WORKING WITH EXTERNAL TOOLS

DireXions 2024

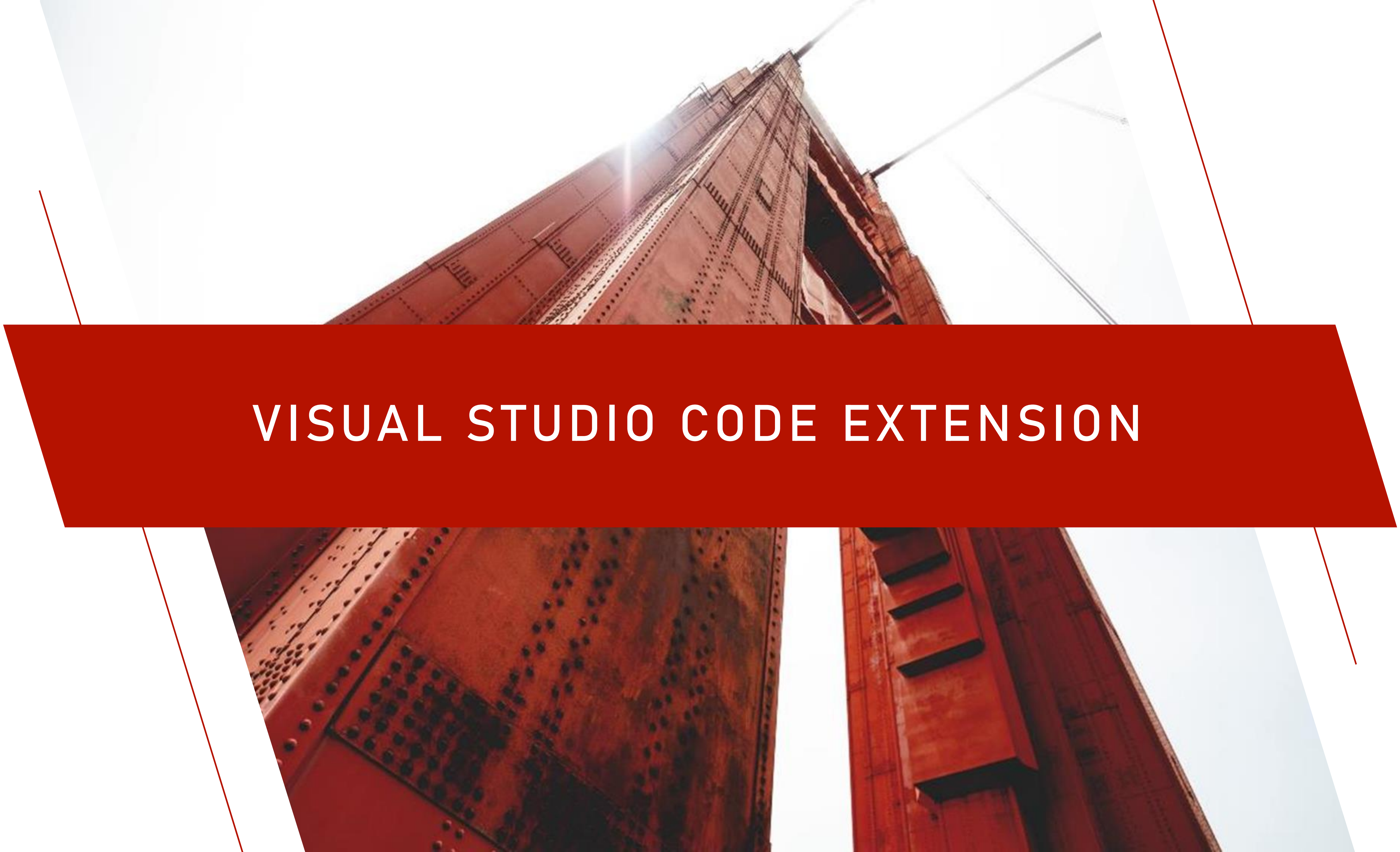
AGENDA

Visual Studio Code Extension

Google Workspace Objects

Working with Web Services

PayPal and Salesforce Examples



VISUAL STUDIO CODE EXTENSION

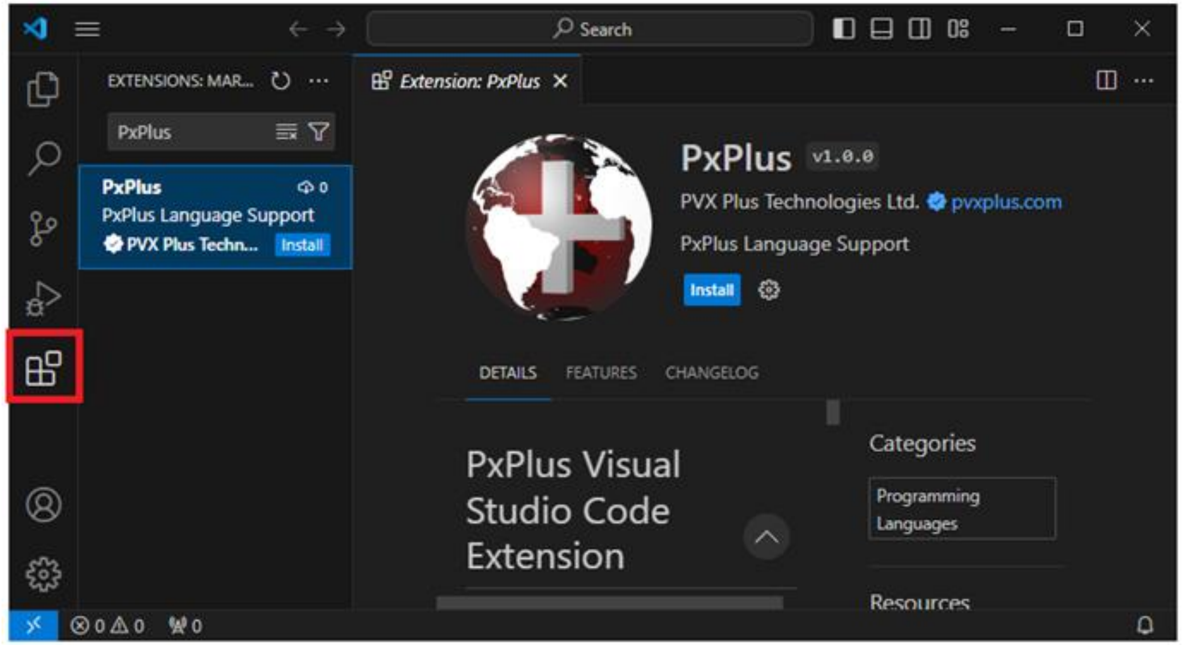
VISUAL STUDIO CODE EXTENSION

- PxPlus Visual Studio Code Extension is an extension for Visual Studio Code to enable working with PxPlus programs
- PxPlus 2024 or newer is required
- It provides the following functionality:
  - Create new PxPlus programs
  - Open and edit existing PxPlus programs
  - Work with binary PxPlus programs
  - Work with text-based PxPlus programs
  - PxPlus syntax highlighting
  - PxPlus program error diagnostics
  - Current document-based Auto Complete

VISUAL STUDIO CODE EXTENSION

How to Install PxPlus Extension in Visual Studio Code

1. Download and install Visual Studio Code from the Visual Studio Code Web site https://code.visualstudio.com/
2. On the Activity Bar on the far left side of Visual Studio Code, click the Extension icon
   This displays the **Extensions** view and the **Extensions Marketplace**
3. Find the PxPlus extension by entering **PxPlus** in the **Search Extensions in Marketplace** input box
4. In the list that displays, click on **PxPlus** and then click the **Install** button to install the PxPlus extension

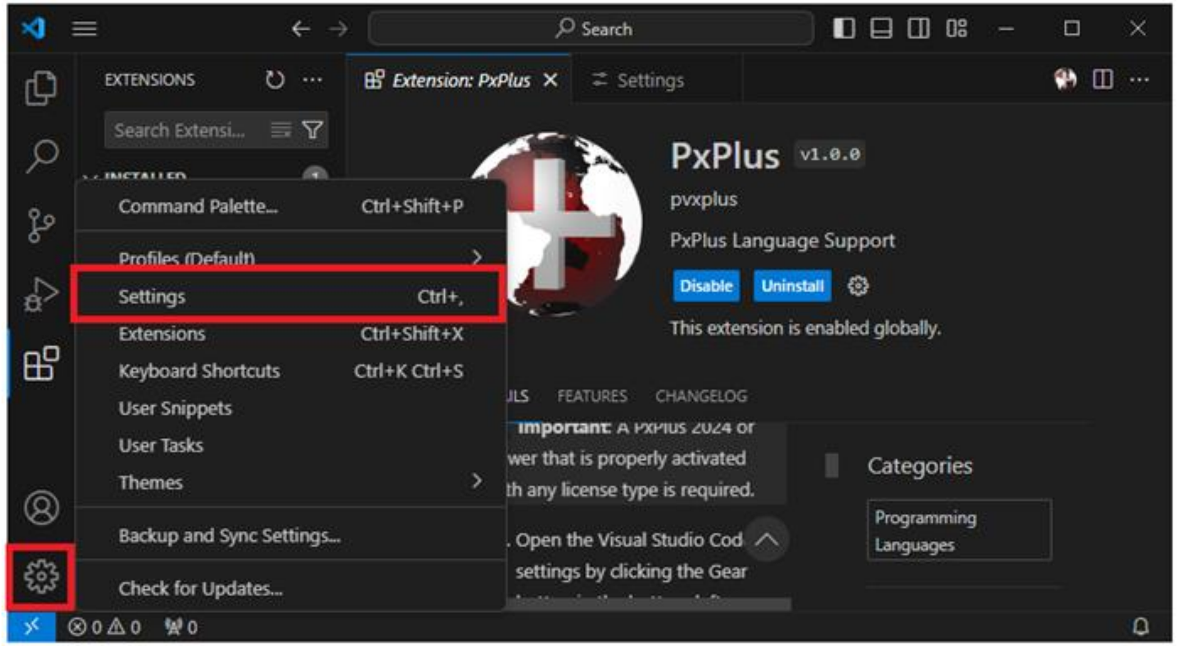


VISUAL STUDIO CODE EXTENSION

How to Set Up PxPlus Extension in Visual Studio Code

Before you can use the PxPlus Visual Studio Code extension, you must tell the extension where PxPlus is installed:

- Open the Visual Studio Code settings by clicking the Gear icon in the bottom left corner and then select **Settings** from the menu that displays
- Expand the **Extensions** node and then select the **PxPlus** node
- Under **PxPlus Path,** enter the path to the directory where the PxPlus executable is located (i.e. C:\PVX Plus Technologies\PxPlus 2024)



This extension is enabled globally.

Important: A PxPlus 2024 or
wer that is properly activated
th any license type is required.

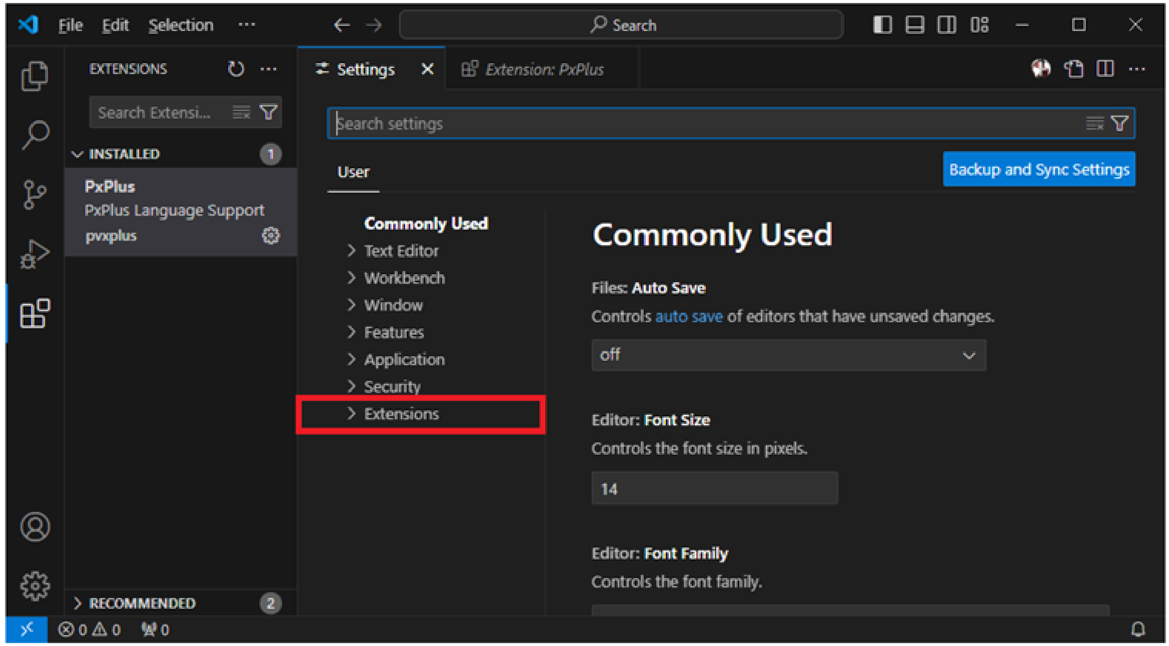
. Open the Visual Studio Cod
settings by clicking the Gear

VISUAL STUDIO CODE EXTENSION

How to Set Up PxPlus Extension in Visual Studio Code

Before you can use the PxPlus Visual Studio Code extension, you must tell the extension where PxPlus is installed:

- Open the Visual Studio Code settings by clicking the Gear icon in the bottom left corner and then select **Settings** from the menu that displays
- Expand the **Extensions** node and then select the **PxPlus** node
- Under **PxPlus Path**, enter the path to the directory where the PxPlus executable is located (i.e. C:\PVX Plus Technologies\PxPlus 2024)



Commonly Used

Files: Auto Save
Controls auto save of editors that have unsaved changes.
off

Editor: Font Size
Controls the font size in pixels.
14

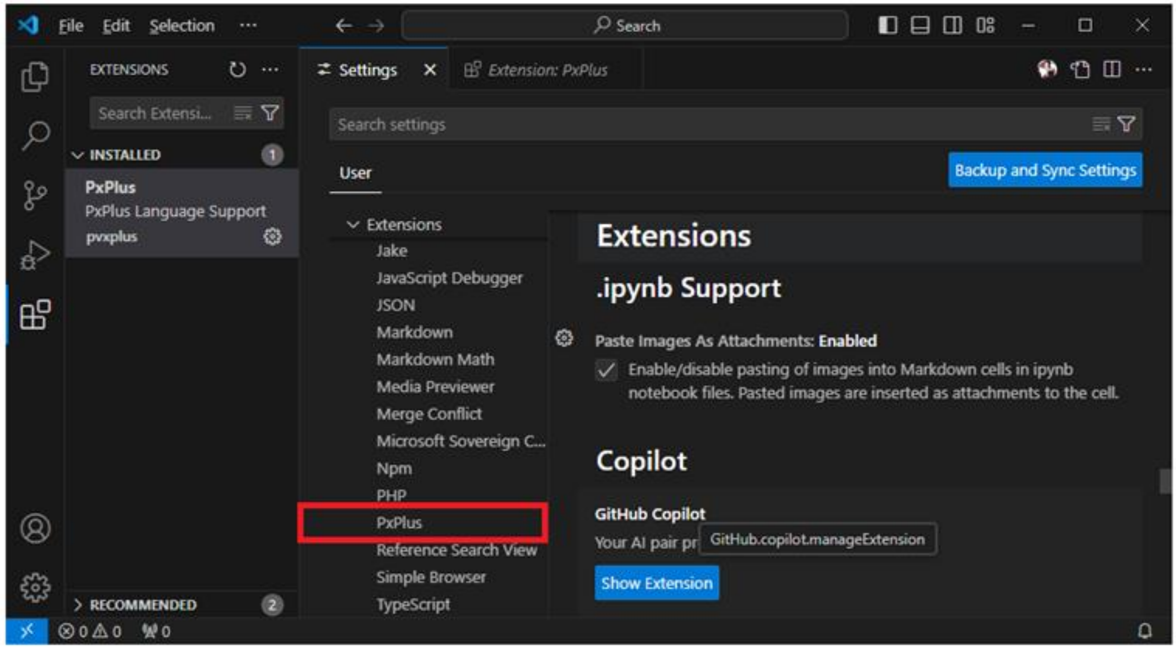
Editor: Font Family
Controls the font family.

VISUAL STUDIO CODE EXTENSION

How to Set Up PxPlus Extension in Visual Studio Code

Before you can use the PxPlus Visual Studio Code extension, you must tell the extension where PxPlus is installed:

- Open the Visual Studio Code settings by clicking the Gear icon in the bottom left corner and then select **Settings** from the menu that displays
- Expand the **Extensions** node and then select the **PxPlus** node
- Under **PxPlus Path,** enter the path to the directory where the PxPlus executable is located (i.e. C:\PVX Plus Technologies\PxPlus 2024)



Paste Images As Attachments: **Enabled**

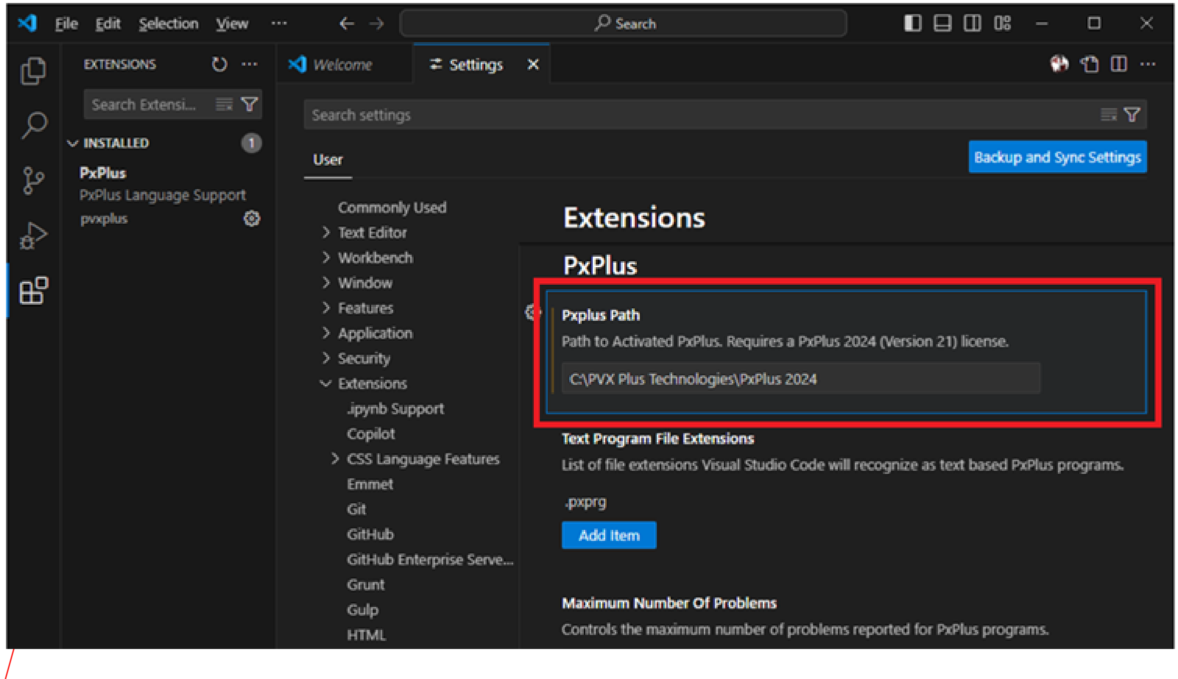
☑ Enable/disable pasting of images into Markdown cells in ipynb notebook files. Pasted images are inserted as attachments to the cell.

