Microsoft OAuth 2.0 Quick Start Guide

This guide will show you how to obtain OAuth 2.0 bearer tokens (also called access tokens) for use with Microsoft 365 mail accounts. To get started, you will need the following:

- 1. A Microsoft Azure account which you will need to register your application. You can create this for free at https://azure.microsoft.com/en-us/free/
- 2. A Microsoft 365 account which you can use for testing purposes. You can create a free mail account at https://outlook.live.com/
- 3. A copy of the Postman utility which can be downloaded for free from <u>https://www.postman.com</u>. We will use this to compose the queries to Microsoft's servers to obtain the authorization code, bearer, and refresh tokens.
- 4. SocketTools 10 Build 1245 which supports authentication using OAuth 2.0 bearer tokens. If you have a current development license, this is a free update. If you have an older version of SocketTools, you will need to upgrade to the current version.

If you already have a Microsoft account, but do not have an Azure account, you can create the Azure account using your current Microsoft account. However, it's not required they be linked.

Once you have created your Microsoft Azure and Microsoft 365 accounts, the first step is to use your browser to open the Azure portal at https://azure.microsoft.com/ and login and select **Portal** from the menu.

	Azure Active Directory				\times	⊵	P	()	<u>č</u>	? 🤅)
Azure services Services		See all						res			
Subscriptions	Azure Lighthouse Azure Migrate Azure OSS Resources No results were found.		Revoke user access in an emergence Resource Groups No results w		tory	1					
Microsoft Lear Learn Azure wit training from M			Secure your apps and infrastructure	Analyze an cloud spen							
Useful links				Azure mobile	арр		_				
Technical Documentatio Azure Migration Tools	n 🗗 Azure Services 🖓 Find an Azure expert		nt Azure Updates ⊡" kstart Center	App Store		Get IT ON Google	Play				

In the search bar, enter Azure Active Directory and select the search result.

Registering Your Application

Home >				
Default Directory Azure Active Directory «		_		
	🕗 Switch tenant 🔳 Delete tenant 🕂 C	create a tenant 🛛 What's new 🛛 💀 Preview	r features 🛛 🛇 Got feedback?	
Overview				
🚀 Getting started		remote work for your employees and partners. Learn m	2078	
Preview hub	Azare Active Directory can help you chable i	remote work for your employees and particles. ceaning	lore	
X Diagnose and solve problems	Default Directory			
Manage	Search your tenant			
🚨 Users				
🎎 Groups	Tenant information	💝 Azure AD Connect		
📫 External Identities	Your role	Status		
🎝 Roles and administrators	Global administrator More info	Not enabled		
Administrative units (Preview)	License Azure AD Free	Last sync		
Enterprise applications	Tenant ID	Sync has never run		
- Devices	surrenter title ware soler recation			
👪 App registrations	Primary domain			
Identity Governance	mildladdor comproved com			
Application proxy				
🔓 Licenses	Sign-ins			
🔶 Azure AD Connect	2 1.5			
됟 Custom domain names	0.5	•		
Mobility (MDM and MAM)	0	1		
🕈 Password reset	Jul 26 Sign ins	Aug 2	Aug 9	
Company branding	1			

You will be in the Default Directory Overview. Select **App registrations** from the left sidebar. This is how you will register your application with Microsoft and is required to obtain your client ID.

App registrations 🛷
+ New registration 🔀 Endpoints 🖉 Troubleshooting 🞍 Download (Preview) 🛛 🛇 Got feedback?
\bigcirc Welcome to the new and improved App registrations (now Generally Available). See what's new and learn more on how it's changed. \rightarrow
Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more
All applications Owned applications Applications from personal account
₽ Start typing a name or Application ID to filter these results
This account isn't listed as an owner of any applications in this directory. View all applications in the directory
View all applications from personal account

Select **New registration** from the menu at the top. This will begin the process of registering a new application associated with your Azure account.

