principal provider.	--
GSP032	Exception Message	--
GSP033	Email Fwd OTP Not Received or Not Found.	Auto OTP Authorization
GSP033E	Exception in DB Operation while Email Fwd OTP Fetch.	Auto OTP Authorizationcc

# GSP Utility API

If using UCL or FREE TaxPro API Library, Reference: Assembly: TaxProGST.API

## 1. GetApiBalance

HTTP Method: GET

Headers:

Property	Description	Data Type
aspid	AspUserId provided by GSP.	String
password	ASP Account password used for ASP Portal login.	String

**Response in below Type:**

```
public class AspApiBalance
{
    public int ApiBal { get; set; }
    public DateTime ApiBalExpDt { get; set; }
}
```

## 2. SetASPPubCer

Used to update newly generated Public Key (".cer" file) to ASP Account.

HTTP Method: POST

Headers:

Property	Description	Data Type
aspid	AspUserId provided by GSP.	String
password	ASP Account password used for ASP Portal login.	String

**Body: Content of Cer File (By replacing -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----)**

**Success Response:** HttpStatusCode.OK (200)

## **GST Public API**

GSTN had made available below APIs as Public API which can be accessed by public without requiring TaxPayer's Authentication; but only through GSP.

# Search GST TaxPayer API

**API Method:** GET

**URL:** <https://api.taxprogs.co.in/commonapi/v1.0/search>

## Headers

Property	Description	Data Type
aspid	AspUserId provided by GSP.	String
password	Password of ASP Account.	String

Note: You may provide above parameters as Query Parameters also.

## Query Parameters

Property	Description	Data Type
action	"TP" - set to value TP for Search TaxPayer Action.	String
gstn	GSTIN of TaxPayer to Search for.	String

**Response to the Search GST TaxPayer API Call will be as below**

Search TaxPayer		
Details	Respective Key (Item)	Response Sample
Registration Date	rgdt:	01/07/2017
Company Type	ctb:	Private Limited Company/ Limited Company..
Constitution of Business	sts:	Active/ Inactive
Legal Name	lgnm:	ABC Corporation LTD.
State Jurisdiction	stj:	Jalandhar
GSTIN	gstn:	11AAAAA2222Q1Z2
Nature of Business Activity	nba:	Recipient of Goods or Services/Wholesale Bus

## Method in TaxProGST.API Lib:

```
public async static Task<TxnRespWithObj<SearchTaxpayerJson>> SearchTaxPayerAsync(APISession ApiSession)
```

**Note:** If you are using "SearchTaxPayer" User Control and setting "DatabaseContext" Property, it will catch TaxPayer's Info in table "GSTINInfoCache".

## Sample for URL Based API:

<https://api.taxprogs.co.in/commonapi/v1.0/search?>

[aspid=ASPID&password=ASPACPassword&Action=TP&Gstin=GSTINOfTaxPayerToSearch](https://api.taxprogs.co.in/commonapi/v1.0/search?aspid=ASPID&password=ASPACPassword&Action=TP&Gstin=GSTINOfTaxPayerToSearch)

# GST API

**Assembly:** TaxProGST.API

GST API and Json Models required to communicate with GST server are implemented in assembly TaxProGST.API which has two namespaces:

- 1. Namespace: TaxProGST.API** - APISession and API Methods
- 2. Namespace: TaxProGST.JsonModels** - Models for accessing GST API

Developer desired to use API Methods directly bypassing User Controls may use classes and methods in this library.

**Dependency:**

**.Net Framework 4.6 Assemblies**

System.ComponentModel.DataAnnotations

System.Security

**Third Party Libraries**

Newtonsoft.Json

RestSharp

## APISession

**Assembly and Namespace:** TaxProGST.API

**Dependency:** RestSharp

APISession is very important class in TaxPro GST User Control Library and developer needs to have excellent understanding of APISession for using TaxPro GST User Control Library to develop multi TaxPayer application.