VISUAL STUDIO CODE EXTENSION

How to Set Up PxPlus Extension in Visual Studio Code

Before you can use the PxPlus Visual Studio Code extension, you must tell the extension where PxPlus is installed:

- Open the Visual Studio Code settings by clicking the Gear icon in the bottom left corner and then select **Settings** from the menu that displays
- Expand the **Extensions** node and then select the **PxPlus** node
- Under **PxPlus Path**, enter the path to the directory where the PxPlus executable is located (i.e. C:\PVX Plus Technologies\PxPlus 2024)



Extensions

PxPlus

Pxplus Path
Path to Activated PxPlus. Requires a PxPlus 2024 (Version 21) license.
C:\PVX Plus Technologies\PxPlus 2024

Text Program File Extensions
List of file extensions Visual Studio Code will recognize as text based PxPlus programs.
.pxprg

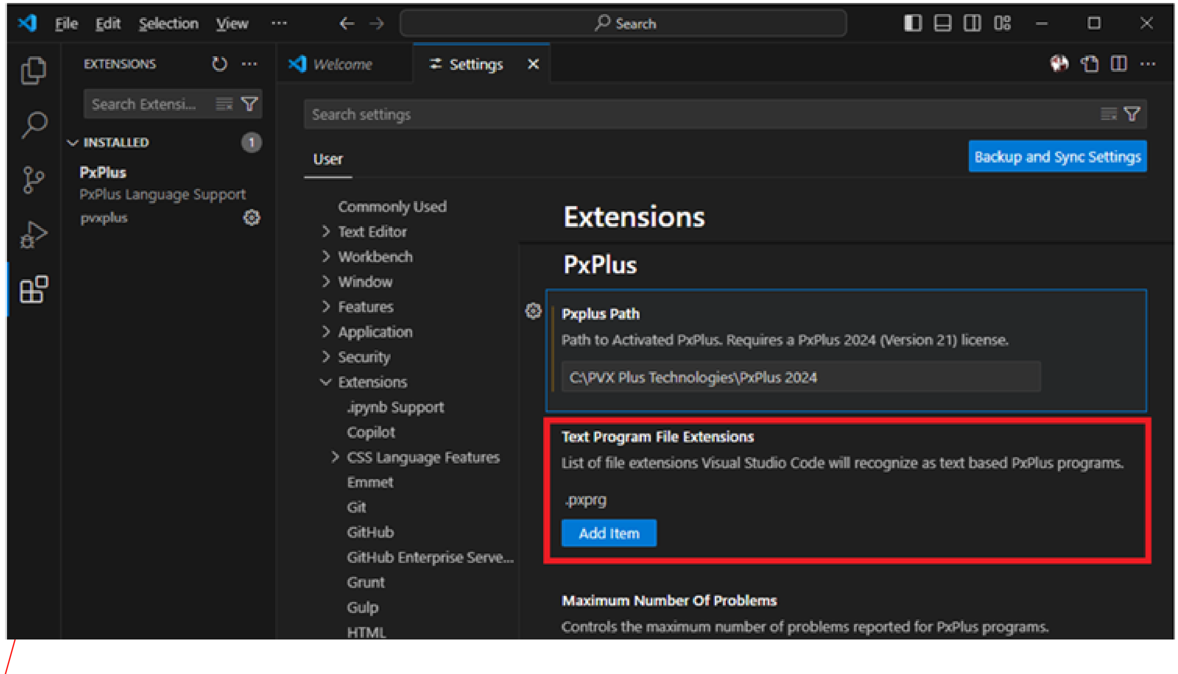
Maximum Number Of Problems
Controls the maximum number of problems reported for PxPlus programs.

VISUAL STUDIO CODE EXTENSION

How to Set Up PxPlus Extension in Visual Studio Code

If you plan to work with text-based PxPlus programs, then you must also specify which file extensions Visual Studio Code should consider a text-based PxPlus program:

- Under **Text Program File Extensions**, add a file extension to the list by clicking the **Add Item** button, then enter a *.xxx* file extension and then click OK
- To add more than one file extension, repeat this step as many times as needed
- By default, the only extension on the list is *.pxprg*



Extensions

PxPlus

Pxplus Path

Path to Activated PxPlus. Requires a PxPlus 2024 (Version 21) license.

C:\PVX Plus Technologies\PxPlus 2024

Text Program File Extensions

List of file extensions Visual Studio Code will recognize as text based PxPlus programs.

.pxprg

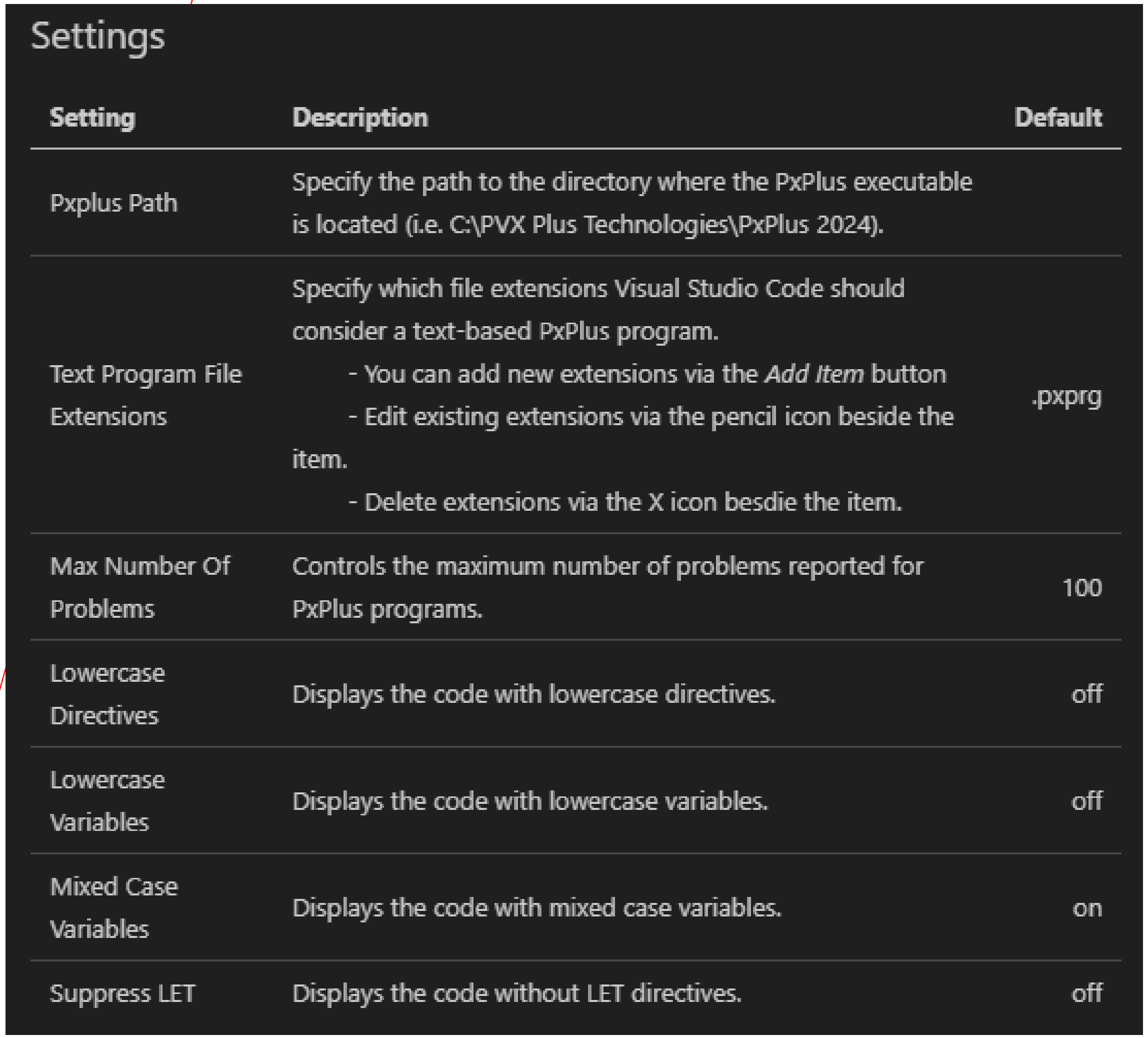
Add Item

Maximum Number Of Problems

Controls the maximum number of problems reported for PxPlus programs.

VISUAL STUDIO CODE EXTENSION

- **The remaining settings are optional and**
  - Control the number of errors reported before it stops reporting new errors
  - Control how the program is displayed/saved
    - These settings are the same for all PxPlus program editors



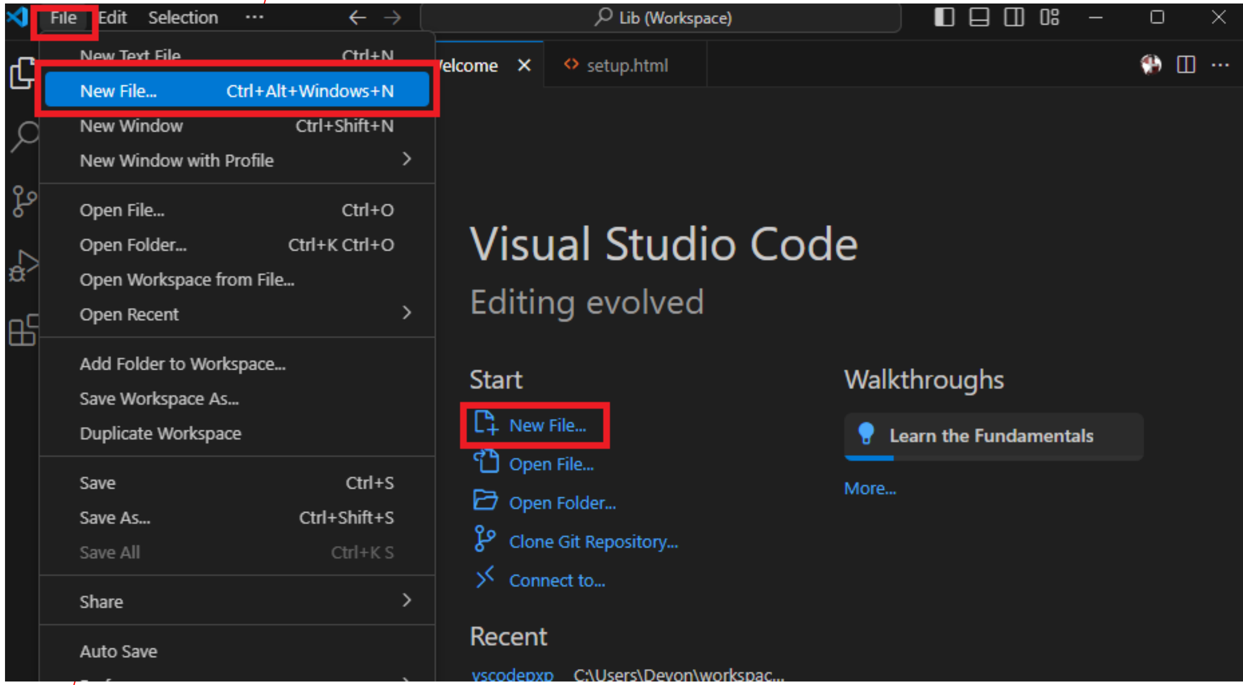
Settings

Setting	Description	Default
Pxplus Path	Specify the path to the directory where the PxPlus executable is located (i.e. C:\PVX Plus Technologies\PxPlus 2024).	
Text Program File Extensions	Specify which file extensions Visual Studio Code should consider a text-based PxPlus program. - You can add new extensions via the *Add Item* button - Edit existing extensions via the pencil icon beside the item. - Delete extensions via the X icon besdie the item.	.pxprg
Max Number Of Problems	Controls the maximum number of problems reported for PxPlus programs.	100
Lowercase Directives	Displays the code with lowercase directives.	off
Lowercase Variables	Displays the code with lowercase variables.	off
Mixed Case Variables	Displays the code with mixed case variables.	on
Suppress LET	Displays the code without LET directives.	off

VISUAL STUDIO CODE EXTENSION

To Create New PxPlus Programs

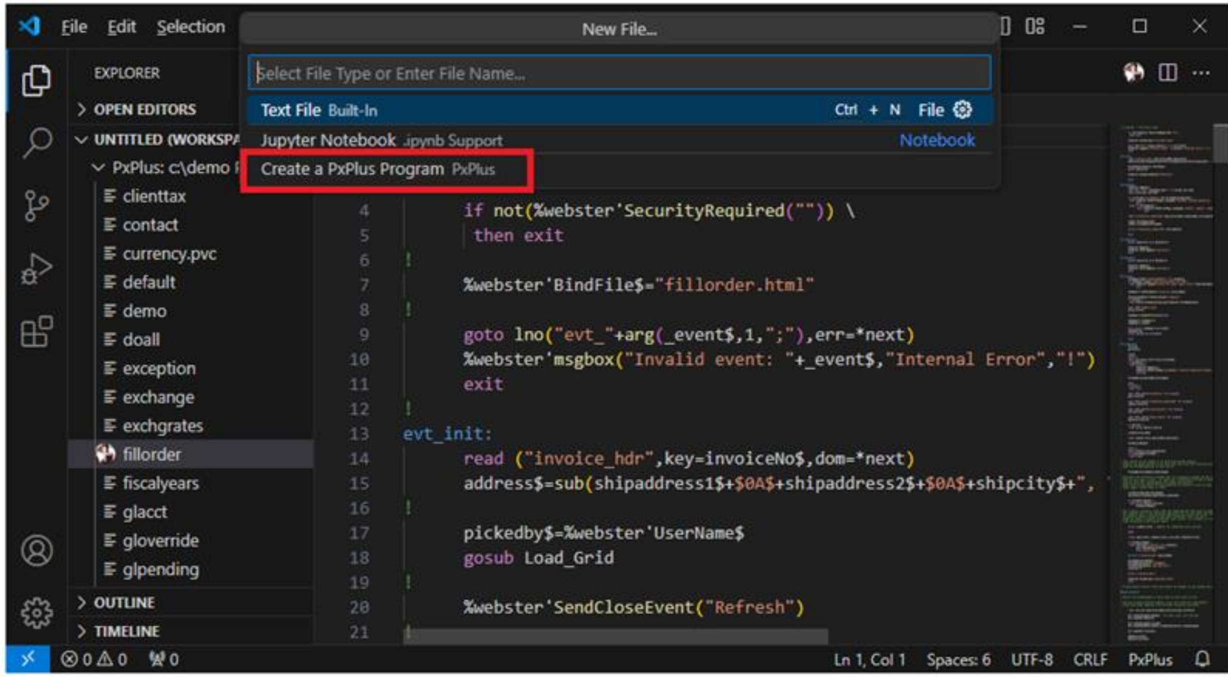
- From the File menu or from the Welcome screen, select **New File**



VISUAL STUDIO CODE EXTENSION

To Create New PxPlus Programs

- A list of possible file types to create will display
- From the list, select **Create a PxPlus Program**

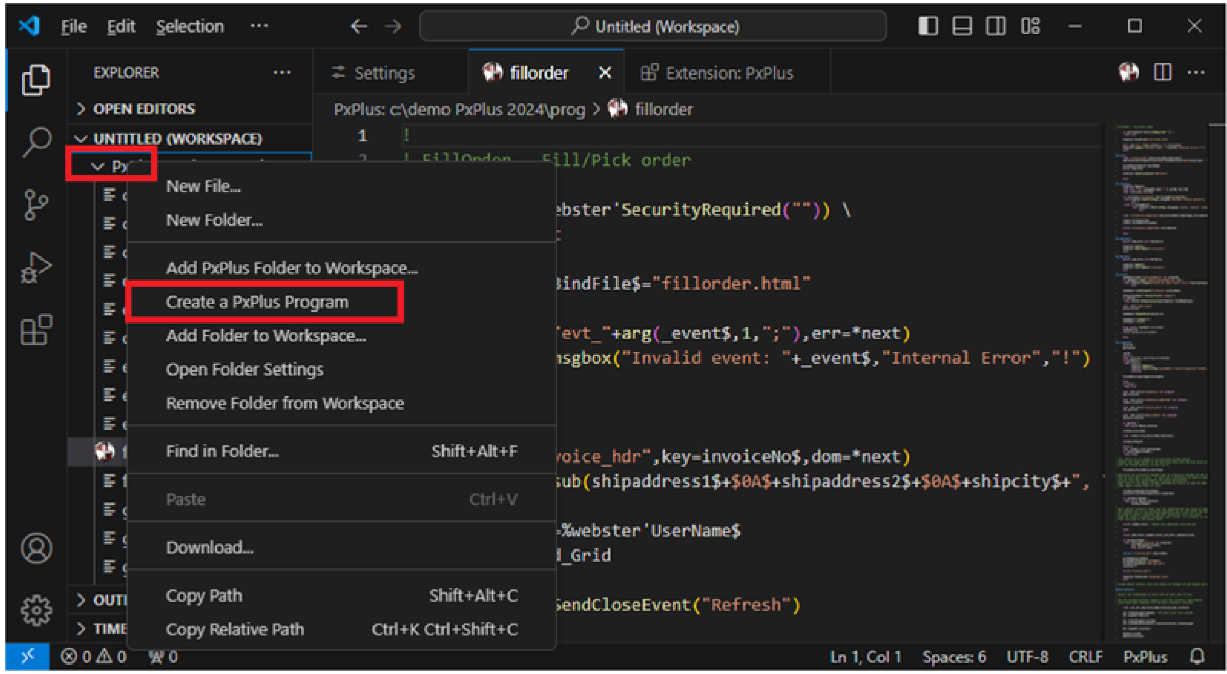


```
      if not(%webster'SecurityRequired("")) \
       then exit
!
      %webster'BindFile$="fillorder.html"
!
      goto lno("evt_"+arg(_event$,1,";"),err=*next)
      %webster'msgbox("Invalid event: "+_event$,"Internal Error","!")
      exit
!
evt_init:
      read ("invoice_hdr",key=invoiceNo$,dom=*next)
      address$=sub(shipaddress1$+$0A$+shipaddress2$+$0A$+shipcity$+",
!
      pickedby$=%webster'UserName$
      gosub Load_Grid
!
      %webster'SendCloseEvent("Refresh")
!
```

VISUAL STUDIO CODE EXTENSION

To Create New PxPlus Programs

- Alternatively right click in the Explorer and select **Create a PxPlus Program**

