part of eex group

grexel

G-REX API Manual

05.08.2025

Version 3.2

1. API: Disclaimer, Fair use, and best practices

1.1 Disclaimer

G-REX APIs are designed to expose some key features of G-REX as a HTTPS service, allowing the use of these features in a more efficient way without having to use the conventional user interface (website). E.g., adding scheduled transactions to an account holder's account can be done exponentially faster, with the possibility to insert over 100 transactions per minute.

It is critical that the API users who are designing applications for connecting and posting transaction requests to the API make sure that the deployed applications do not create transaction loops (same transaction / or transaction sequences repeatedly posted to the API). To prevent this, please refer to the Fair use policy and Best Practices sections of this manual.

Grexel Systems Ltd. Does not take any responsibility for errors caused by user applications.

Grexel Systems Ltd. Does not take any responsibility for the use of any 3rd Party tools. The users accept full responsibility for using 3rd party tools and acknowledge the use of such tools at their own risk. Grexel disclaims all liability for any direct, indirect, or consequential damages, including data loss or financial loss, resulting from 3rd party tool usage. Users use 3rd party tools voluntarily and release Grexel from any claims or obligations.

1.1.1 API Support

Any possible support regarding the G-REX API integration would require a direct contractual relationship with Grexel and the requesting party. Information about API support and contractual matters can be requested from info@grexel.com or from the Grexel support portal.

1.2 Fair Use policy

To maintain optimum performance, and to ensure that the API is available to all our customers, limits are imposed on the usage of the API. Applications should be written in a way which does not violate this policy.

1.2.1 Accepted request frequencies.

The following outlines the limit for the API requests:

- 500 requests per hour
- 100 requests per minute

1.2.2 Request types to avoid.

- Requests without minimal filtering of $top and $skip filter functions should not be done.

- Certain types of requests should be generally avoided. They can be performed for testing purposes or to create a basis for local databases but should not be committed automatically or regularly.
- By any means avoiding:
    - GET all available account holder transaction information.
    - GET all available account holder certificate information.

Using these large-scale queries to get statistical information for reporting or larger overview is allowed if the requests are not automatically repeated over short periods of time (polling). If large amounts of information are required to be updated regularly, the best practice would be to execute the full dataset query **with $top and $skip** once and update the obtained data by performing smaller queries that can be built by adding filers to full dataset queries.

1.2.3 Monitoring and blocking

The API provider can monitor the API usage automatically in real time or by retrospective log analysis. If a Fair Use Policy violation is detected, the violating account holder may be temporarily blocked from making additional calls.

No fair usage statistics or metrics are provided. Instead, account holders should monitor their own API usage or build the limitations to client software.

1.3 Best Practices

1.3.1 HTTP standard

HTTP standards should be enforced for all requests. Requests should have the correct headers and necessary information. Required headers are:

- Accept
- Content-Type
- Content-Length
- Host
- User-Agent

Note: Missing headers might result in a violation in the firewall and multiple violations will block the request. (A blocked request in the firewall will result in a HTTP Status Code 403: Forbidden).

1.3.2 Break for confirmation.

G-REX Transaction API sends a response message for each write request (POST, PUT, PATCH and DELETE functions) sent to the API. The response consists of a status code and an attached message.

To minimize the risk of posting the same transaction twice or posting the same sequence of transactions multiple times to the API, it is useful to first verify the successful completion of the first request via successful response code before submitting the next requests. Once the successful response is displayed to the user, only then should the next request be sent.

1.3.3 Use filters and correct headers

- Requests should include filtering of $top and $skip filter functions when retrieving information.
- Please use correct (REST API) headers when making a request.

2. Introducing G-REX

G-REX is a new generation certificate registry which gives Issuing Bodies and Account Holders an agile and secure way to manage different types of energy certificates in one user-friendly system. G-REX is built to support energy certificates with different standards and to adapt to the different needs of each Issuing Body. This user manual explains some of the key processes behind authentication and use of G-REX APIs.

The G-REX application consists of API interfaces and the user interface. APIs are responsible for handling any operations G-REX allows the users to do. All features of G-REX can be performed directly through the APIs that follow the REST standard or through the user interface. The User interface is based on Angular framework.

Note: For access to the G-REX UX User Manuals for Issuing Body and Account Holder users, please download those from the G-REX registry UI or contact the Grexel team.