### APISession Properties

Property	Description	Data Type
ApiSetting	Stores all variables required to access API endpoint along with. ApiSetting would be fixed even for accessing API for Multiple TaxPayers and would not change depending on TaPayer. ApiSetting members are listed in table below.	TaxProGST.API.APISettin
ApiLoginDetails	Store all TaxPayer specific values viz. GSTIN, StateCode, UserID, Encryption Keys, Session Validity, etc Members are listed in table below.	TaxProGST.API. APILoginDetails
ReturnPeriod	ReturnPeriod in "MMyyyy" format. Required for all API access except for Authorization API.	String
TxnID	Unique Transaction ID. Internally generated on StartAPITxn call by API enabled control.	String
UserIP	Required to set if in case of hosted application accessed by number of users on internet. If not set, will be set to IP of ASP server or Desktop Application from where API is being accessed.	String
CancleApiTxn	Set to True if API transaction to be canceled in StartPITxn. Would be useful if user want to disable API access from single point (depending of active License, etc).	bool
CancelApiMsg	Set the message in this property if API transaction disabled by setting CancleApiTxn to True.	String

### ApiSetting Members (Mandatory Fields marked with \*)

Property	Description	Data Type
ID	Unique ID - Primary Key in case DbContext needs to be created on ApiSetting/	Int
ASPName	Friendly Name for ApiSetting	String
AspPassword	AspPassword or Security Key	
IsActive	Indicates if API indicated by this setting is Active.	bool
IsDefault	Indicates if API is default API in case multiple API settings created.	bool
AspWebsite	Used to download APISetting using <a href="#">EditAPISetting</a> control.	String
AspUserId	User ID on ASP/GSP	String
AspEK	Encryption Key used by ASP - GSP communication.	Stirng

AspDscPassword	DSC password in case ASP uses signed request to establish session.	String
UrlAuth*	Authorization API Endpoint Url (full url with version number)	String
UrlReturn*	Returns API Endpoint Url (full url with version number)	String
UrlLedger*	Ledger API Endpoint Url (full url with version number)	String
UriASPUtil	GSP Authorization and Services Endpoint URL.	String
AspSessionID	ASP - GSP Session identifier.	String
AspSessionValidityMins	Validity of ASP / GSP Session in minutes. AspSessionExp would be calculated by adding these many Minutes to current ASP / GSP refresh session call success time.	Int
AspSessionExp	API engine will refresh ASP / GSP session after this time at the beginning of StartApiTxn call.	DateTime

### ApiLoginDetails Members (All fields are mandatory)

Property	Description	Data Type
GSTIN	GSTIN of TaxPayer	String
GstPortalUserID	UserID of TaxPayer on GST Portal. Use same User Id that is used at the time of GST registration.	String
AppKey	AppKey used for obtaining OTP as a part of <a href="#">Authorization Token generation process</a> .	String
AuthToken	Authorization Token obtained at the end of successful Authorization API Call.	String
TokenExp	DateTime the AuthToken expires.	DateTime
SessionEK	Session Encryption Key (EK) obtained at the end of successful Authorization API Call or Extend Session API Call.	String

Refer to TPSession class in Sample Project which demonstrates how to write wrapper with TaxPayer dependency injection in APISession to provide TaxPayer specific values to ApiLoginDetails members. .

### Events

Event	Description
StartingApiTxn	The event occurs as first step of StartApiTxn method call. May be used to set CancelApiTxn property to false to cancel API Transaction.

### Methods

Method	Description
StartApiTxn Paramaters: RestRequest	StartApiTxn method performs below tasks: <ol style="list-style-type: none"> <li>1. Invoke StartingApiTxn Event</li> <li>2. Generate unique TxnID for every call</li> <li>3. Check for required ApiSetting and ApiLoginDetails paramaters.</li> <li>4. Cancel the transaction if CancelApiTxn property set to false</li> </ol>

	5. Check and refresh ASP-GSP session. (by calling GetKey API).
<p>LogAPITxn  Paramaters: TaxProGST.API.  APITxnLogArgs</p>	<p>Virtual method with no base implementation. <a href="#">TaxProGST.Controls.GSTSession</a> class derived from APISession impliments logging of API Transactions to file in program folder.  Override this method in wrapper class to impliment API Transaction logging. (Refer to TPSession class in Sample Project.</p>



# API Methods

Assembly and Namespace: TaxProGST.API

API Methods for **Authentication Actions** are listed in below table:  
uses **GST/GSP/ASP** API end point stored in **ApiSetting.UrlAuth**

## 1. AuthAPI Class

GST API Action (In Body)	Methods	Parameter Type	Description
OTPREQUEST	GetOTPAsync	• APISession	This method is used to Get OTP asynchronously.
AUTHTOKEN	GetAuthTokenAsync	• APISession • OTP	This method is used to Get Authorization Token asynchronously.
REFRESHTOKEN	RefreshAuthTokenAsync	• APISession • PfxFile	This method is used to Refresh Authorization Token asynchronously.