Home > Default Directory | App registrations >

Register an application

* Name
The user-facing display name for this application (this can be changed later).
My Test Application 🗸
Supported account types
Who can use this application or access this API?
O Accounts in this organizational directory only (Default Directory only - Single tenant)
O Accounts in any organizational directory (Any Azure AD directory - Multitenant)
Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
O Personal Microsoft accounts only
Help me choose
Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.
Public client/native (mobile) http://127.0.0.1/testApp/

You can choose whatever name you wish for your application. This will be displayed to the user and can be changed later. For this guide, we will use "My Test Application" for the name.

For the supported account types, select **Accounts in any organizational directory**. For the redirect URI, select **Public client/native** from the dropdown list, and use a local loopback IP address for the redirect URI. This tells Microsoft you're developing a native desktop application, and how to provide you with the authorization code needed to access the mail account.

You should be sure to use a unique address for the redirect URI. We recommend following it with an abbreviated name for your application. It needs to be a valid URI and you should keep it short and concise. For this guide, we will use the redirect URI http://127.0.0.1/testApp/

You may wonder if using a non-secure connection is safe. Because 127.0.0.1 is the local loopback IP address for your device, a connection will never be attempted beyond your desktop system. To prevent possible conflicts with misconfigured firewalls, we recommend using the IP address and not http://localhost/testApp

Once you have completed the registration of your application, Microsoft will provide you with your client ID. This is the first critical piece of information you'll need to obtain OAuth 2.0 bearer tokens.

Home > Default Directory | App registrations >

S My Test Application | Authentication 🖉

9 7 11	
	🔚 Save 🗙 Discard 🛇 Got feedback?
Overview	Platform configurations
🗳 Quickstart	
🚀 Integration assistant (preview)	Depending on the platform or device this application is targeting, additional configuration may be required such as redirect URIs, specific authentication settings, or fields specific to the platform.
Manage	+ Add a platform
🔤 Branding	
Authentication	^ Mobile and desktop applications Quickstart Docs ♂
📍 Certificates & secrets	Redirect URIs
Token configuration	The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating users. Also referred to as reply URLs. Learn
->- API permissions	more about Redirect URIs and their restrictions 🖉
🙆 Expose an API	https://login.microsoftonline.com/common/oauth2/nativeclient
Owners	https://login.live.com/coulth20_desktop.srf (LiveSDK)
Roles and administrators (Preview)	msal5729d13a-7464-4e04-b93f-b218c981c02d://auth (MSAL only)
0 Manifest	http://127.0.0.1/testApp/
Support + Troubleshooting	Add URI
<i>Troubleshooting</i>	
New support request	Supported account types
	Who can use this application or access this API?
	Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
	All users with a work or school, or personal Microsoft account can use your application or API. This includes Office 365 subscribers.

Although you can always return to this screen to get your ID, we recommend you copy down your client ID and keep in a safe place. With Microsoft, the client IDs are UUIDs and are unique to every application registered. The other IDs are not needed for our purposes.

Select **Authentication** on the left sidebar and this will display more information about your application. You can use this to confirm that the redirect URI and supported account types are correct.

Now that we've registered our application and have the client ID, the next step is to use Postman to create the queries we will send to Microsoft's servers to obtain the authorization code. And with that authorization code, we can request the bearer (access) token and refresh token which are used to access the Microsoft 365 mail account.

Creating Postman Collections

🢋 Postman				- C	0	×
File Edit View Help						
🕂 New Import Runner 📭 🔻	🚼 My Workspace 👻 🗼 Invite	<u>o</u> s	▲ ♥ (Upgra	ade	•
Q Filter	+	No Environn	nent	· · · ·	•	\$
History Collections APIs						
+ New Collection Tr	1					
Microsoft Test App 0 requests						
	Open Launchpad Comparison C	g <mark>Start X</mark>				
Find and Replace E Console Console	습· Bootcamp	Build	Browse		μ 	?

Open Postman and create a new collection by selecting **New Collection** and give it a name. For this example, we will name the collection "**Microsoft Test App**". You can also add a description of it if you wish, then press **Create**.

These collections are where you will store the requests we will send to Microsoft's servers. We will create the requests first, and then we'll begin the process of obtaining the tokens we need.