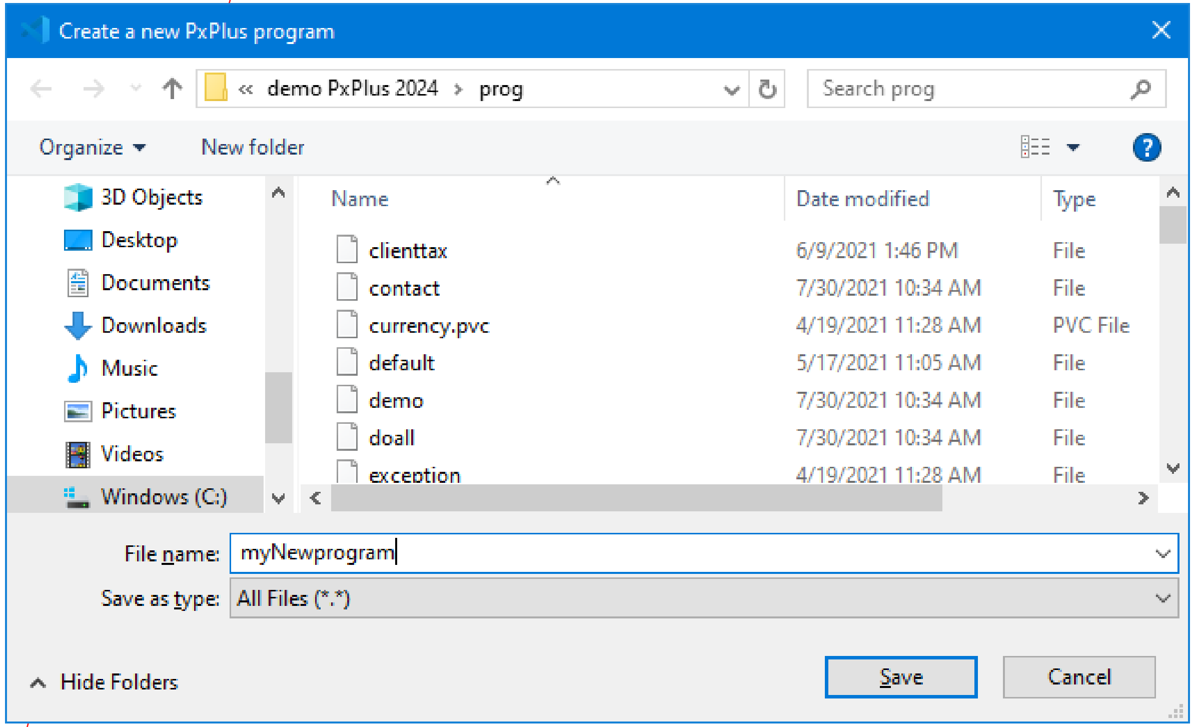


```
!
! FillOrder - Fill/Pick order
ebster'SecurityRequired("")) \
BindFile$="fillorder.html"
'evt_"+arg(_event$,1,";"),err=*next)
nsgbox("Invalid event: "+_event$,"Internal Error","!")
voice_hdr",key=invoiceNo$,dom=*next)
sub(shipaddress1$+$0A$+shipaddress2$+$0A$+shipcity$+",
=%webster'UserName$
l_Grid
SendCloseEvent("Refresh")
```

VISUAL STUDIO CODE EXTENSION

To Create New PxPlus Programs

- Use the Save dialog to choose a pathname for your new program



Name	Date modified	Type
clienttax	6/9/2021 1:46 PM	File
contact	7/30/2021 10:34 AM	File
currency.pvc	4/19/2021 11:28 AM	PVC File
default	5/17/2021 11:05 AM	File
demo	7/30/2021 10:34 AM	File
doall	7/30/2021 10:34 AM	File
exception	4/19/2021 11:28 AM	File

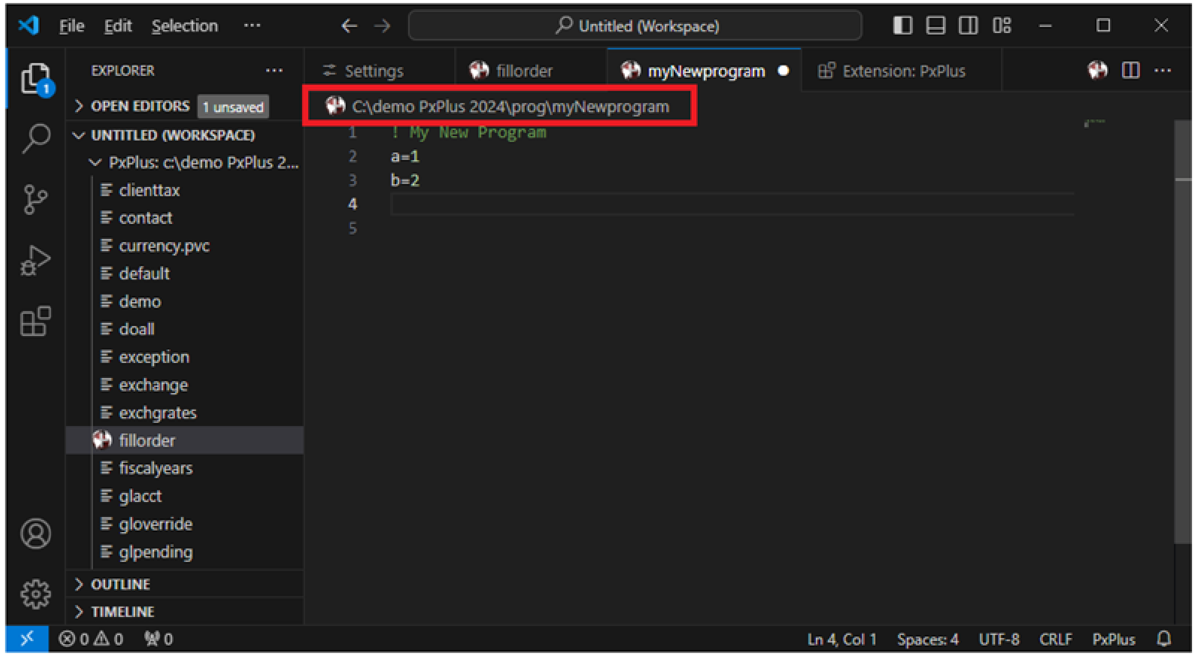
File name: myNewprogram

Save as type: All Files (*.*)

VISUAL STUDIO CODE EXTENSION

To Create New PxPlus Programs

- Now that you have a new PxPlus program you can begin coding



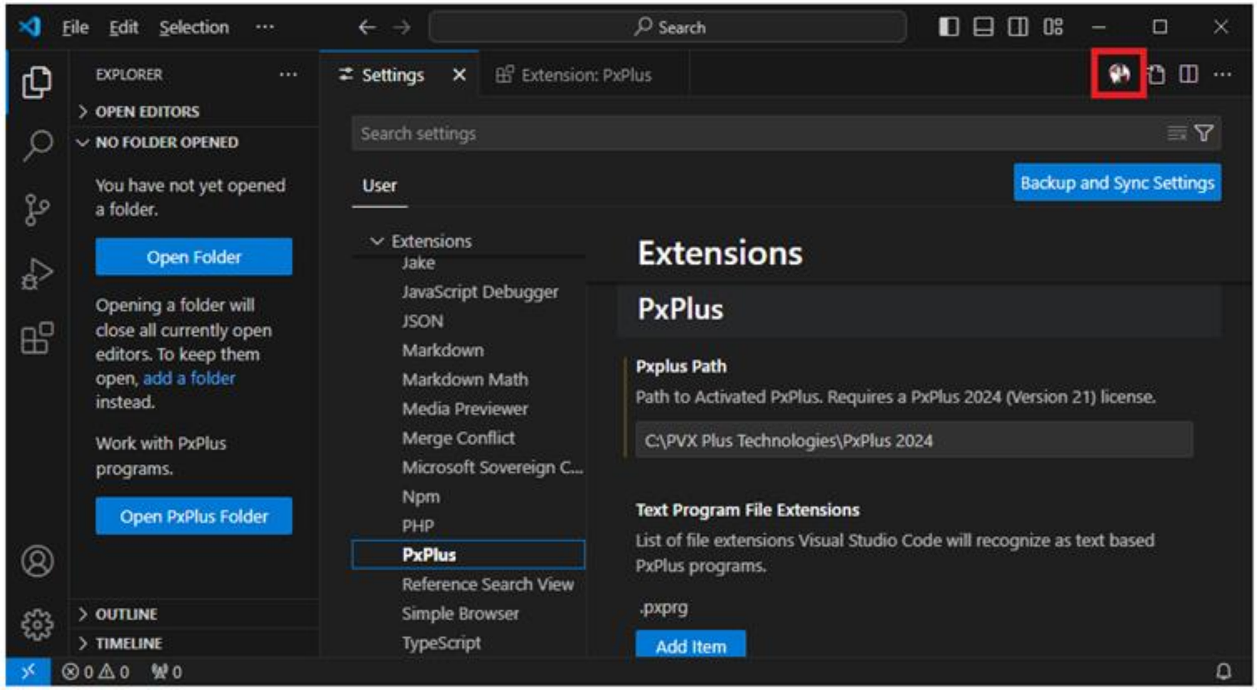
C:\demo PxPlus 2024\prog\myNewprogram

```
! My New Program
a=1
b=2
```

VISUAL STUDIO CODE EXTENSION

To Edit Existing PxPlus Programs

- Add a PxPlus folder to the workspace
  - Click on the PxPlus icon in the top right corner of an Edit window



You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

Work with PxPlus programs.

Extensions

PxPlus

Pxplus Path

Path to Activated PxPlus. Requires a PxPlus 2024 (Version 21) license.

C:\PVX Plus Technologies\PxPlus 2024

Text Program File Extensions

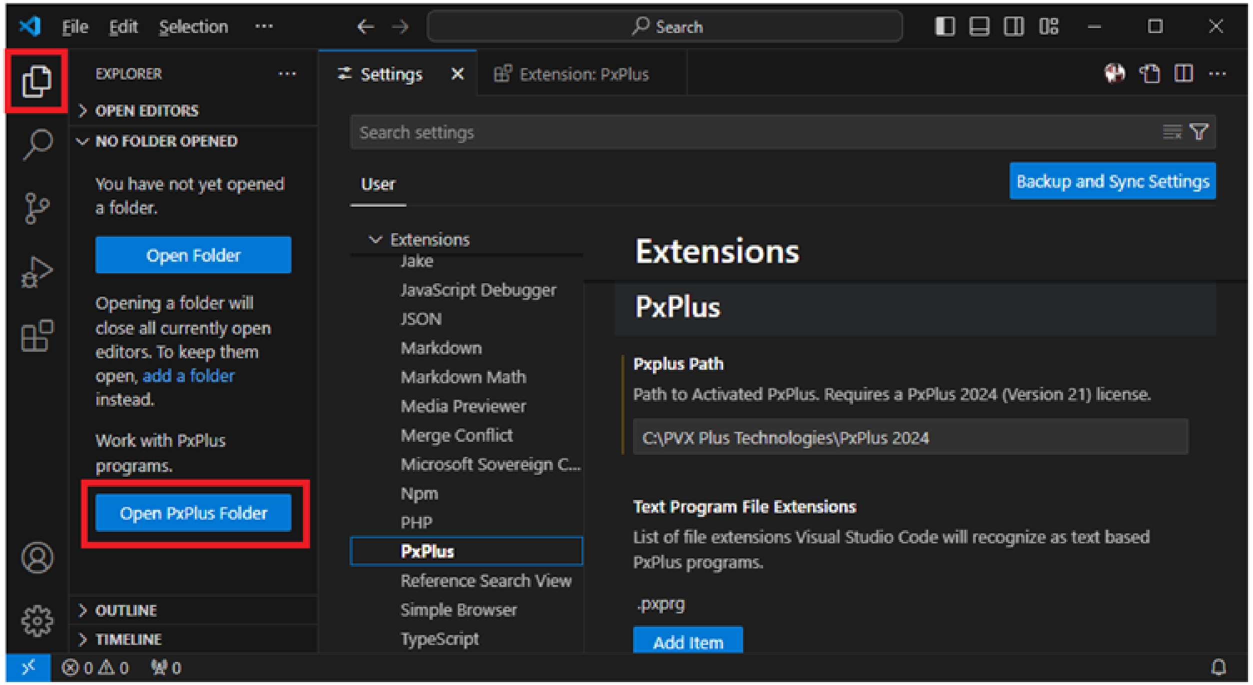
List of file extensions Visual Studio Code will recognize as text based PxPlus programs.

.pxprg

VISUAL STUDIO CODE EXTENSION

To Edit Existing PxPlus Programs

- Add a PxPlus folder to the workspace
  - Alternatively click the **Open PxPlus Folder** button in the Explorer window when no folders are open



You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

Work with PxPlus programs.

Extensions

PxPlus

Pxplus Path

Path to Activated PxPlus. Requires a PxPlus 2024 (Version 21) license.

C:\PVX Plus Technologies\PxPlus 2024

Text Program File Extensions

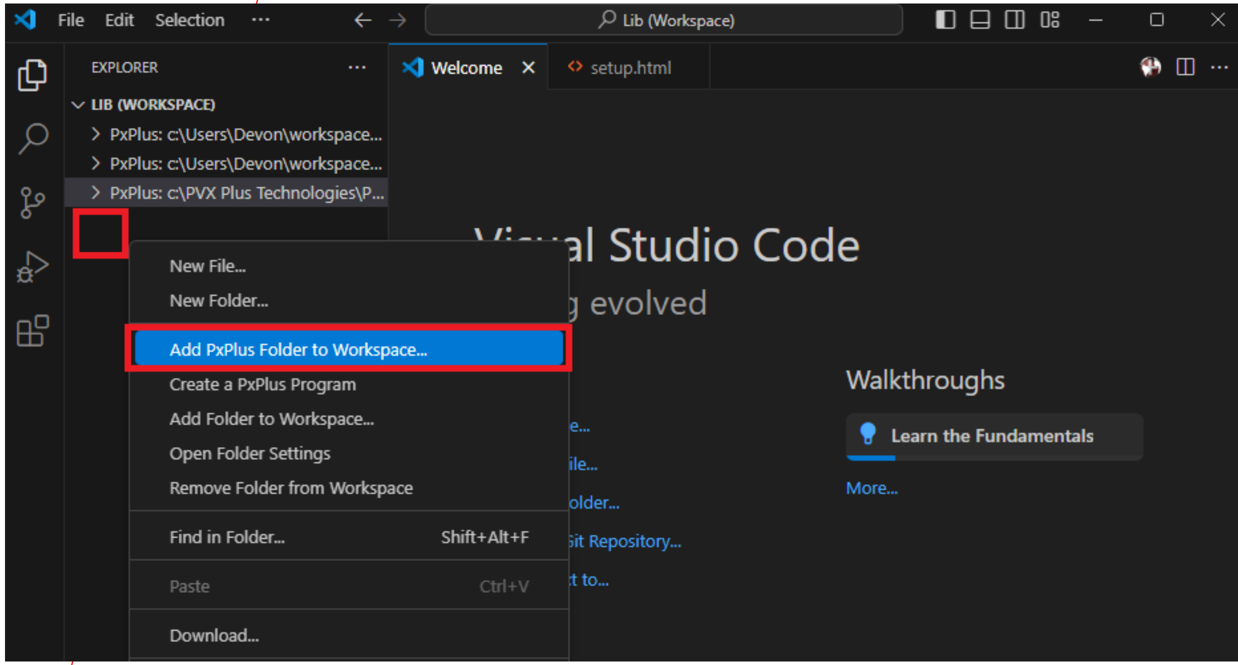
List of file extensions Visual Studio Code will recognize as text based PxPlus programs.

.pxprg

VISUAL STUDIO CODE EXTENSION

To Edit Existing PxPlus Programs

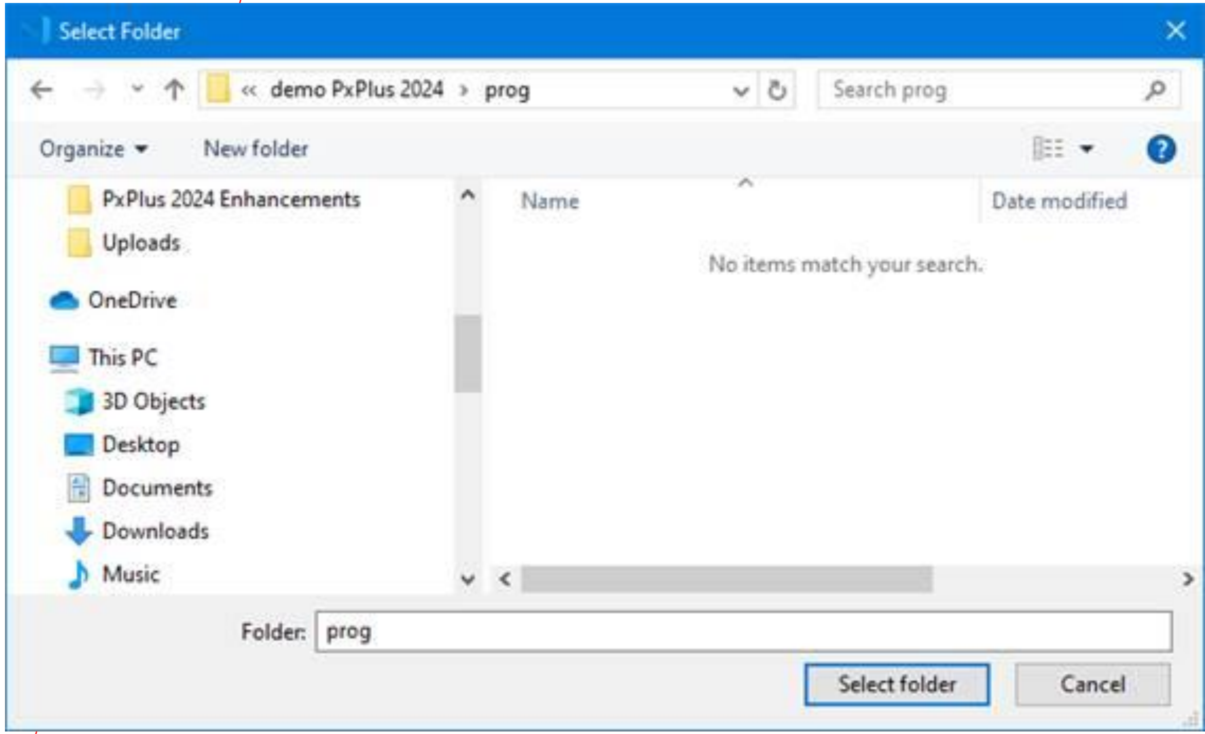
- Add a PxPlus folder to the workspace
  - Another way is to click the **Add PxPlus Folder to Workspace...** menu item from the Explorer right click menu



VISUAL STUDIO CODE EXTENSION

To Edit Existing PxPlus Programs

- Select a PxPlus folder to add to the workspace



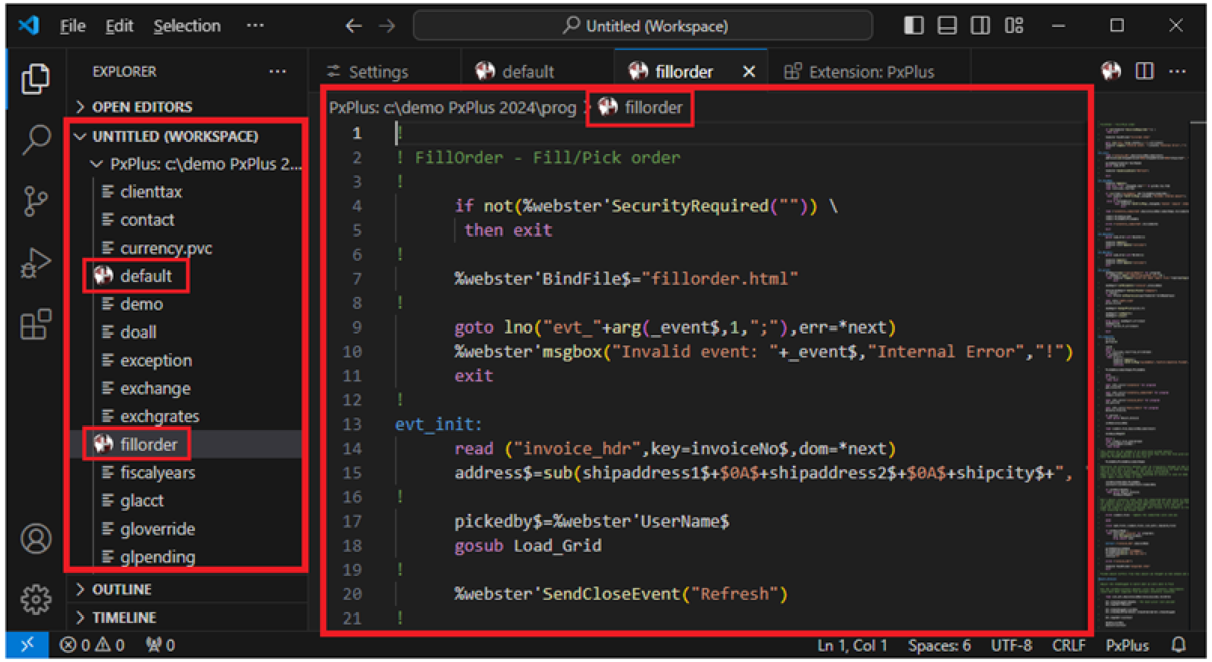
No items match your search.