## GSTR API Methods

**Assembly and Namespace:** TaxProGST.API

API Methods for different GSTR actions are listed in below table:

API Models keeps on changing & evolving. Developers are requested to use Object Browser in Visual Studio to view list of methods and their signatures in above namespace. GSTR methods are under respective GSTRnAPI class.

## Json Models

**Assembly:** TaxProGST.API **Namespace:** TaxProGST.JsonModels

Json Models are used to hold data before serialization while generating API payload and deserializing API response payload.

Json Models keep on changing & evolving. Developers are requested to use Object Browser in Visual Studio to view list of models available in above namespace.

# List of Controls in TaxPro GST User Control Library

Category	Assembly	Controls
TooBox Controls	TaxProGST.Controls	<a href="#">ProgressPanel</a>
		<a href="#">SummaryPanel</a>
		<a href="#">SendSMS</a>
		<a href="#">Form3BOffsetLiability</a>
Authorization, Return Tracking, Ledgers	TaxProGST.Controls.Common	<a href="#">EditAPISetting</a>
		<a href="#">APIAuthorization</a>
		<a href="#">ReturnStatus</a>
		<a href="#">AdvTaxPaidDetails</a>
		<a href="#">PartialITCDetails</a>
		<a href="#">TSDetails</a>
		<a href="#">CashLedger</a>
		<a href="#">ITCLedger</a>
		<a href="#">LiabilityLedger</a>
		<a href="#">UtilizeCash</a>
		<a href="#">UtilizeITC</a>
		<a href="#">GSPDailyAPITxn</a>
		GST Returns
<a href="#">GSTR1Return</a>		
<a href="#">GSTR2Return</a>		
<a href="#">GSTR2Upload</a>		
<a href="#">GSTR1AUpload</a>		
<a href="#">GSTR1AReturn</a>		
<a href="#">GSTR3BReturn</a>		

# ToolBox Controls

Assembly and Namespace: TaxProGST.Controls

ToolBox Controls are controls used by TaxPro GST User Control Library itself as a part of building state of art user experience. These controls are also available as User Controls so that developer would be able to use these controls in their projects. Also Data Model for complex controls are exposed for efficient DataBinding in WPF projects.

ToolBox Controls has below controls:

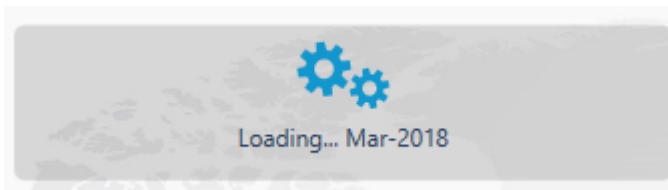
- ProgressPanel
- SummaryPanel
- SendSMS

# ProgressPanel

Assembly and Namespace: TaxProGST.Controls

ProgressPanel control has two views.

## InProgress View.



Set properties (Before starting Long running Task)

**InProgress** = True

**ProgressText** = "Fetching Data..."

## Last Action View



Set properties (At the end of Long running Task)


**InProgress** = False

**LastAction** = "Get Data Success." time will be prefixed internally.

## ProgressPanel Properties

Property	Description	Data Type
InProgress	Set true to activate progress animation and show <i>ProgressText</i>	bool
ProgressText	Display text below progress animation	String
LastAction	Text to display in Last Action View (when InProgress is false). Date time would be prefixed to LastAction text as shown in above image. <b>LastAction text automatically displayed in red color if LastAction text contains words "FAILED", "CANCEL", "NOT A VALID".</b>	String

## Events

Event	Description
ActivityLogClickEvent	 When not InProgress, "Activity Log" button will be displayed along with LastAction. This is RoutedEvent with RoutingStrategy Tunnel and will be raised on clicking this button. This Tunneling event will propagate from child to parent. Handle this event on every Page/Window/Form or alternatively handle this event only in Parent (eg: MDI Container) to open Activity Log Window.