The three requests we'll create in Postman are:

Authorization

This request is used to obtain the authorization code and is the first step towards getting the bearer token. This request is sent with GET to <u>https://login.microsoftonline.com/common/oauth2/v2.0/authorize</u> and the query parameter will provide your client ID and other needed information.

Request Token

This request is used to obtain the bearer and refresh tokens needed to access the user's account. This is done using a POST to <u>https://login.microsoftonline.com/common/oauth2/v2.0/token</u> and the post data will contain the authorization code provided in the previous step.

Refresh Token

This request is used to refresh the bearer token after it has expired. Bearer tokens have a short lifespan, typically an hour, and will expire. Refresh the token is also done using a POST to https://login.microsoftonline.com/common/oauth2/v2.0/token with the refresh token provided in the post data.

🤗 Postman		-	- 🗆	×
File Edit View Help				
🕂 New Import Runner 📭 🔻	🖬 My Workspace 🔻 🕹 Invite 🧿 📽 🖡	• • •	Upgrade	
Q Filter	No Enviro	onment	• •	\$
History Collections APIs				
+ New Collection Trash				
► Microsoft Test App ☆ 0 requests				
→ Share Collection ▲ Manage Roles ▲I Rename Ctrl+E ✓ Edit 【:] Create a fork [:] Create Pull Request [:] Merge changes [:] Add Request [:] Add Folder [:] Duplicate Ctrl+D ↓ Export 小 Monitor Collection [:] Publish Docs ↓ Remove from workspace [:] Delete Del	Open Launchpad Learn how to debug requests and perform manual testing Start X			
Q Find and Replace Image: Console	🕒 Bootcamp 🛛 Build	Browse	= • ज	0

Create the first request, which is Authorization. Select **Add Request** from the menu on the left side of the display, and name it. You can also add a description if you wish.

💋 Postman				– 🗆 X
File Edit View Help				
🕂 New Import Runner 📭 🔻	My Workspace 🔻	🛃 Invite	० २ २ २ २ २	∎ Upgrade ▼
Q Filter	GET Authorization X + ••••		No Environment	• • *
History Collections APIs	Authorization		Examples 0 🔻	BUILD 💉 📮
+ New Collection Trash	GET v https://login.microsoftonline.com	n/common/oauth2/v2.0/authorize?client_id=5729d13	3a-7464-4e04-t Send	▼ Save ▼
Microsoft Test App			Send	Save
1 request	Params Authorization Headers (7) Be	ody Pre-request Script Tests Settings		Cookies Code
GET Authorization	KEY	VALUE	DESCRIPTION	••• Bulk Edit
	client_id	1778413a 7466 469 artista) 8,001,002		
	scope	offline_access https://outlook.office.com/POP./		
	response_type	code		
	response_mode	query		
	redirect_uri	http://127.0.0.1/testApp/		
	Key	Value	Description	
	Response			~
		Hit Send to get a response		
	📩, Learn	how to debug requests and perform manual testing	g Start X	
		😌 Bootcamp	Build Browse	v ?

Name the request **Authorization** and it will default to using GET and enter the URL to Microsoft's authorization server https://login.microsoftonline.com/common/oauth2/v2.0/authorize

Populate the query parameters with five values which comprise the request.

client_id	The client ID you obtained when you registered your application with Azure.
scope	This informs Microsoft what kind of access you want to the user's mail account. In this guide, we'll request access for POP3, IMAP and SMTP. The scopes are:
	offline_access https://outlook.office.com/POP.AccessAsUser.All https://outlook.office.com/IMAP.AccessAsUser.All https://outlook.office.com/SMTP.Send
	Each scope should be separated by a space. We'll include offline_access so that Microsoft will provide us with a refresh token.
response_type	The response type should always be code for our purposes. This tells Microsoft we want an authorization code.
response_mode	The response mode should always be query which indicates we want the authorization code returned as a query in our redirect URI.
redirect_uri	This is the redirect URI we provided when we registered the application. This should be http://127.0.0.1/testApp/