Folder: prog

VISUAL STUDIO CODE EXTENSION

To Edit Existing PxPlus Programs

- Any PxPlus programs or text-based PxPlus programs that are opened via this workspace will be recognized as a PxPlus program by Visual Studio Code
- This enables all the features and allows PxPlus programs to be edited

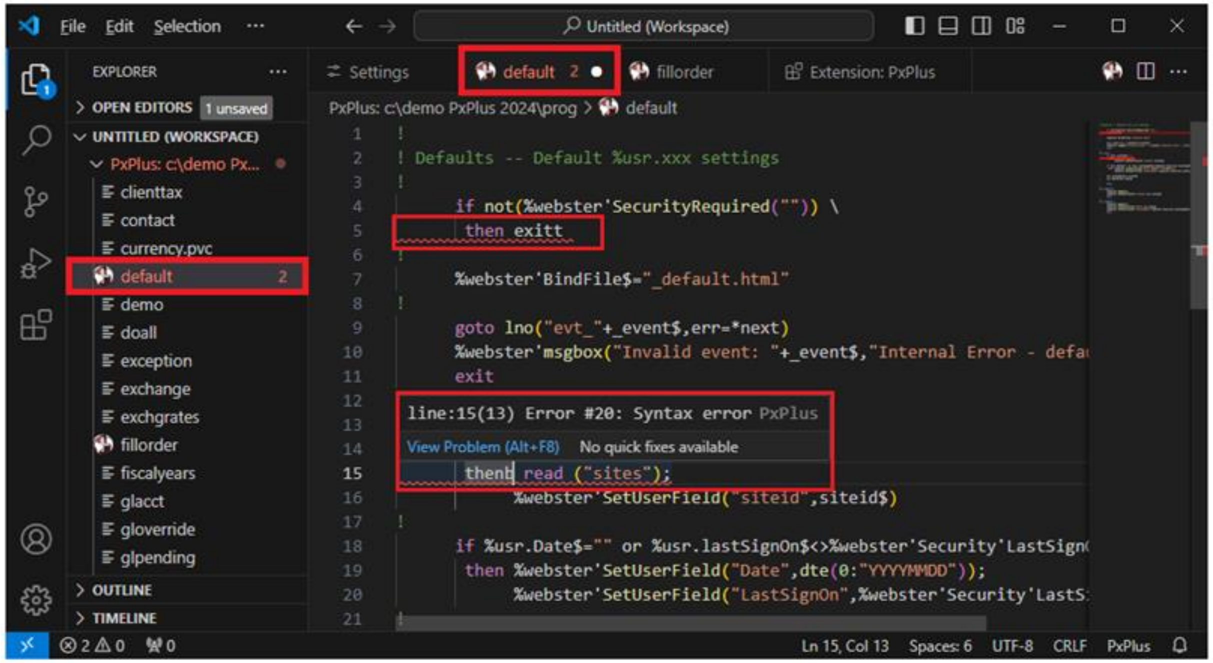


```
!
! FillOrder - Fill/Pick order
!
      if not(%webster'SecurityRequired("")) \
       then exit
!
      %webster'BindFile$="fillorder.html"
!
      goto lno("evt_"+arg(_event$,1,";"),err=*next)
      %webster'msgbox("Invalid event: "+_event$,"Internal Error","!")
      exit
!
evt_init:
      read ("invoice_hdr",key=invoiceNo$,dom=*next)
      address$=sub(shipaddress1$+$0A$+shipaddress2$+$0A$+shipcity$+",
!
      pickedby$=%webster'UserName$
      gosub Load_Grid
!
      %webster'SendCloseEvent("Refresh")
!
```

VISUAL STUDIO CODE EXTENSION

To Edit Existing PxPlus Programs

- Syntax errors are identified with a wavy line, and the number or errors are displayed beside the program name
- Hovering over the error displays a dialog that provides additional details about the error



```
1   !
2   ! Defaults -- Default %usr.xxx settings
3   !
4         if not(%webster'SecurityRequired("")) \
5          then exitt
6   !
7         %webster'BindFile$="_default.html"
8   !
9         goto lno("evt_"+_event$,err=*next)
10        %webster'msgbox("Invalid event: "+_event$,"Internal Error - defa
11        exit
12
13
14
15         thent read ("sites");
16              %webster'SetUserField("siteid",siteid$)
17  !
18        if %usr.Date$="" or %usr.lastSignOn$<>%webster'Security'LastSign
19         then %webster'SetUserField("Date",dte(0:"YYYYMMDD"));
20              %webster'SetUserField("LastSignOn",%webster'Security'LastS
21  !
```

```
line:15(13) Error #20: Syntax error PxPlus
View Problem (Alt+F8)    No quick fixes available
```

GOOGLE WORKSPACE OBJECTS

GOOGLE WORKSPACE OBJECTS

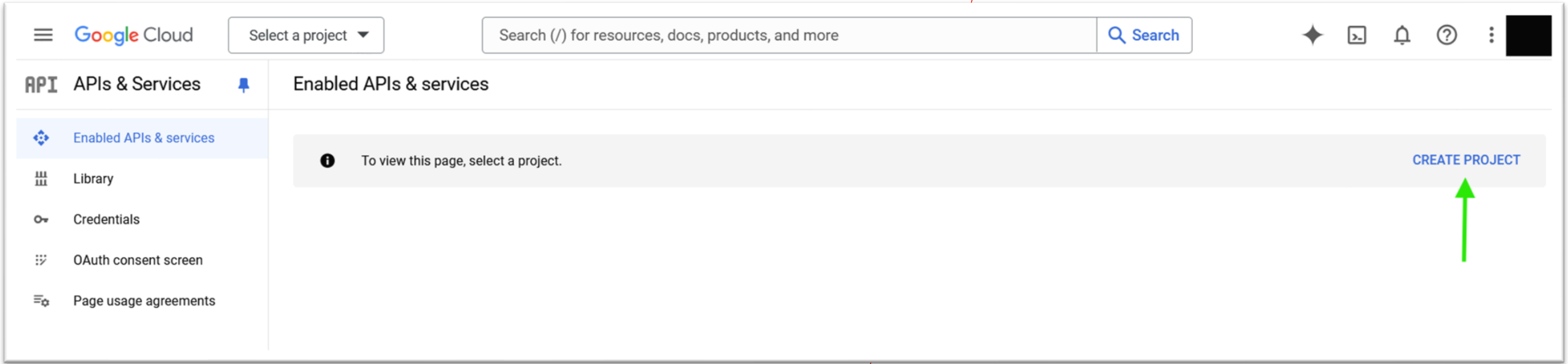
PxPlus objects that were created to work with the Google Workspace® cloud-based applications

- **Google Drive (*obj/GoogleDrive) - Works with the Google Workspace® Drive file storage application**
  - List files
  - Upload/Download files
  - Create folders
  - Export files
  - Delete files
  - Copy files
- **Google Docs (*obj/GoogleDocs) - Works with the Google Workspace® Docs word processing application**
  - Create document
  - Insert/Append/Delete text
  - Find and replace/insert text
  - Set font
  - Find text
  - Set color
  - Uses Drive object
  - Export as Word document
- **Google Sheets (*obj/GoogleSheets) - Works with the Google Workspace® Sheets spreadsheet application**
  - Create spreadsheet
  - Read/Write to cells
  - Find & replace data in cells
  - Create sheets
  - Create/Modify/Delete named ranges
  - Insert/Delete Columns
  - Insert/Delete Rows
  - Set font
  - Set number format
  - Set color
  - Uses Drive object
  - Export as Excel spreadsheet

GOOGLE WORKSPACE OBJECTS

- Need to do the Google API App Setup to obtain a Client ID and Client Secret
- Must allow access to the needed Google APIs
- To test the Google Workspace objects, you need to add, as a "Test User", the Google account that you will use to test the objects and whose Drive, Docs and/or Sheets you will access
- Once testing is done, you can publish the app, which makes it available for others (besides Test Users) and removes the unverified warning
    - Publication requires Google to verify the app, and Google informs you about what this process requires when you publish the app

GOOGLE WORKSPACE OBJECTS

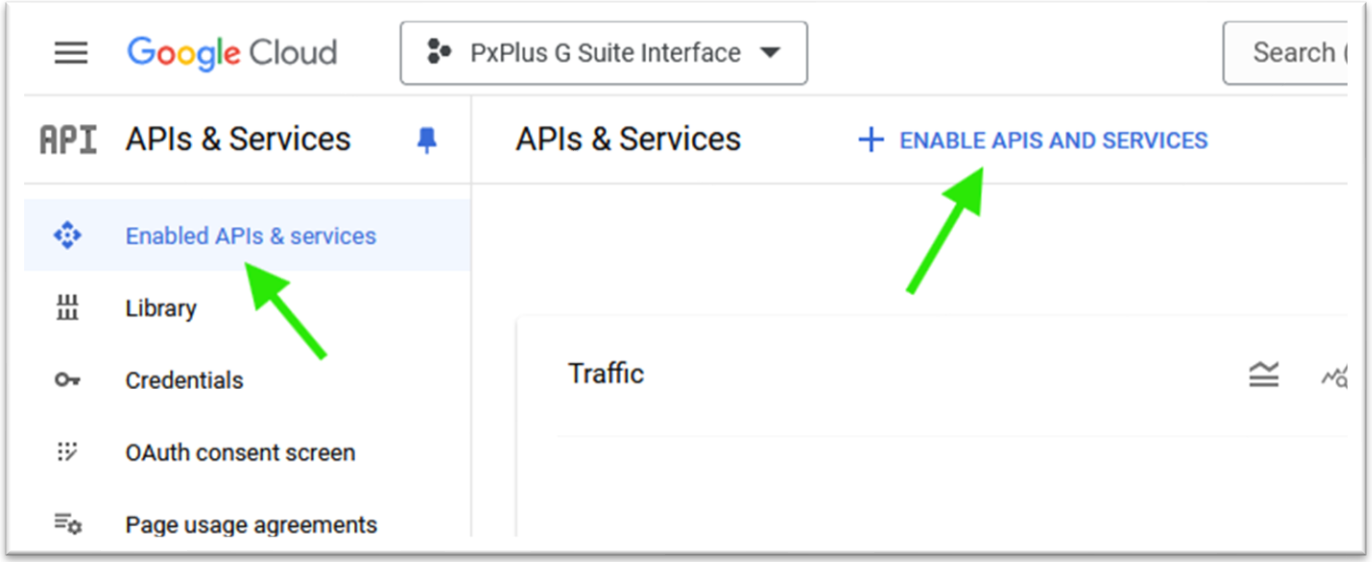


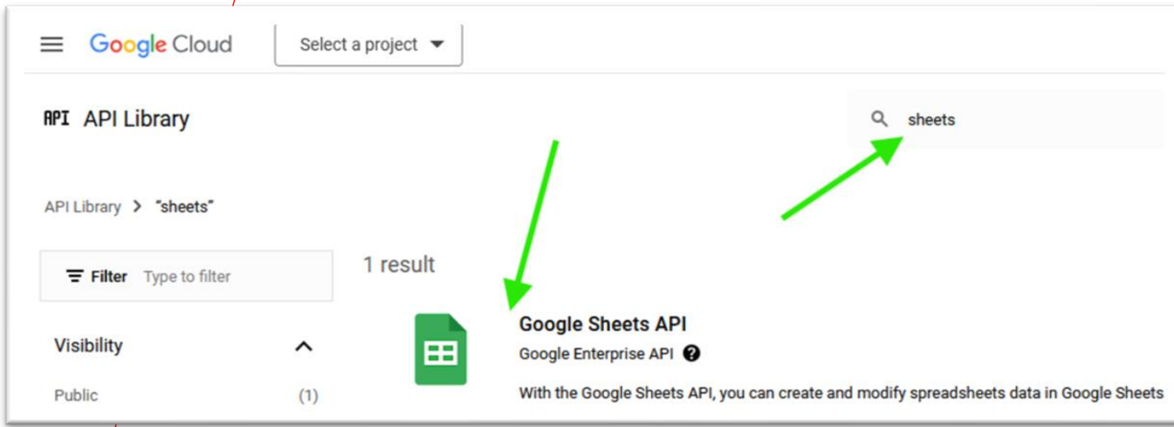
Enabled APIs & services

To view this page, select a project.

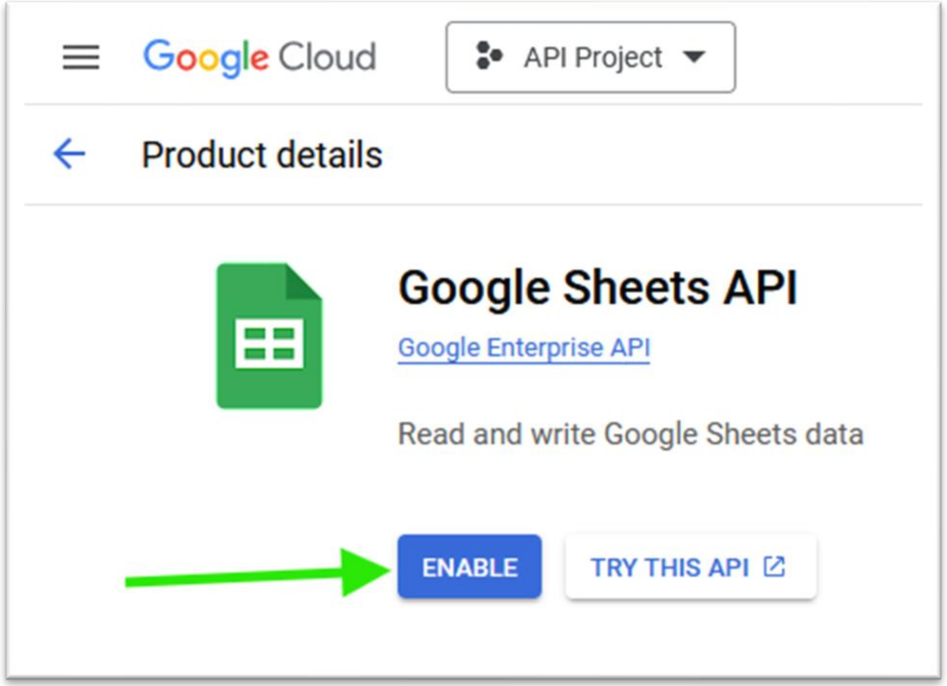
- Go to the [Google API Console]
- Create or select a project
- Input a name and if needed an organization
- Click CREATE

GOOGLE WORKSPACE OBJECTS





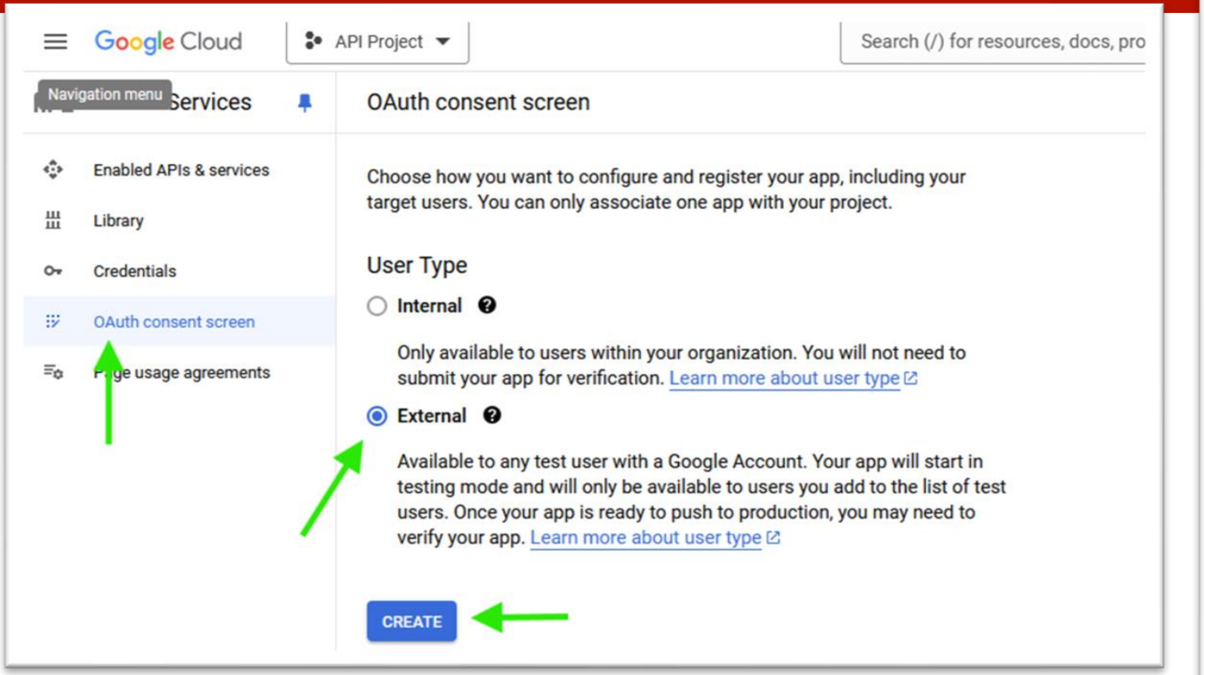
With the Google Sheets API, you can create and modify spreadsheets data in Google Sheets



Read and write Google Sheets data

- **Allow access to Google APIs**
  - Enable Google Drive API
  - Enable Google Docs API
  - Enable Google Sheets API

GOOGLE WORKSPACE OBJECTS



OAuth consent screen

Choose how you want to configure and register your app, including your target users. You can only associate one app with your project.