## Event Handler Code C# Sample:

```
//Add Global Root level handler for Tunneled Routed Event ActivityLogClick
DockingManager.AddHandler(TaxProGST.Controls.ProgressControl.ActivityLogClickEvent,
    new RoutedEventHandler(ShowActivityLog));
```

```
//Code to show/activate ActivityLog Window
public void ShowActivityLog(object sender, RoutedEventArgs e)
```

```
{  
    //Code to show/activate ActivityLog Window  
}
```

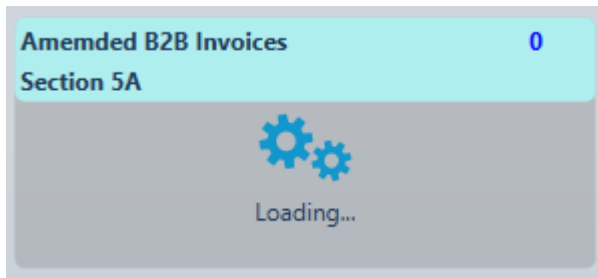
**Note:** Class **ProgressControlModel** with ProgressPanel properties grouped together and INotifyPropertyChanged interface implemented for each property available in the same Namespace for DataBinding.

# SummaryPanel

Assembly and Namespace: TaxProGST.Controls

SummaryPanel control has two views.

## InProgress View.

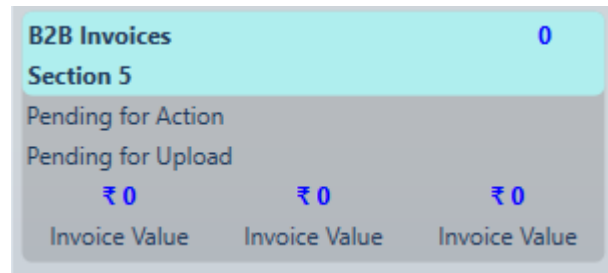


Set properties (Before starting Summary Loading)

**InProgress** = True

**ProgressText** = "Fetching Data..."

## Summary View



Set properties (At the end of Summary Loading)

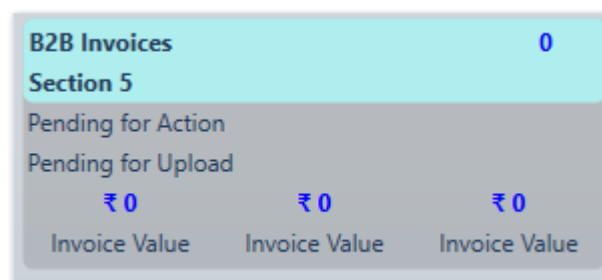
**InProgress** = False

**LastAction** = "Get Data Success." time will be prefixed internally.

## ProgressPanel Properties

Property	Description	Data Type
InProgress	Set true to activate progress animation and show <i>ProgressText</i>	bool
ProgressText	Display text below progress animation	String
SummaryData	SummaryPanel properties grouped together and INotifyPropertyChanged interface implemented for each property available in the same Namespce for DataBinding.	TaxProGST.Core. SummaryPanelModel

Other position specific properties to set different sections of SummaryPanel is listed in below diagram:





# SendSMS

Assembly and Namespace: TaxProGST.Controls

Not RELEASED - Under Development

# GSTSession

**Assembly and Namespace:** TaxProGST.Controls

GSTSession is a wrapper for class APISession which implements loading of ApiSetting and ApiLoginDetails from Json file. If host application wants to use API enabled controls only for single GSTIN then GSTSession may be used out-of-box.

**APISession** is base class of GSTSession; i.e. GSTSession class is derived from **APISession** class and has all properties and methods of APISession class apart from below 2 more properties.