🤣 Postman File Edit View Help			– 🗆 X				
🕂 New Import Runner 📭 🔻	My Workspace 🔻	🕹 Invite	🧿 📽 🖨 🌲 🎔 🏢 Upgrade 🔻				
Q Filter	GET Authorization POST Request	Token X + ••••	No Environment 🔻 💿 🌞				
History Collections APIs	Request Token		Examples 0 🔻 BUILD 💉 📮				
+ New Collection Trash	POST • https://login.microsoftonline.com	n/common/oauth2/v2.0/token	Send v Save v				
2 requests	Params Authorization Headers (9) Body	Pre-request Script Tests Settings	Cookies Code				
GET Authorization	none form-data e x-www-form-urlencod	led 🔴 raw 🔍 binary 🔍 GraphQL					
POST Request Token	✔ client_id	TOM IN MERINARY REPORTS					
	✔ code						
	scope	offline_access https://outlook.office.com/POP./					
	redirect_uri	http://127.0.0.1/testApp/					
	grant_type	authorization_code					
	Кеу	Value	Description				
	Response						
	Hit Send to get a response						
	, Learn	how to debug requests and perform manual testing	Start X				
S Find and Replace Console		🔂 Bootcamp	Build Browse				

Next, create the **Request Token** request which uses POST. Select **Body** from the menu, and the encoding type **x-www-form-urlencoded** as shown above. The URL is https://login.microsoftonline.com/common/oauth2/v2.0/token

Populate the parameters with five values which comprise the request.

client_id	The client ID you obtained when you registered your application with Azure.
code	This is the authorization code which will be obtained using the previous request. We don't have the code yet, so leave this blank.
scope	This is the same list of scopes as provided with the authorization request.
redirect_uri	This is the redirect URI we provided when we registered the application. This should be http://127.0.0.1/testApp/
grant_type	This value should be authorization_code which tells Microsoft we want the bearer and refresh tokens.

🧭 Postman			– 🗆 X			
File Edit View Help						
🕂 New Import Runner 📭 🔻	My Workspace 🔻	🗼 Invite	🖸 📽 👂 🌲 🎔 🏥 Upgrade 🔻			
Q Filter	GET Authorization POST Request Token	POST Refresh Token X + ···	No Environment 🔻 💿 🌞			
History Collections APIs	 Refresh Token 		Examples 0 🔻 BUILD 💉 📮			
+ New Collection Trash	POST v https://login.microsoftonline.com	n/common/oauth2/v2.0/token	Send - Save -			
Microsoft Test App 3 requests	Params Authorization Headers (9) Body	✓ ● Pre-request Script Tests Settings	Cookies Code			
GET Authorization	none form-data • x-www-form-urlencod					
POST Request Token	KEY	VALUE	DESCRIPTION ••• Bulk Edit			
Neirean roken	client_id	1779ad Tao 7666 (april artist of 16,000) (350				
	✓ refresh_token					
	redirect_uri	http://127.0.0.1/testApp/				
	grant_type	refresh_token				
	Кеу	Value	Description			
	Response					
		Hit Send to get a response				
	👌, Learn	how to debug requests and perform manual testing	Start X			
		🔁 Bootcamp	Build Browse			

The final request type will be **Refresh Token** which is used to refresh an expired bearer token. This will also use a POST to https://login.microsoftonline.com/common/oauth2/v2.0/token

There are four values which comprise the request.

client_id	The client ID you obtained when you registered your application with Azure.
refresh_token	The refresh token you obtained when the initial authorization was granted. This token has a long validity period and is required to get a new bearer token after it has expired. We don't have the refresh token yet, so leave this blank.
redirect_uri	This is the redirect URI we provided when we registered the application. This should be http://127.0.0.1/testApp/
grant_type	This value should be refresh_token which tells Microsoft we want a new bearer token.