User Type

○ Internal

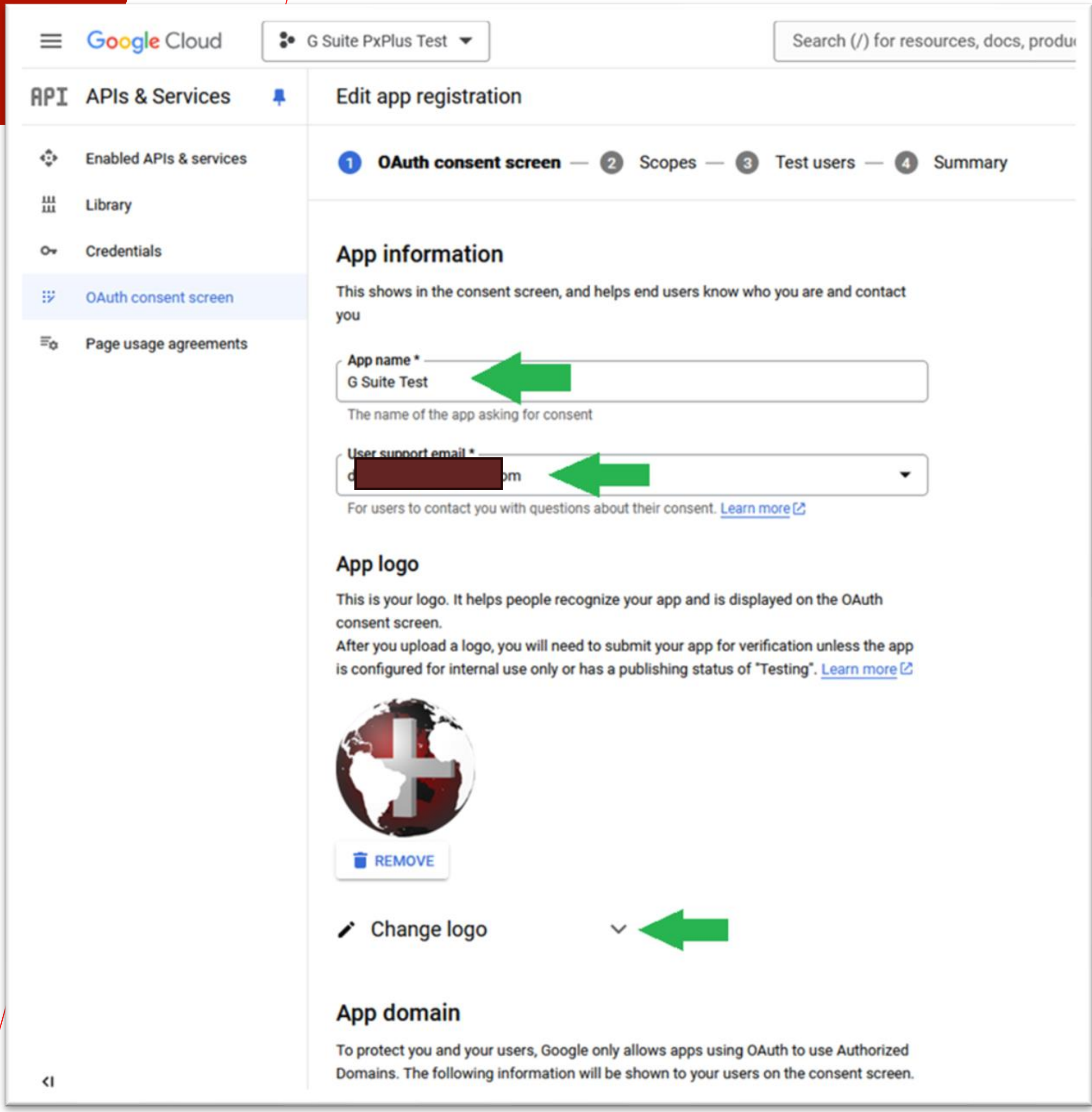
Only available to users within your organization. You will not need to submit your app for verification. Learn more about user type

● External

Available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app. Learn more about user type

CREATE

- **Create an OAuth consent screen**
  - Click OAuth consent screen
  - Select External and click CREATE
  - Set the Application name, Application logo, and Support e-mail that will display on your application consent screen



Edit app registration

1 OAuth consent screen — 2 Scopes — 3 Test users — 4 Summary

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *: G Suite Test

The name of the app asking for consent

User support email *: [redacted]

For users to contact you with questions about their consent. Learn more

App logo

This is your logo. It helps people recognize your app and is displayed on the OAuth consent screen.
After you upload a logo, you will need to submit your app for verification unless the app is configured for internal use only or has a publishing status of "Testing". Learn more

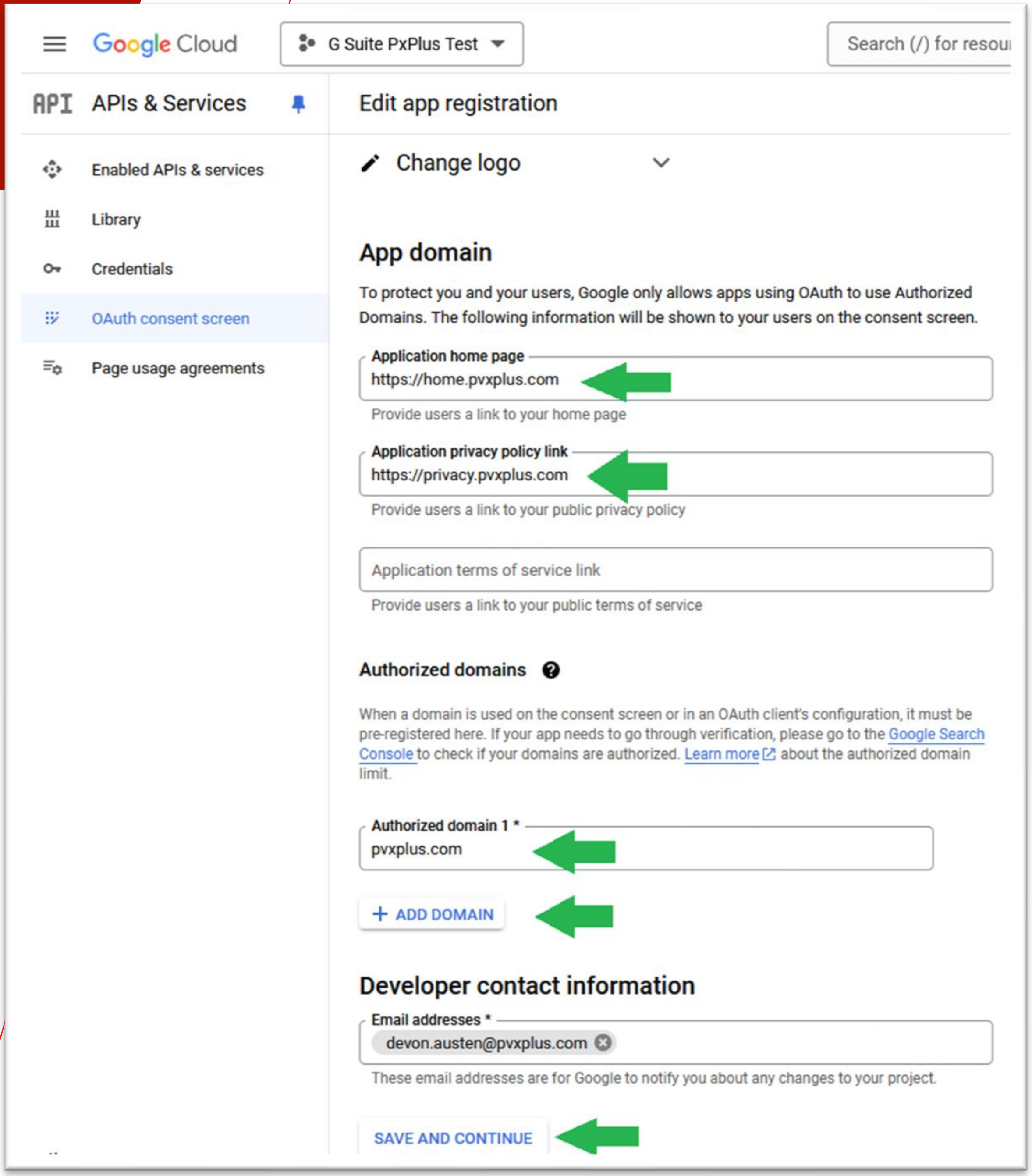
Change logo

App domain

To protect you and your users, Google only allows apps using OAuth to use Authorized Domains. The following information will be shown to your users on the consent screen.

GOOGLE WORKSPACE OBJECTS

- **Create an OAuth consent screen**
  - Set Application links
  - Input pvxplus.com for Authorized domains
  - Click SAVE AND CONTINUE



Edit app registration

App domain

To protect you and your users, Google only allows apps using OAuth to use Authorized Domains. The following information will be shown to your users on the consent screen.

Application home page: https://home.pvxplus.com
Provide users a link to your home page

Application privacy policy link: https://privacy.pvxplus.com
Provide users a link to your public privacy policy

Application terms of service link
Provide users a link to your public terms of service

Authorized domains

When a domain is used on the consent screen or in an OAuth client's configuration, it must be pre-registered here. If your app needs to go through verification, please go to the Google Search Console to check if your domains are authorized. Learn more about the authorized domain limit.

Authorized domain 1 *: pvxplus.com

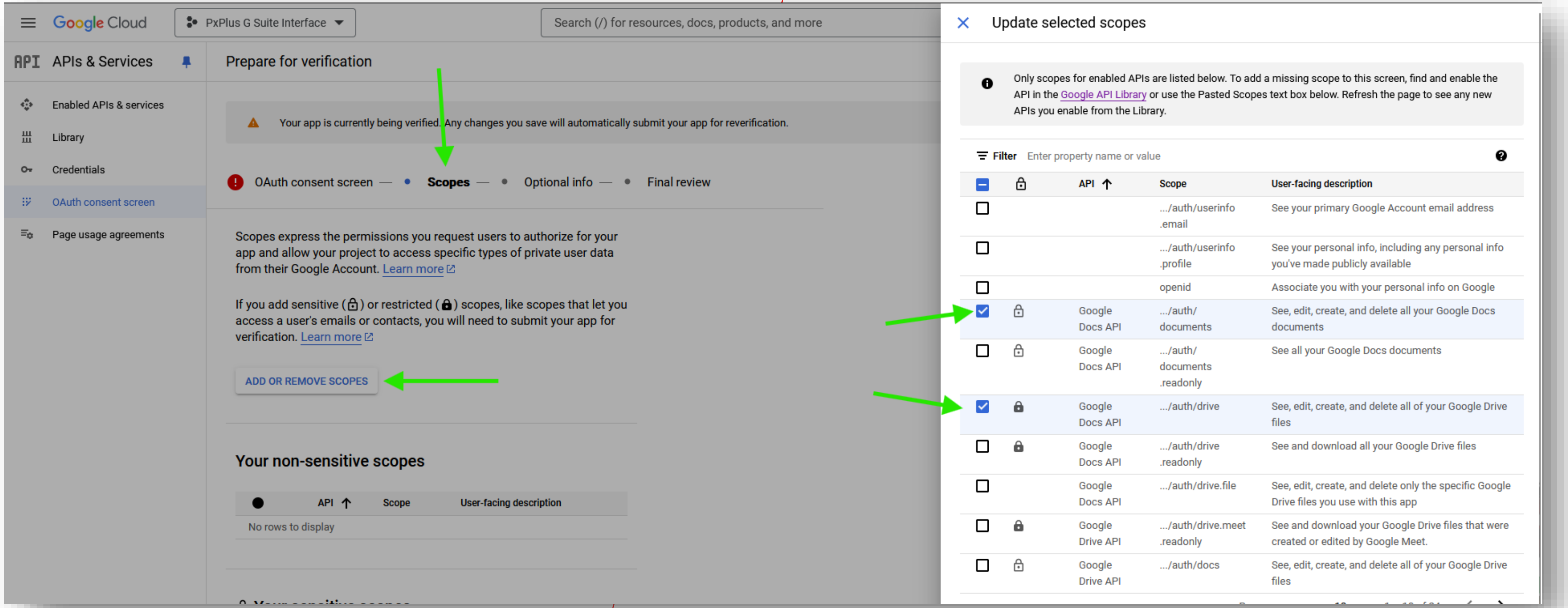
+ ADD DOMAIN

Developer contact information

Email addresses *: devon.austen@pvxplus.com
These email addresses are for Google to notify you about any changes to your project.

SAVE AND CONTINUE

GOOGLE WORKSPACE OBJECTS



Google Cloud | PxPlus G Suite Interface | Search (/) for resources, docs, products, and more

APIs & Services

- Enabled APIs & services
- Library
- Credentials
- OAuth consent screen
- Page usage agreements

Prepare for verification

Your app is currently being verified. Any changes you save will automatically submit your app for reverification.

OAuth consent screen — • **Scopes** — • Optional info — • Final review

Scopes express the permissions you request users to authorize for your app and allow your project to access specific types of private user data from their Google Account. Learn more

If you add sensitive (🔓) or restricted (🔒) scopes, like scopes that let you access a user's emails or contacts, you will need to submit your app for verification. Learn more

ADD OR REMOVE SCOPES

Your non-sensitive scopes

●	API ↑	Scope	User-facing description
No rows to display			

Update selected scopes

Only scopes for enabled APIs are listed below. To add a missing scope to this screen, find and enable the API in the Google API Library or use the Pasted Scopes text box below. Refresh the page to see any new APIs you enable from the Library.

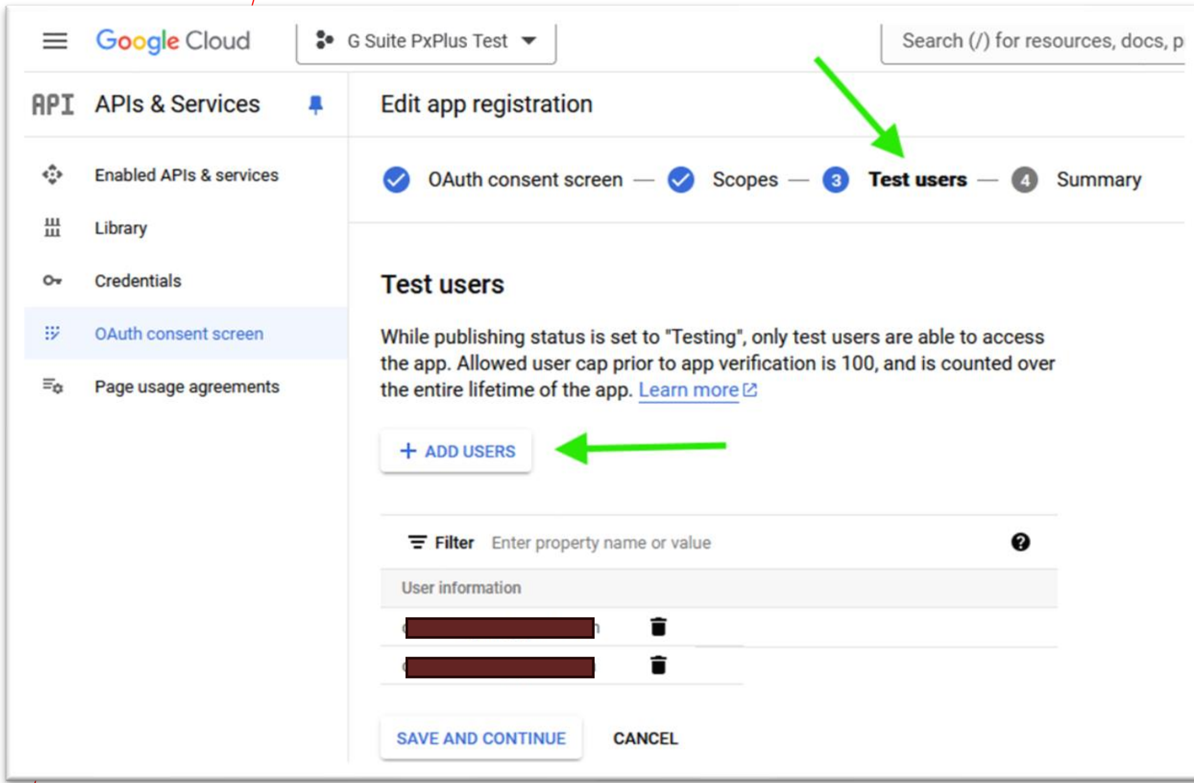
Filter Enter property name or value

☐	🔓	API ↑	Scope	User-facing description
☐			.../auth/userinfo .email	See your primary Google Account email address
☐			.../auth/userinfo .profile	See your personal info, including any personal info you've made publicly available
☐			openid	Associate you with your personal info on Google
☑	🔓	Google Docs API	.../auth/ documents	See, edit, create, and delete all your Google Docs documents
☐	🔓	Google Docs API	.../auth/ documents .readonly	See all your Google Docs documents
☑	🔒	Google Docs API	.../auth/drive	See, edit, create, and delete all of your Google Drive files
☐	🔒	Google Docs API	.../auth/drive .readonly	See and download all your Google Drive files
☐		Google Docs API	.../auth/drive.file	See, edit, create, and delete only the specific Google Drive files you use with this app
☐	🔒	Google Drive API	.../auth/drive.meet .readonly	See and download your Google Drive files that were created or edited by Google Meet.
☐	🔓	Google Drive API	.../auth/docs	See, edit, create, and delete all of your Google Drive files

- **Define scopes**
  - Click ADD OR REMOVE SCOPES
    - Select Google Drive API ../auth/drive
    - Select Google Docs API ../auth/documents
    - Select Google Sheets API ../auth/spreadsheets
  - Click UPDATE and then click SAVE AND CONTINUE

GOOGLE WORKSPACE OBJECTS

- Click ADD USERS
  - Only emails added here will be able to use this API until app is published
  - While unpublished, the test users will be warned the app is not verified
- Click SAVE AND CONTINUE