GSTSession constructor takes two parameters

1. LoadAPISettingsFromConfigFile - Default value True
2. LoadAPILoginDetailsFromConfigFile - Default value True

Property	Description	Data Type
LoadAPISettingsFromConfigFile	Flag indicating whether ApiSetting is being loaded from Json config file "ApiSetting.json" in application folder.	bool
LoadAPILoginDetailsFromConfigFile	Flag indicating whether ApiLoginDetails is being loaded from Json config file "ApiLoginDetails.json" in application folder.	bool

## NOTE:

- On successful GetOTP API call, APIAuthorization control examines LoadAPILoginDetailsFromConfigFile flag and if found True, updates ApiLoginDetails.AppKey value in "ApiSetting.json" config file.
- On successful GetAuthToken API call, APIAuthorization control examines LoadAPILoginDetailsFromConfigFile flag and if found True, updates ApiLoginDetails.AuthToken, ApiLoginDetails.TokenExp, and ApiLoginDetails.SessionEK values in "ApiSetting.json" config file.
- If LoadAPILoginDetailsFromConfigFile flag is false, wrapper class implementing and managing session has responsibility to handle events GetOTPCompleted, and GetAuthTokenCompleted and update above values in TaxPayers stored API session data.

## Methods

Method	Description
LogAPITxn Parm: TaxProGST.API. APITxnLogArgs	Implements logging of API Transactions to file "GstApiTxnLog.txt" in program folder

## Form3BOffsetLiability

**Assembly and Namespace:** TaxProGST.Controls

## GSTR Controls

**Assembly and Namespace:** TaxProGST.Controls.GSTR

GSTR Controls are GST Return controls used to file various data to GST Server including Invoices, Credit/Debit Notes, Advance Tax and Tax Paid Detail, and Returns.

### Common Properties of GSTR (and other API enabled) Controls

**Applies to Controls in Assembly and Namespace:** TaxProGST.Controls.GSTR and TaxProGST.Controls.Common

Property	Description	Type
GSTIN	GSTIN of TaxPayer.	String
ShowInstruction	Flag to Show or Hide Instructions and/or Notes on User Controls. False by Default.	bool
ProgressPanel	This property controls ProgressPanel control present on each API enabled control.	TaxProGST.Controls.ProgressControlModel
GstSession	Set this property to APISession for taxpayer. Every API enabled control sets this property to default GSTSession saved in Json configuration using EditAPISetting control. If you want to develop application supporting multiple TaxPayer, assign APISession for the particular TaxPayer. Refer user guide section <a href="#">GST API &gt; API Session</a> for more detail.	TaxProGST.Controls.GSTSession
TaxPayerID TaxPayerName	Pass-through parameters. Used while opening other GSTR Control from one control. Like open GSTR1Return from GSTR1Upload control.	String

# GSTR1Upload: Upload Sale Register (Invoice Data)

Assembly and Namespace: TaxProGST.Controls.GSTR

First step in GST compliance is uploading of Sale Invoices to GST System. *GSTR1Upload* control helps uploading of Sales Register Data to GST server using API.

## ProgressPanel Properties

Property	Description	Data Type
FY	Set Financial Year	String
Month	Set Return Month.	String
FilingPeriod	Filing Period will be calculated if both FY and Month properties are set. It indicates return FilingPeriod in format "MMyyyy".	String

## Events

Event	Description

## GSTR1Return

Assembly and Namespace: TaxProGST.Controls.GSTR

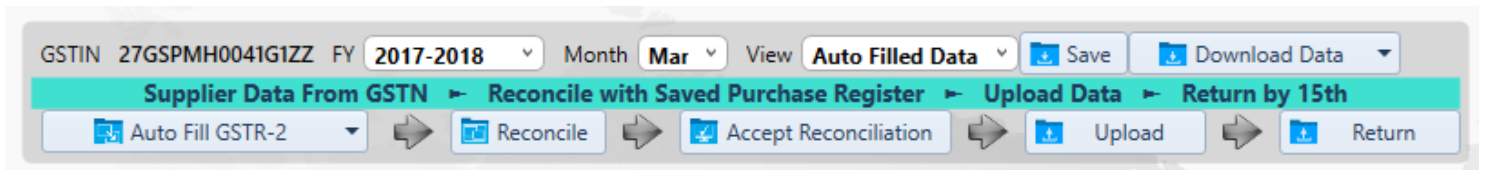
ProgressPanel control has two views.