Postman				– 🗆 X
File Edit View Help	👪 My Workspace 🔻 🖌	🛵 Invite	3 v r a v 🗉	= Upgrade 🔻
Q Filter	GET Authorization X POST Request Token	POST Refresh Token + ···	No Environment	• • •
History Collections APIs	 Authorization 		Examples 0 🔻	BUILD 💉 📮
+ New Collection Trash	GET • https://login.microsoftonline.com/	common/oauth2/v2.0/authorize?client_id=	Send	▼ Save ▼
Microsoft Test App 3 requests	Params Authorization Headers (7) Bod	ly Pre-request Script Tests Settings		Cookies Code
GET Authorization	KEY	VALUE	DESCRIPTION	••• Bulk Edit
POST Request Token POST Refresh Token	<pre>client_id</pre>	177564736-7666-6686-6876-62786-8876-6836		
Post Refresh loken	scope	offline_access https://outlook.office.com/POP./		
	response_type	code		
	response_mode	query		
	✓ redirect_uri	http://127.0.0.1/testApp/		
	Key	Value	Description	
	Response	Hit Send to get a response	s Start X	·
G Find and Replace F Console		🗇 Bootcamp	Build Browse	· · · ?

Now that we have created all three request types and saved them in Postman, we can use them to obtain our bearer token. Copy the URL from the **Authorization** request and paste it into a browser address bar.

You should not click **Send** in Postman, because this step requires you to login with your Microsoft 365 account and this must be done interactively in a browser.

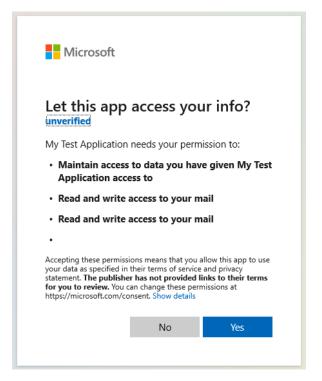
	New	r tab	×	+ ~										
÷	Ö	ŵ	+ttps://lo	gin.microsoftonl	ne.com/common/	/oauth2/v2.0/autho	orize?client_id=	(36) (36 (166)	ana ana ana	&us co	pe=offline_ac	cess https://outlo	ok.office.com/POP.	AccessAsUser.A
		Micro	osoft									<u>م</u>		

When you hit enter, you will be taken to Microsoft's site and prompted to login with your account. If you have two-factor authentication enabled, you will need to confirm the login attempt.

Because your application hasn't been verified, you will get a warning from Microsoft about your test application accessing your account. This is normal during the development and testing period for your application. However, when it's time for you to release your application to the public, you will need to go through the verification process.

You can learn more about <u>publisher verification</u> on Microsoft's website.

After you've logged in, you will be asked to confirm that you want My Test Application to have access to your account.



Once you've completed the login process and confirmed you want your test application to have access to the account, Microsoft will redirect back to the redirect URI which you specified when the application was registered with Azure.

\square Can't reach this page $ imes$ $+$ $ imes$		
→ ① û 127.0.0.1/testApp Code=M.R3_BAY.	all' ann fha fhr to the	□ ☆
,°	Hmmmcan't reach this page Try this • Make sure you've got the right web address: http://127.0.0.1 • Search for "http://127.0.0.1" on Bing • Refresh the page Details • Report this issue Privacy statement	

Because there won't be any web server able to respond, the browser will return an error. But this isn't a problem because the information we need is returned in the address bar. You will notice the address bar now has a value which looks like your return URI:

http://127.0.0.1/testApp/?code=[your_authorization_code]

What follows the request URI after the **?code=** portion is the authorization code we're interested in. This is going to be a string of letters, numbers, and symbols which you should copy and paste into **Request Token** request we've created in Postman.