Test users

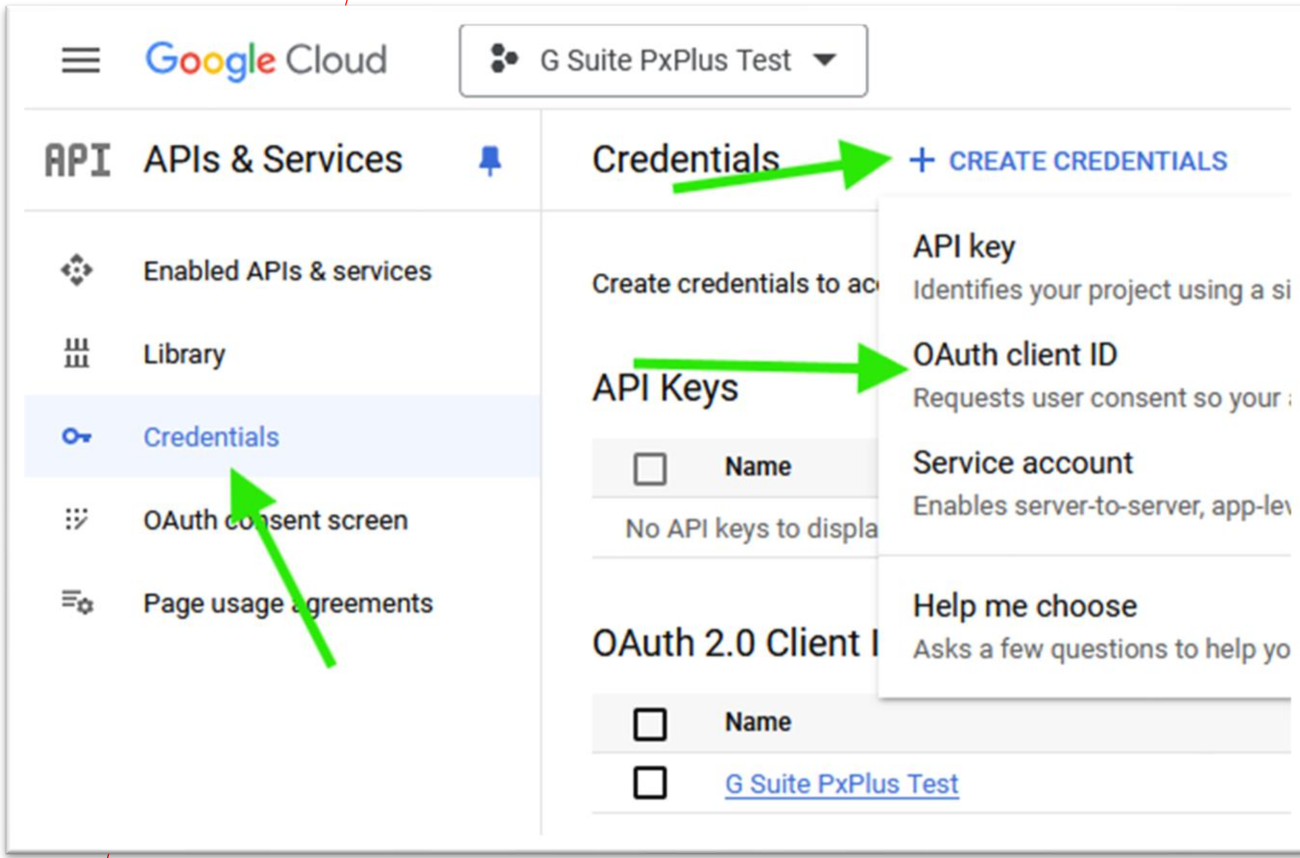
While publishing status is set to "Testing", only test users are able to access the app. Allowed user cap prior to app verification is 100, and is counted over the entire lifetime of the app. Learn more

[redacted]

[redacted]

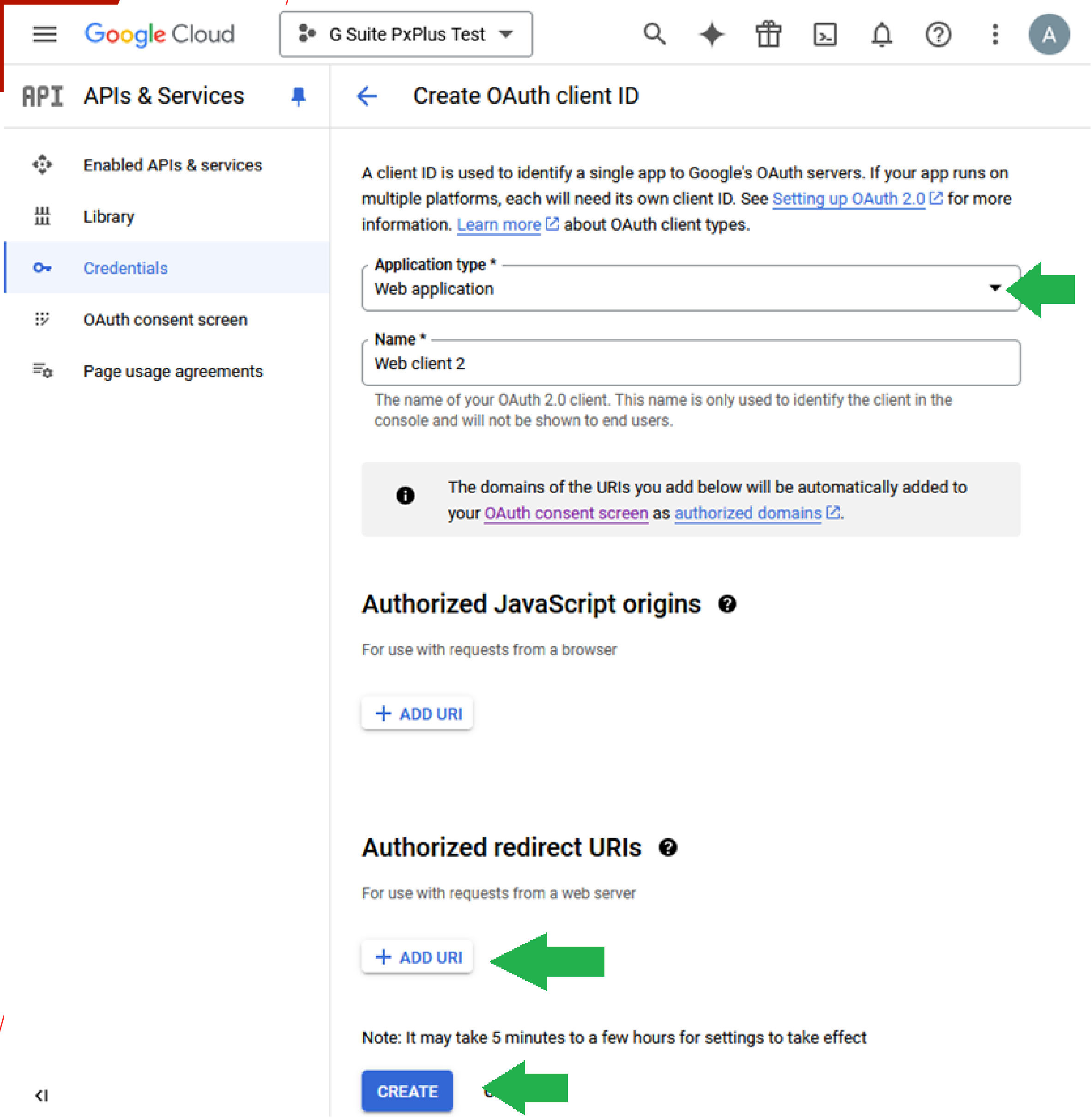
GOOGLE WORKSPACE OBJECTS

- **Create Client ID and Client Secret**
    - Click Credentials
    - Click Create credentials
    - Click OAuth client ID



GOOGLE WORKSPACE OBJECTS

- **Create Client ID and Client Secret**
  - For Application type, select Web application
  - Authorized redirect URIs
    - Click ADD URI
      - Add https://www.pvxplus.com/oauth.pvp
  - Click CREATE



Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 for more information. Learn more about OAuth client types.

Application type *: Web application

Name *: Web client 2

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorized domains.

Authorized JavaScript origins

For use with requests from a browser

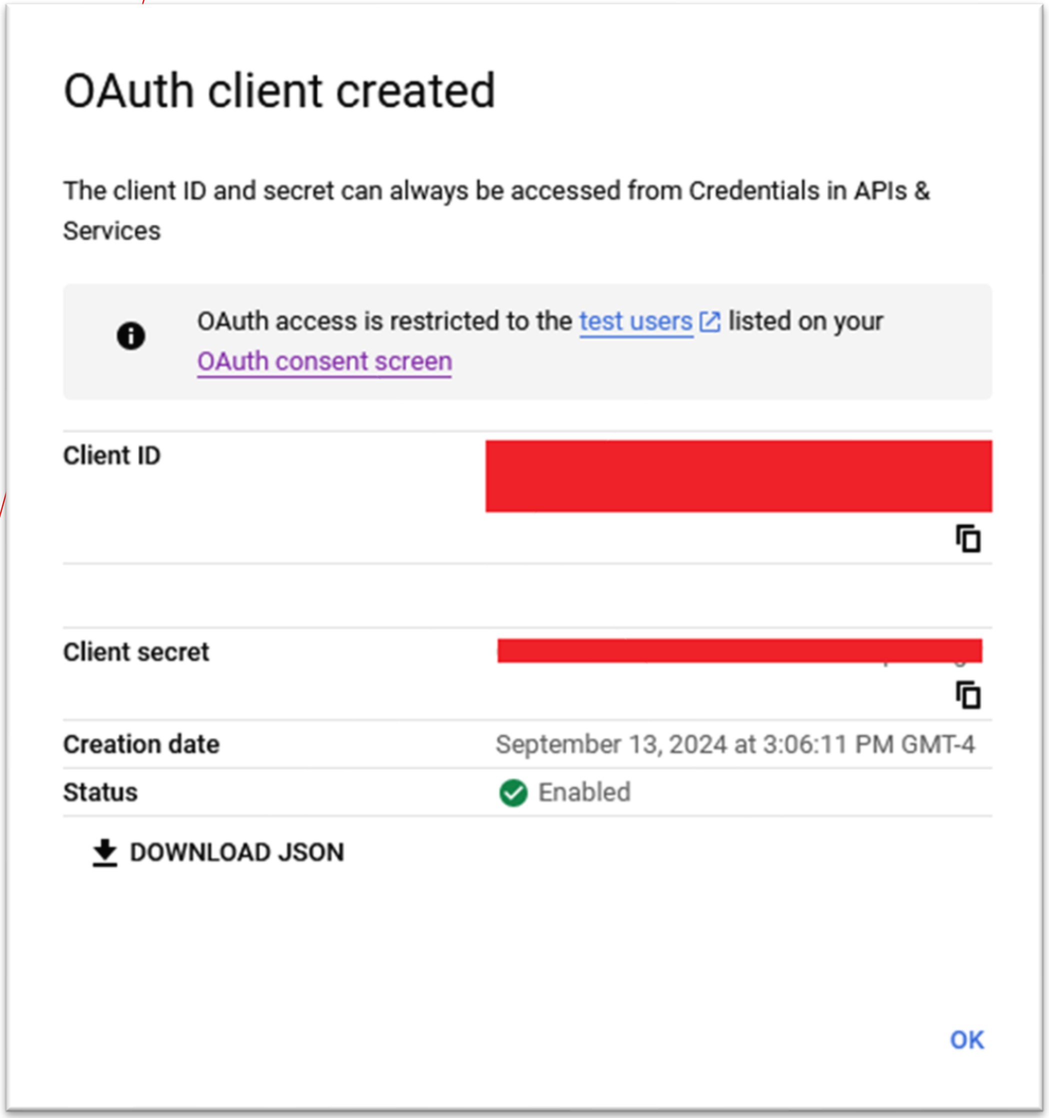
Authorized redirect URIs

For use with requests from a web server

Note: It may take 5 minutes to a few hours for settings to take effect

GOOGLE WORKSPACE OBJECTS

- **Create Client ID and Client Secret**
  - Copy/Save the client ID and client secret
  - These numbers are needed by the Google Workspace objects



OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

OAuth access is restricted to the test users listed on your OAuth consent screen

Client ID [redacted]

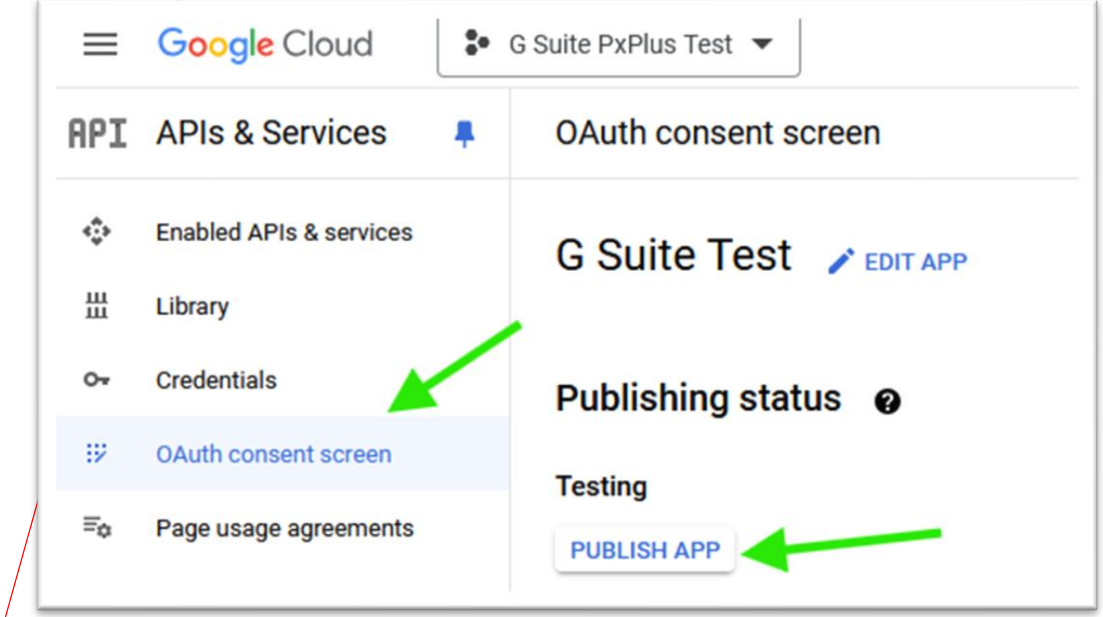
Client secret [redacted]

Creation date September 13, 2024 at 3:06:11 PM GMT-4

Status ✓ Enabled

GOOGLE WORKSPACE OBJECTS

- **When done testing, you can publish the app**
  - This will allow anyone to use the app
  - Removes the unverified warning



GOOGLE WORKSPACE OBJECTS

- To instantiate a Google Workspace object, include the Client ID and Client Secret

```
sheets_obj=new("*obj/GoogleSheets",client_ID$,client_secret$[,login_token$])
```

- During instantiation, user will be asked to select and login to a Google account and allow PxPlus access to it via a web browser
- When this process is completed, the **Login()** method must be run to complete login
- Once logged in, the **login_token$** property can be accessed and saved
  - This may optionally be used the next time the object is instantiated to avoid having to select a Google account and login again
- Methods and properties are all designed to be similar to the Microsoft Word and Excel objects

GOOGLE WORKSPACE OBJECTS

- **Files are stored in your Google Drive cloud storage, not as local files**
- **Google Drive supports directories and sub-directories**
- **All methods that support a path allow the Google Drive directories and the file name to be specified so that files can be organized more easily**
  - For example, to create a new document in the Google Drives directory "Template", the following method can be used:

    driveCreateDocument("Templates\invoice")
- **To work with a local file, upload it to Google Drive by using the UploadFile() method**
  - May convert the file into a Google file type such as a document or spreadsheet
    - List of supported conversions: https://developers.google.com/drive/api/v3/manage-uploads#import_to_google_docs_types
- **To download a Google file such as a document or spreadsheet, use the DownloadFile() or ExportFile() method**
  - Export converts the file into a local file type of file, determined by the file extension specified in the output path
    - List of supported conversions: https://developers.google.com/drive/api/v3/ref-export-formats

GOOGLE WORKSPACE OBJECTS

- **Most times the first thing you will need to do is open a file**
  - To open a file you use the Google Drive path or file ID in one of the open methods
    - **OpenDocumentByID(document_fileID$)** or **OpenDocumentByPath(document_path$)**
    - **OpenSpreadsheetByID(spreadsheet_fileID$)** or **OpenSpreadsheetByPath(spreadsheet_path$)**
    - **OpenFileByID(fileID$)** or **OpenFileByPath(file_path$)**
- **When a file is opened, the object assigns it an index**
  - The index start at 1 and goes up for each open file
  - The **ACTIVE_DOCUMENT, ACTIVE_SPREADSHEET,** and **ACTIVE_FILE** properties can be used to get the index of the last opened file
    - If no file is specified in a method call, the active index is used
  - THE **DOCUMENTS_COUNT, SPREADSHEETS_COUNT,** and **FILES_COUNT** properties can be used to find out how many opened files there are
    - You can use this to loop through each open file if the loop starts at 1 and goes to the count

GOOGLE WORKSPACE OBJECTS

For documents you can work with different

- **Paragraphs**
  - Paragraphs are referenced via a 1 based index number
    - Paragraph 1 is the 1st paragraph in the document
    - Paragraph 10 is the 10th paragraph in the document or appended to the end of the document if there is less than 10 paragraphs in the document
    - **docs'InsertParagraph("My new paragraph.",10)**
- **Sentences**
  - These work the same as paragraphs and are just a 1 based index
  - **docs'InsertSentence("My new sentence. ",130)**
- **Modify text at a searched for location**
  - **docs'FindInsert("something"," was changed")**
    - Defaults to inserting after the find position
    - Use the property **INSERT_BEFORE** to have it insert before the fond position

GOOGLE WORKSPACE OBJECTS

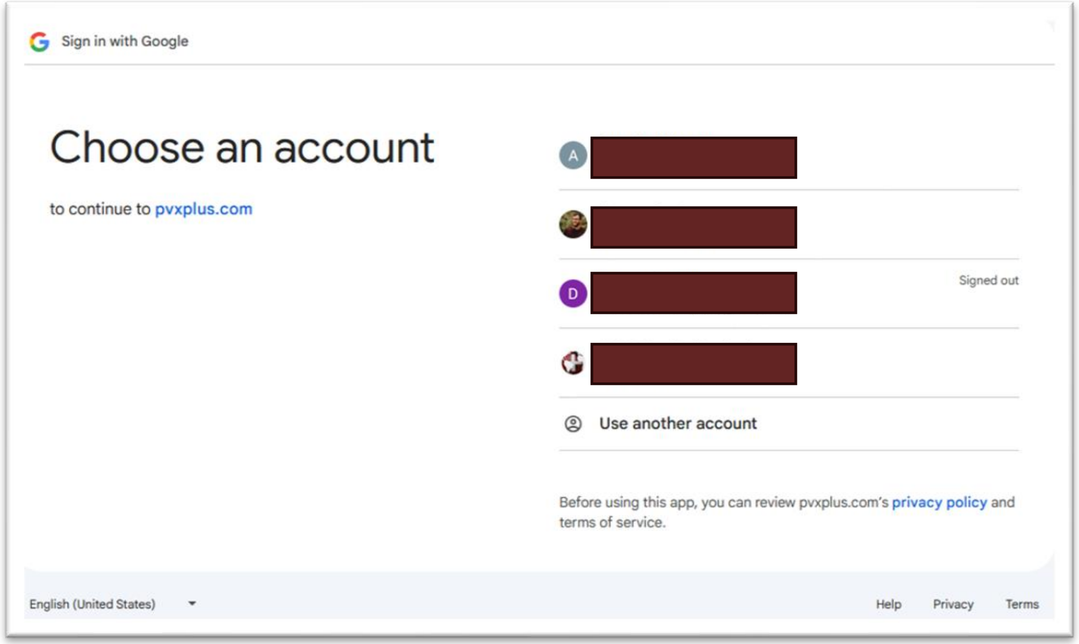
For spreadsheets you can work with different