Property	Description	Data Type
FY		String
Month		String
FilingPeriod		String
SignType		String
SignID		String
SummaryAsOnDate		DateTime
SummaryDesc		String
ReturnAckNo		String

**Please use Object Browser to check Properties, Methods, Events available**

# GSTR2Upload

Assembly and Namespace: TaxProGST.Controls.GSTR



Please use Object Browser to check Properties, Methods, Events available

## GSTR2Return

Assembly and Namespace: TaxProGST.Controls.GSTR

Properties (Refer to common properties listed in topic GSTR Controls)

Property	Description	Data Type
FY		String
Month		String
FilingPeriod		String
SignType		String
SignID		String
ReturnAckNo		String

Please use Object Browser to check Properties, Methods, Events available



# GSTR1AReturn

Assembly and Namespace: TaxProGST.Controls.GSTR

## Properties

Property	Description	Data Type
FY		String
Month		String
FilingPeriod		String
SignType		String
SignID		String
ReturnAckNo		String

Please use Object Browser to check Properties, Methods, Events available

## **GSTR1AUpload**

**Assembly and Namespace:** TaxProGST.Controls.GSTR

**Please use Object Browser to check Properties, Methods, Events available**

## Common Controls

**Assembly and Namespace:** TaxProGST.Controls.Common

Common Controls are controls used to Configure API Settings, Authorize TaxPayer for API access, View Return Filing Status, Get Different Ledgers.

**Common Properties of API enabled Controls:** Refer section [GSTR Controls](#).

## **EditAPISetting**

**Assembly and Namespace:** TaxProGST.Controls.Common

The usages of EditAPISetting control is demonstrated in Windows Forms DEMO Project in UCL Download.

**Please use Object Browser to check Properties, Methods, Events available**

## **APIAuthorization**

**Assembly and Namespace:** TaxProGST.Controls.Common

The usages of EditAPISetting control is demonstrated in Windows Forms DEMO Project in UCL Download.

**Please use Object Browser to check Properties, Methods, Events available**

## Return Common Controls

**Assembly and Namespace:** TaxProGST.Controls.Common

Common API enabled controls to check and track Return and Tax details are grouped under Returns section of this user guide.

# ReturnStatus

Assembly and Namespace: TaxProGST.Controls.Common

## Properties

# AdvTaxPaidDetails

Assembly and Namespace: TaxProGST.Controls.Common

## Properties



# PartialITCDetails

Assembly and Namespace: TaxProGST.Controls.Common

## Properties

## **TDSDetails**

**Assembly and Namespace:** TaxProGST.Controls.Common

**Not Released**

# Ledger Common Controls

**Assembly and Namespace:** TaxProGST.Controls.Common

Common API enabled controls to get different ledgers viz. Cash Ledger, ITC Ledger (Input Tax Credit), and Liability Ledger are grouped under Ledgers section of this user guide.

# CashLedger

Assembly and Namespace: TaxProGST.Controls.Common

**Please use Object Browser to check Properties, Methods, Events available**

# ITCLedger

Assembly and Namespace: TaxProGST.Controls.Common

**Please use Object Browser to check Properties, Methods, Events available**

# LiabilityLedger

Assembly and Namespace: TaxProGST.Controls.Common

Please use Object Browser to check Properties, Methods, Events available

## UtilizeCash

Assembly and Namespace: TaxProGST.Controls.Common

Please use Object Browser to check Properties, Methods, Events available

# UtilizeITC

Assembly and Namespace: TaxProGST.Controls.Common

**Please use Object Browser to check Properties, Methods, Events available**



# TaxProGST Core

**Assembly and Namespace:** TaxProGST.Core

Implements [Database Models](#) and Data Copy and Cloning Tools. User would be more interested in [Database Models](#) to built their storage engine to back controls in TaxPro GST User Control Library.

# GSTRTools

**Assembly and Namespace:** TaxProGST.Core

GSTRTools are used internally by TaxPro GST User Control Library to copy data to-and-fro from DBModels to JsonModels. Generally developers using TaxPro GST User Control Library does not require to access this part of the library. Advanced users may explore the methods available in this assembly.

**Please use Object Browser to check Properties, Methods, Events available**

## Database Models

**Assembly and Namespace:** TaxProGST.Core

Implements Database Models used by Microsoft Entity Framework to implement DbContext. These classes are ready with all DataAnnotations like [Key], [MaxLength], [ForeignKey], etc

**Please use Object Browser to check Properties, Methods, Events available**

# Sample Project

Sample Project may be downloaded from

<http://taxpro.co.in/TaxPro/ProductDownloads.aspx>