Edit View Help					
New Import Runner 📭 🔻	My Workspace 🔻	🕯 Invite 📀) & * * * (Upgrade	
C Filter	GET Authorization POST Request Token	POST Refresh Token + •••	No Environment	• •	
History Collections APIs	Request Token		Examples 0 v	BUILD 💉 I	
- New Collection Trash	POST v https://login.microsoftonline.com/common/oauth2/v2.0/token Send v Save				
Microsoft Test App 3 requests	Params Authorization Headers (9) Body	 Pre-request Script Tests Settings 		Cookies (
GET Authorization	🔵 none 🛛 form-data 🏾 🖲 x-www-form-urlencode	d 🔵 raw 🔵 binary 🔵 GraphQL			
OST Request Token	client_id	1979bal Tau-Pelen Autor antiprasit Robbit antis			
ost Refresh Token	Code	M.R3_BAY			
	✓ scope	$off line_access \ https://outlook.office.com/POP.{}$			
	✓ redirect_uri	http://127.0.0.1/testApp/			
	✔ grant_type	authorization_code			
	Кеу	Value	Description		
	Body Cookies (4) Headers (14) Test Results	Can Status: 200 OK	Time: 616 ms Size: 2.48 KB	Save Respons	
	Pretty Raw Preview Visualize JSON 1 2 "token_type": "Bearer", 3 "scope": "https://outlook.office.com	POP.AccessAsUser.All https://outlook.office	.com/IMAP.AccessAsUser.All	https://outlool	
	<pre>4 "expires_in": 3600, 5</pre>	Thursday), davide - car - Buarrier - 71() - og (Millionauto o Ref 5 i Thursday og Turk, Frank andre 15 i Jackson og Sando	anan te ta suata da te a su de terra an 1976 - San	Distriction Pade	

With the authorization code pasted into the **code** field, press Send to send the request.

If all has gone well, the server will return a JSON response which contains your bearer token (named access_token) and the refresh token. You will note the expires_in value is 3600 seconds, which is one hour. This is how long the token can be used to access the account you logged in with. After the token expires, you'll need to refresh it.

To obtain a new, fresh bearer token you will need to send another request to the server. Copy the refresh_token value into the **Refresh Token** request we created earlier and click **Send**.

New Import Runner 📭 🔻	My Workspace 🔻	🛃 Invite) v + 4 • 🖲	Upgrade
k Filter	GET Authorization POST Request Token	POST Refresh Token + ····	No Environment	• ©
History Collections APIs	▶ Refresh Token		Examples 0 🔻	SUILD 💉 E
New Collection Trash				
Microsoft Test App	POST	/common/oauth2/v2.0/token	Send	 Save
3 requests	Params Authorization Headers (9) Body	Pre-request Script Tests Settings		Cookies (
ET Authorization	none form-data s-www-form-urlencod			
OST Request Token				
DST Refresh Token	KEY	VALUE	DESCRIPTION	••• Bulk Ed
	✓ client_id	27729641 Taur 1966en 16486en 600381 4421 1662881 148530		
	refresh_token	M.R3_BAY		
	redirect_uri	http://127.0.0.1/testApp/		
	grant_type	refresh_token		
	Key	Value	Description	
	Body Cookies (4) Headers (14) Test Results	Can Status: 200 OK	Time: 530 ms Size: 2.49 KB	Save Response
	Pretty Raw Preview Visualize JSON			
	<pre>1 { 2 "token_type": "Bearer", 3 "scope": "https://outlook.office.com 4 "expires_in": 3600, 5 "ext_expires_in": 3600, 6 ("access_token":)"EwBgA+ 7 Tetresn_token": "M.R3_BAY. 8 }</pre>	m/POP.AccessAsUser.All https://outlook.office	e.com/IMAP.AccessAsUser.All h	ttps://outlook

By providing the refresh token, the server will return a new bearer token which is valid for another hour. Microsoft will also send a new refresh token as well, but you can continue to use the original refresh token which was obtained with the initial authorization request.

This completes the guide of how to register your application with Azure Active Directory and use that to obtain the bearer and refresh tokens for a Microsoft 365 account. By performing this process manually using Postman, it should help clarify the process of how the tokens are obtained and how an expired bearer token is refreshed.

If you are using SocketTools .NET or the SocketTools ActiveX controls, the mail components have a new property called **BearerToken** which you can use to specify the OAuth 2.0 bearer token in lieu of a password. For the SocketTools Library Edition, the various APIs have been extended to support bearer tokens as a valid authentication method. Refer to the technical reference documentation for more specific information about how to update your application to use OAuth 2.0 authorization.