- Sheets in the spreadsheet
  - **sheets'CreateSheet("Added Sheet")**
  - **sheets'SetSheet(3)** or **sheets'SetSheet("Added Sheet")**
  - Object keeps track of **ACTIVE_SHEET** which allows code to not specify a sheet defaulting to active sheet
- Ranges
  - **sheets'AddNamedRange("totals","A30:D30")**
  - **sheets'SetRange("A30:D30")** or **sheets'SetRange("totals")**
  - Object keeps track of **ACTIVE_RANGE$** which allows code to not specify a range defaulting to the active range
- Cells
  - Can be specified as a specific cell "A12" or a range or named range
- Find/replace
  - **sheets'FindReplaceAll("[CompanyName]",companyName$)**

GOOGLE WORKSPACE OBJECTS

```
clientID$="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
clientSecret$="xxxxxxxxxxxxxxxx"
gd=new("*obj/GoogleDocs",clientID$,clientSecret$)
input "Press any key to continue after logging into Google account and allowing PxPlus access:",*;print ""
gd'Login()
!
if gd'OpenDocumentByPath("myGoogleDoc")=0 then exit 15
gd'FindReplaceAllByPath("[FirstName]", "Jack" ,"all")
gd'FindSetFont("New discount","Algerian,18,bold")
gd'FindSetColor("25% off coupon","red")
gd'ExportDocument(path$+clientID$+".docx")
gd'CloseDocument(1)
!
drop object gd
```

GOOGLE WORKSPACE OBJECTS



Sign in with Google

Choose an account

to continue to pvxplus.com

A [redacted]

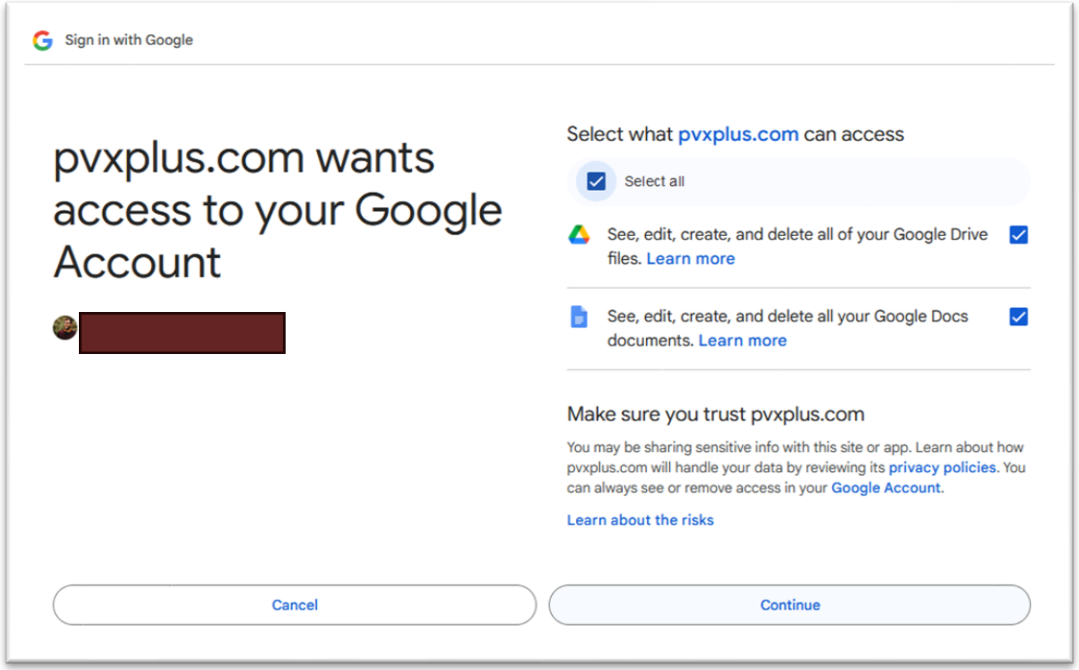
[redacted]

D [redacted] Signed out

[redacted]

Use another account

Before using this app, you can review pvxplus.com's privacy policy and terms of service.



Sign in with Google

pvxplus.com wants access to your Google Account

[redacted]

Select what pvxplus.com can access

☑ Select all

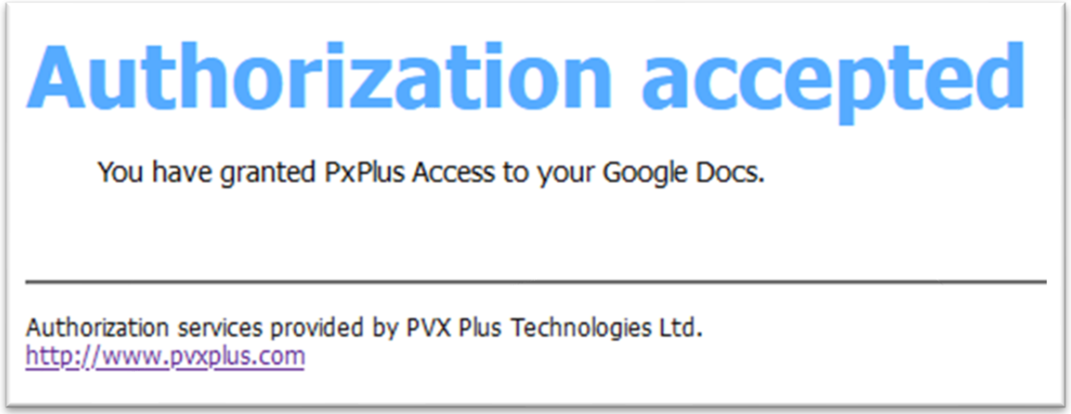
See, edit, create, and delete all of your Google Drive files. Learn more ☑

See, edit, create, and delete all your Google Docs documents. Learn more ☑

Make sure you trust pvxplus.com

You may be sharing sensitive info with this site or app. Learn about how pvxplus.com will handle your data by reviewing its privacy policies. You can always see or remove access in your Google Account.

Learn about the risks



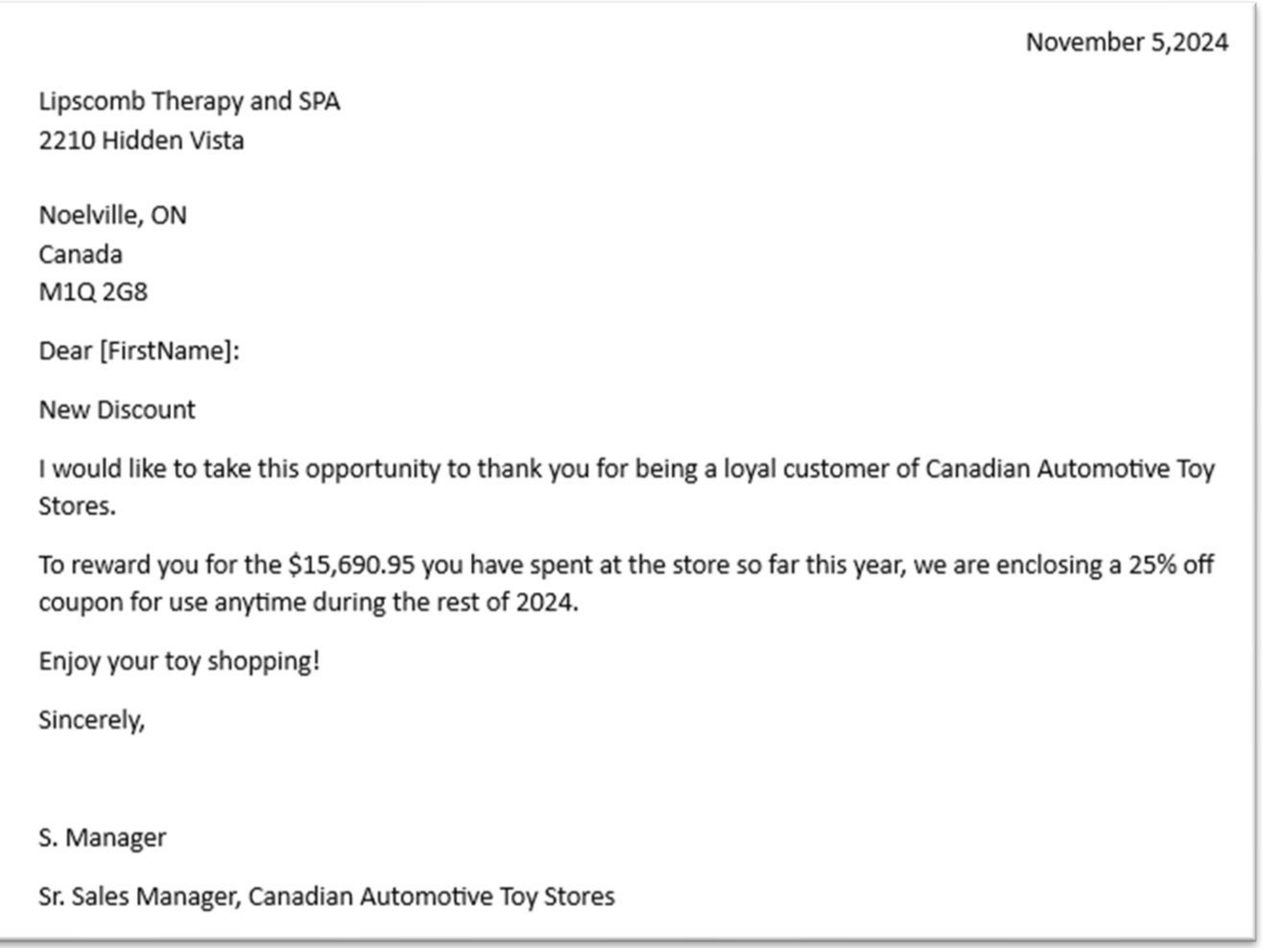
Authorization accepted

You have granted PxPlus Access to your Google Docs.

Authorization services provided by PVX Plus Technologies Ltd.
http://www.pvxplus.com

GOOGLE WORKSPACE OBJECTS

Before



November 5,2024

Lipscomb Therapy and SPA
2210 Hidden Vista

Noelville, ON
Canada
M1Q 2G8

Dear [FirstName]:

New Discount

I would like to take this opportunity to thank you for being a loyal customer of Canadian Automotive Toy Stores.

To reward you for the $15,690.95 you have spent at the store so far this year, we are enclosing a 25% off coupon for use anytime during the rest of 2024.

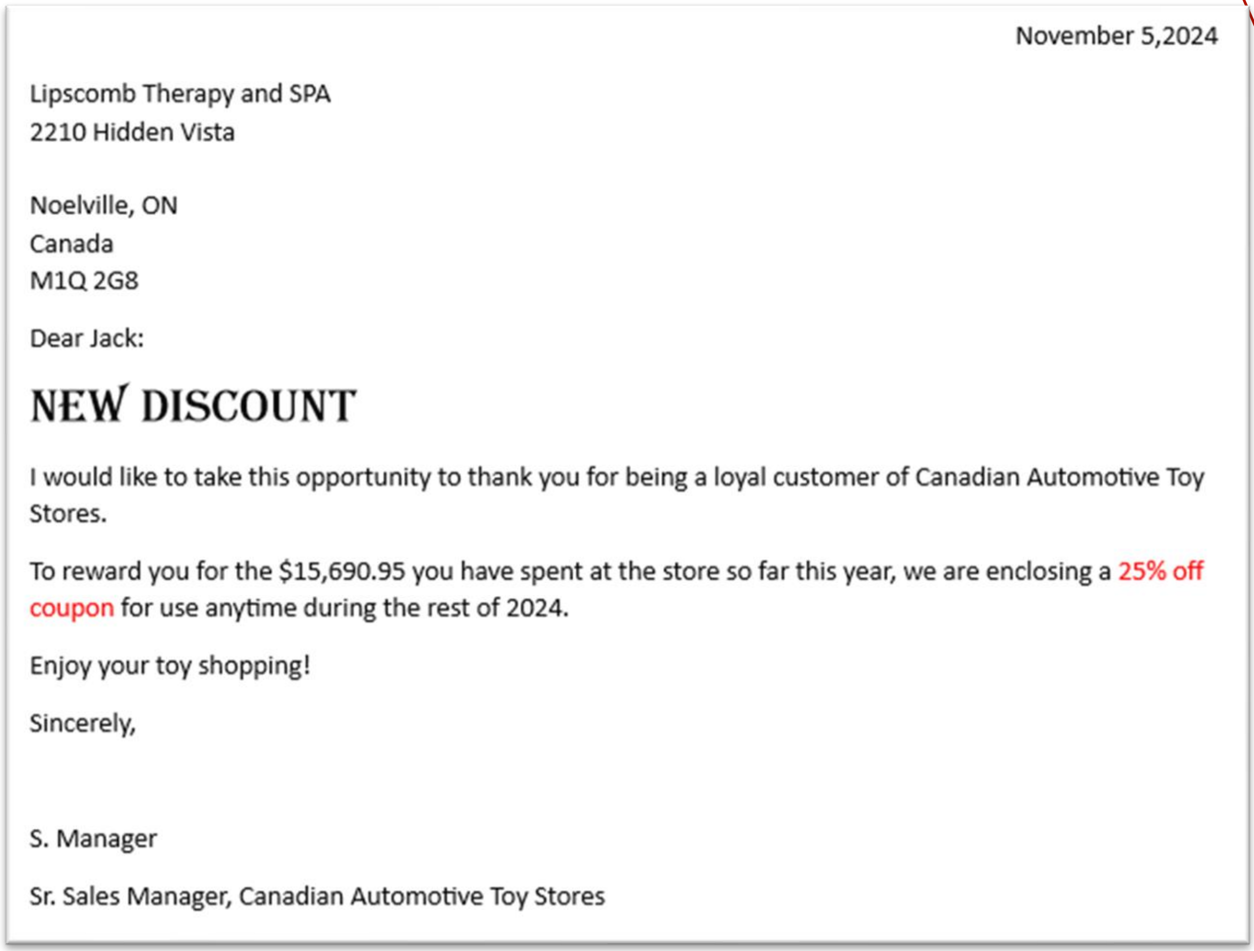
Enjoy your toy shopping!

Sincerely,

S. Manager

Sr. Sales Manager, Canadian Automotive Toy Stores

After



November 5,2024

Lipscomb Therapy and SPA
2210 Hidden Vista

Noelville, ON
Canada
M1Q 2G8

Dear Jack:

NEW DISCOUNT

I would like to take this opportunity to thank you for being a loyal customer of Canadian Automotive Toy Stores.

To reward you for the $15,690.95 you have spent at the store so far this year, we are enclosing a 25% off coupon for use anytime during the rest of 2024.

Enjoy your toy shopping!

Sincerely,

S. Manager

Sr. Sales Manager, Canadian Automotive Toy Stores

WORKING WITH WEB SERVICES

WORKING WITH WEB SERVICES

- Examples of Web Services
  - Check the price of an item
  - Place an order for a product
  - Update contact information
  - Check the weather
- A Web Service exchanges information between applications using the standard Web protocol
  - Usually the data is in the form of JSON or XML
- Web requests have two main components
  - **Header** - Describes the message
    - The header consists of simple concise text lines of Keyword: Value and primarily defines content size, type, source and destination
  - **Body** - Contains the message

WORKING WITH WEB SERVICES

- The processing of Web requests begins with a URL
  - http://target-system-name/target…?parameters
- Two basic types of requests
  - **GET** - Requests the contents of a target from the server
  - **POST** - Sends some data (i.e. a form) to a specific target on the server

WORKING WITH WEB SERVICES

- PxPlus provides a utility for submitting a request with data and returning the result

  "*plus/web/request"
- Supports secure (**https:**//) and unsecure (**http:**//) requests
- Uses HTTP version 1.1 protocol
  - Can force HTTP version 1.0 protocol by defining the global variable %**web_request_http_ver$** as "1.0"
- By default, the user agent will be the same as if the request came from a Windows version of Google Chrome with the version of PxPlus appended to it
  - Can be overridden by including a user agent header in the **extrahdrs$** argument

    extrahdr$="User-Agent: MyApp/1.00.0000"
- Default request is a **GET** if no post data is provided; otherwise, a **POST** is used
- Eight entry points for additional request types are available: **GET, POST, DELETE, PUT, TRACE, OPTIONS, PATCH** and **HEAD**
- This utility also allows you to specify the content type, SSL certificate and additional headers

WORKING WITH WEB SERVICES

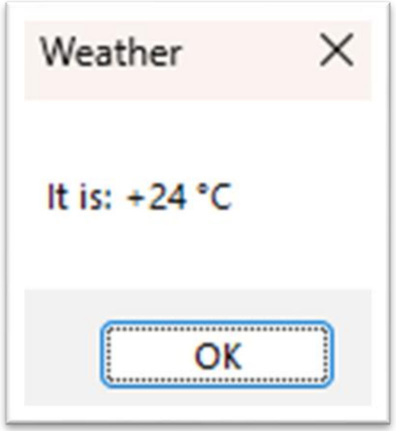
```
CALL "*plus/web/request", url$, postdata$, recvdata$, recvhdr$, mimetype$, certificate$, extrahdrs$, method$, timeout
```

Argument	Description
url$	Specifies target server and if secure or non-secure
postdata$	Data to send to the server
recvdata$	String variable to receive response
recvhdr$	Header of response received
mimetype$	Optional content type of data
certificate$	Optional path of certificate, if needed
extrahdrs$	Optional additional headers to include on send
method$	HTTP request method
timeout	Timeout value (if not provided, defaults to 15 seconds)

WORKING WITH WEB SERVICES

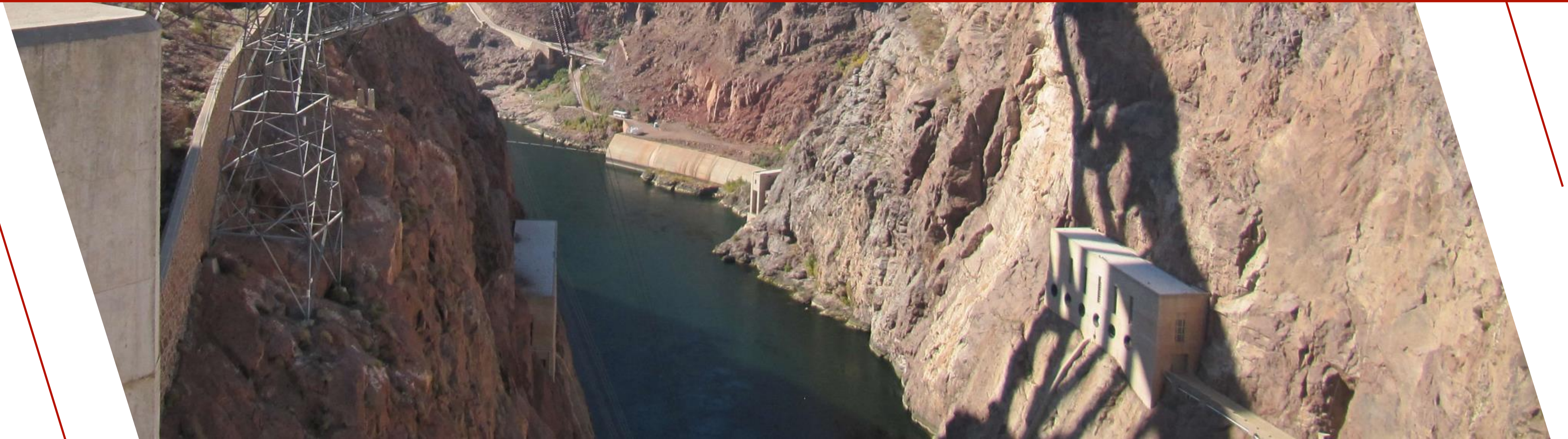
Example - Web request getting current weather for Toronto

```
CALL "*plus/web/request","https://goweather.herokuapp.com/weather/toronto","", Resp$,Hdr$,"","","",30
dim load forecast${all}=resp$
msgbox "It is: "+forecast$["temperature"],"Weather"
```



It is: +24 °C

PAYPAL AND SALESFORCE EXAMPLES



PAYPAL AND SALESFORCE EXAMPLES

- Example objects are included to provide help with developing your own interfaces to REST-based Web services
  - **PayPal® (*obj/example/paypal/api)**
  - **Salesforce® (*obj/example/salesforce/api)**
- The example code can be used as a reference, or it can be copied and tailored to your specific needs
- **Important Note:**
  - These examples are provided only to serve as a guide when developing your own REST interfaces and are not supported by PxPlus since the PayPal® and Salesforce® APIs may be subject to change without notice

PAYPAL AND SALESFORCE EXAMPLES

PayPal® Example Object

- Handles authentication with PayPal® and allows you to send and receive money using PayPal®
  - This object currently points to the PayPal® sandbox API for testing but can be easily modified to point to the live PayPal® API once testing is done
- Before you can interface with PayPal® from PxPlus, a Client ID and Client Secret must be obtained from PayPal®
  - To do this, first create a PayPal® developer account and then follow the instructions provided at this Web site: **https://developer.paypal.com/docs/api-basics/manage-apps/**
- The PayPal® object runs an EZWeb server on the port specified in the constructor
  - The EZWeb server is used for handling PayPal® order payment call-backs
  - If an order payment is completed, the HTML page ***obj/example/paypal/paycomplete.htm** will be loaded
  - If the order payment is cancelled, the HTML page ***obj/example/paypal/paycancel.htm** will be loaded
  - Currently, the callback HTML pages are simply text stating whether the payment was completed or cancelled

PAYPAL AND SALESFORCE EXAMPLES

PayPal® Example Object

- To instantiate the PayPal® object using the handle **paypal** (where **paypal** could be any numeric variable), enter the following command:

  ```
  paypal=NEW("*obj/example/paypal/api", clientID$, clientSecret$, ezPort)
  ```

- No need to login to a PayPal® account on object instantiation

PAYPAL AND SALESFORCE EXAMPLES

Items Object

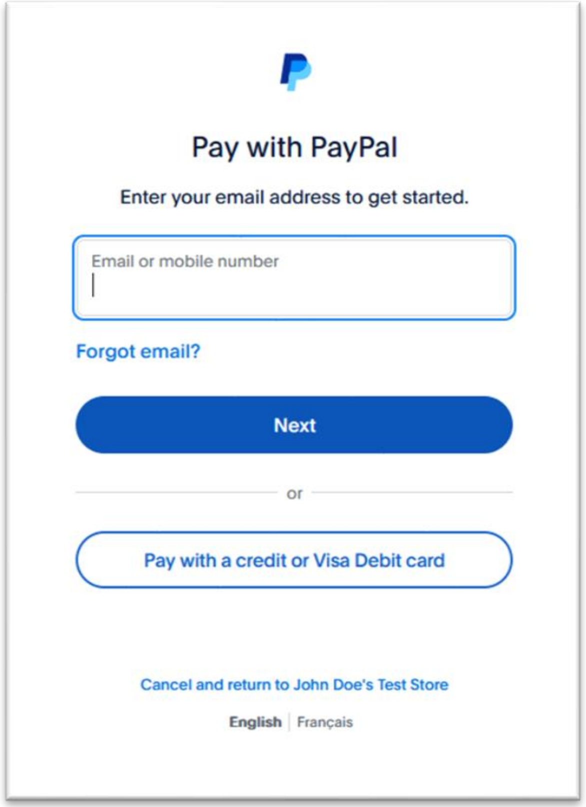
- PxPlus has also provided the Items object used by the **PayPal** object
- Purpose of the Items object is to create orders and invoices
  - Use this object to add all the items to the order/invoice and set the tax and shipping cost
  - It will automatically calculate the sub-total and total
- To instantiate the Items object using the handle **items** (where **items** could be any numeric variable), enter the following command:

```
items=NEW("*obj/example/paypal/items")
```

PAYPAL AND SALESFORCE EXAMPLES

```
clientID$="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
clientSecret$="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
listenPort=8080
paypal=new("*obj/example/paypal/api",clientID$,clientSecret$,listenPort)
items=new("*obj/example/paypal/items")
items'TaxPercentage=13
items'ShippingCost=11
items'AddItem("CPU",459.49)
items'AddItem("Memory",112.99)
orderId$=paypal'NewOrder$(items)
input "Press any key to continue after paying the order",*;print ""
 if paypal'CaptureOrder(orderId$)=1 \
          then print "Captured payment" \
          else msgbox "Failed to capture payment","Error"
drop object items
drop object paypal
```

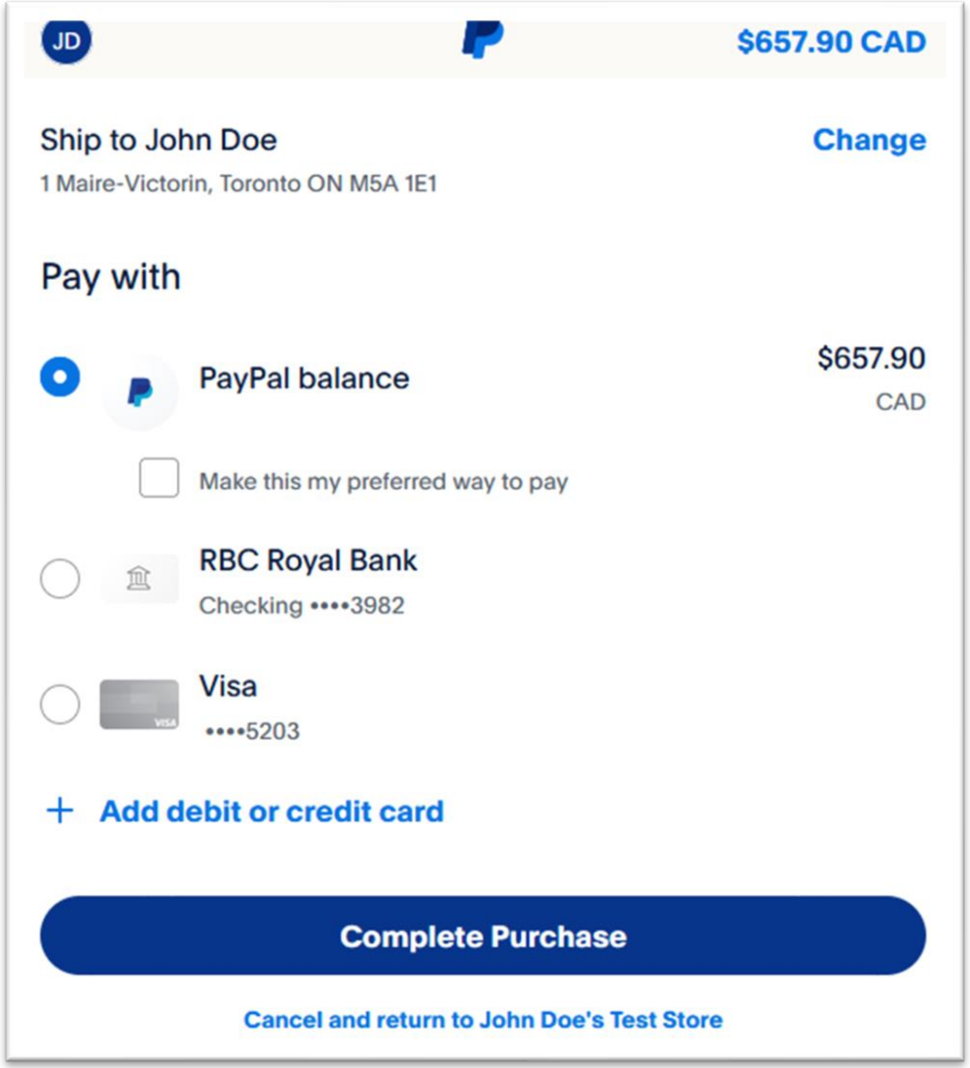
PAYPAL AND SALESFORCE EXAMPLES



Pay with PayPal

Enter your email address to get started.

Email or mobile number
Forgot email?
Next
or
Pay with a credit or Visa Debit card
Cancel and return to John Doe's Test Store
English | Français



JD $657.90 CAD

Ship to John Doe

1 Maire-Victorin, Toronto ON M5A 1E1

Pay with

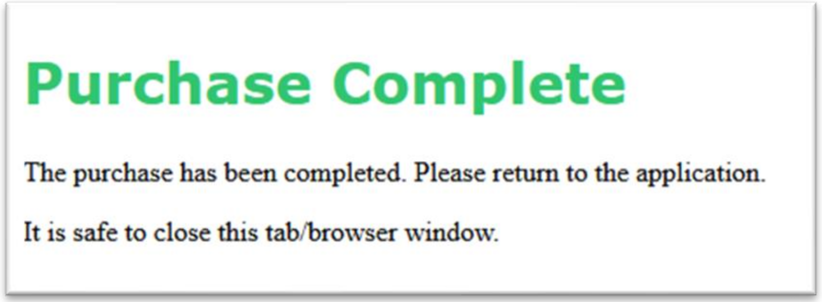
● PayPal balance $657.90 CAD

□ Make this my preferred way to pay

○ RBC Royal Bank Checking ••••3982

○ Visa ••••5203

Change
+ Add debit or credit card
Complete Purchase
Cancel and return to John Doe's Test Store



Purchase Complete

The purchase has been completed. Please return to the application.

It is safe to close this tab/browser window.

PAYPAL AND SALESFORCE EXAMPLES

Salesforce® Example Object

- Handles authentication with Salesforce® and allows you to work with Salesforce® Accounts and Opportunities, as well as perform a Salesforce® query
- Before you can interface with Salesforce® from PxPlus, a Client ID and Client Secret must be obtained from Salesforce®
  - To do this, first create a Salesforce® developer account, and then follow the instructions provided at both of these Web sites:

    https://help.salesforce.com/articleView?id=connected_app_create_basics.htm&type=5

    https://help.salesforce.com/articleView?id=connected_app_create_api_integration.htm&type=5

PAYPAL AND SALESFORCE EXAMPLES

Salesforce® Example Object

- To instantiate the Salesforce® object using the handle **salesforce** (where **salesforce** could be any numeric variable), enter the following command:

  Salesforce=NEW("*obj/example/salesforce/api", clientId$, clientSecret$[, refreshToken$])

- During object instantiation, the user is asked to select and login to a Salesforce® account via the default Web browser
  - When this process is completed, the **Login()** method must be run to complete the login to Salesforce®
- The Refresh token can optionally be used to avoid having to select a Salesforce® account and login more than once
  - Once logged in to Salesforce®, the **RefreshToken$** property can be accessed
  - The next time the Salesforce® object is instantiated, just include the saved Refresh token, and it will be logged into the same Salesforce® account automatically
- The login lasts for 15 minutes by default. After that, you need to call **Login()** again; otherwise, the methods will fail

PAYPAL AND SALESFORCE EXAMPLES

Account Object

- Used to create, read and update Salesforce® account
- Only includes a subset of the account fields and can be modified by adding or removing fields as needed for your own application (Required field: Name$)
- To instantiate the Account object using the handle **account** (where **account** could be any numeric variable), enter the following command:

```
account=NEW("*obj/example/salesforce/account")
```

PAYPAL AND SALESFORCE EXAMPLES

Opportunity Object

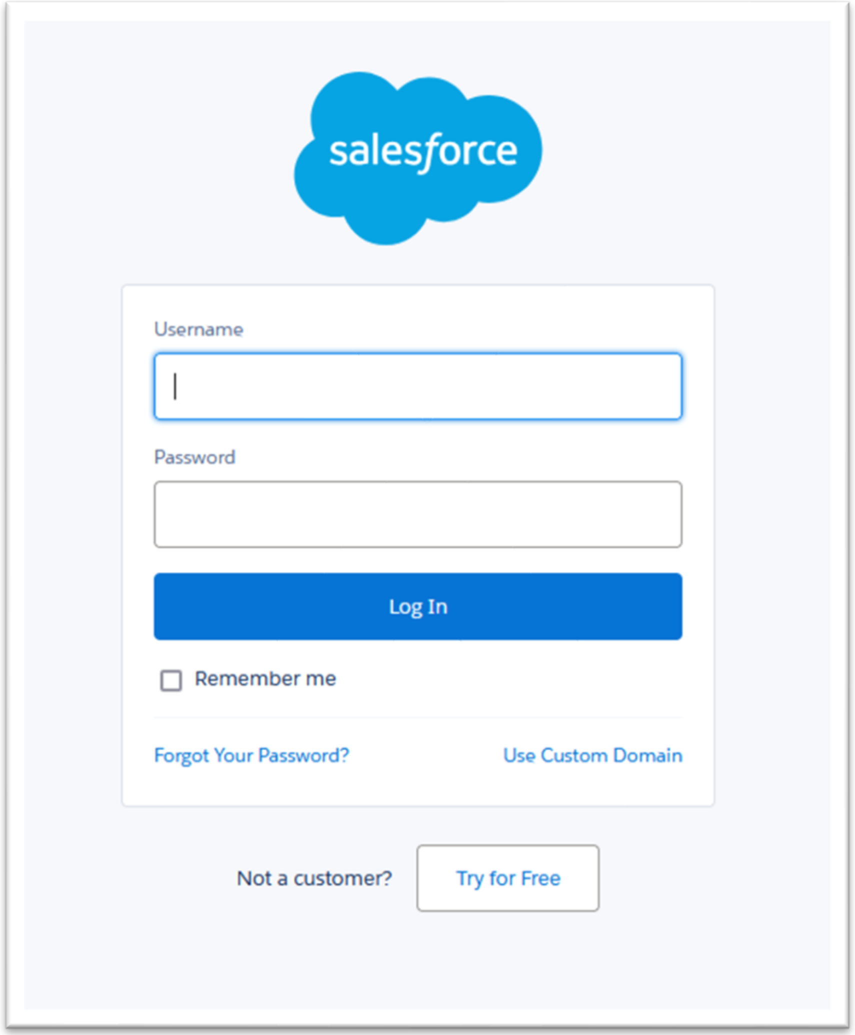
- Used to create, read and update Salesforce® opportunities
- Only includes a subset of the opportunity fields and can be modified by adding or removing fields as needed for your own application (Required fields: CloseDate$, Name$, StageName$)
- To instantiate the Opportunity object using the handle **opportunity** (where **opportunity** could be any numeric variable), enter the following command:

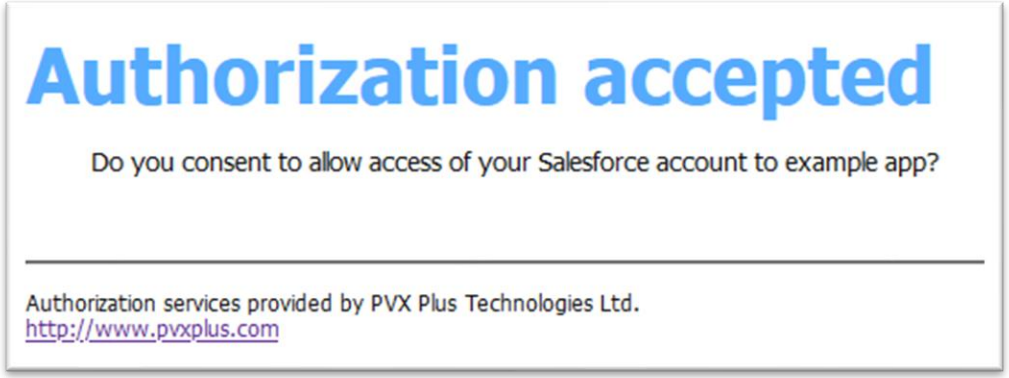
```
opportunity=NEW("*obj/example/salesforce/opportunity")
```

PAYPAL AND SALESFORCE EXAMPLES

```
clientID$="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
clientSecret$="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
salesforce=new("*obj/example/salesforce/api",clientID$,clientSecret$)
input "Press any key to continue after logging into Salesforce account and allowing PxPlus access:",*;print ""
salesforce'Login()
newOpportunity=new("*obj/example/salesforce/opportunity")
newOpportunity'Name$="Example Corp Server Upgrades"
newOpportunity'AccountId$="0014x00000CMnEIAAL"
newOpportunity'Description$="They need to buy 50 new servers to upgrade old ones."
newOpportunity'Amount=30000
newOpportunity'CloseDate$="2024-12-30"
newOpportunity'StageName$="Prospecting"
newOpportunityId$=salesforce'NewOpportunity$(newOpportunity)
drop object newOpportunity
drop object salesforce
```

PAYPAL AND SALESFORCE EXAMPLES

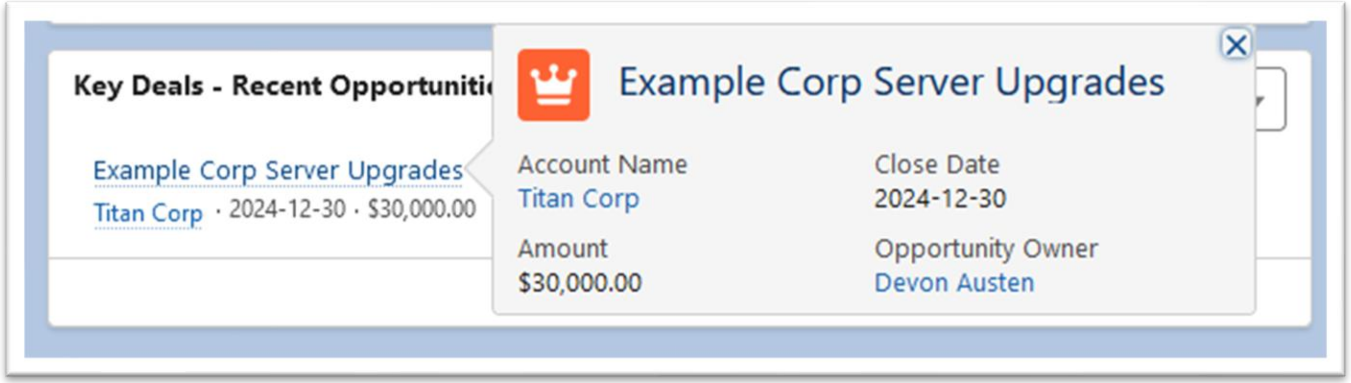




Authorization accepted

Do you consent to allow access of your Salesforce account to example app?

Authorization services provided by PVX Plus Technologies Ltd.
http://www.pvxplus.com



Key Deals - Recent Opportuniti

Example Corp Server Upgrades
Titan Corp · 2024-12-30 · $30,000.00

Example Corp Server Upgrades

Account Name
Titan Corp

Close Date
2024-12-30

Amount
$30,000.00

Opportunity Owner
Devon Austen