2.1 G-REX APIs, Authentication, Postman, and documentation upload

2.1.1 Authentication and G-REX

The G-REX built-in authentication and authorization uses Microsoft Entity ID (previously called Azure Active Directory) B2C (ME-ID B2C) including Multi-factor Authentication. After authentication, the ME-ID B2C issues a token that the user uses to access the services regardless of whether the user is accessing the system from G-REX website, APIs directly, or via an external system. Multi-Factor Authentication is required when logging in and could be required as a confirmation in other cases as well.

2.1.2 API platform for API calls (Postman)

Authentication for G-REX APIs can be for example done via the Postman API platform. Before accessing G-REX APIs, it is first necessary to download Postman to your device. Postman is a free-to-use tool and sign-up might be required.

Postman can be downloaded from the following link: https://www.postman.com/downloads/

Please note that Postman is a 3rd party tool (see chapter 1.1) and similar tools are available.

Grexel has no official affiliation with Postman. Users accept full responsibility for Postman and acknowledge the use of Postman at their own risk. Grexel disclaims all liability for any direct, indirect, or consequential damages, including data loss or financial loss, resulting from Postman usage. Users use Postman voluntarily and release Grexel from any claims or obligations.

2.2 G-REX environments, token use cases and collections

To get started, you need to import the G-REX environments into Postman (Chapter 2.2.1). An environment is needed for both 1) G-REX where you will make the API calls and 2) G-REX tokens, where you will get access tokens to put into the G-REX environment (1).

These environments are available in appendices 1 and 2 (respectively) for G-REX production and in appendices 4 and 5 (respectively) for G-REX demo. These environments contain pre-filled variables for use with G-REX production domains. Please see chapter 2.2.1 on how those are imported to postman.

In addition to the environment, you also need to import the G-REX token use cases collection, which contains the GET PKCE request for obtaining an access token via sign-in, as well as a POST request for getting an access token via a refresh token. This collection is available in Appendix 3 and is the same for G-REX production and demo. The import of the collection is described in chapter 2.2.2.

When both environments and token use case collections are imported, you can move into importing actual G-REX API collections, which are used to get and post requests to G-REX. These collections can be imported from the G-REX open API documentation, following the instructions in chapter 2.2.3.

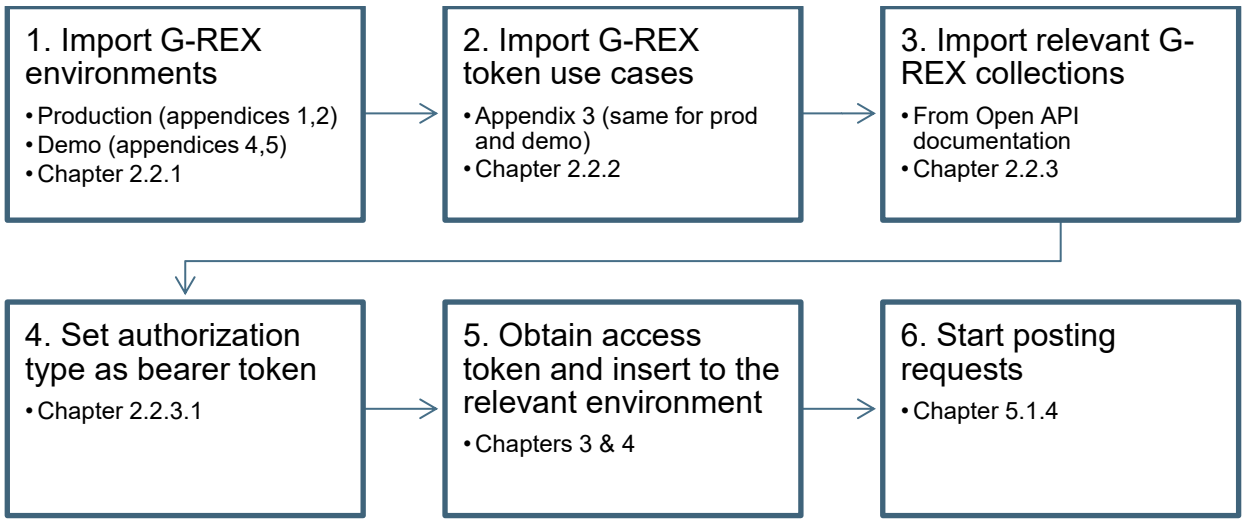

Figure 1 Basic flow for getting started

2.2.1 Importing the G-REX Environment

One must set up the G-REX environment before making API calls. To set up the environment follow these steps:

1. Copy the raw text from Appendix 1: G-REX Production environment for Postman
2. Open Postman and press import (See Figure 2 Import to postman)
3. Paste the raw text into the text field (See Figure 3 Pasting raw text)
4. The Environment will be imported and will appear on the left in the Environments menu (See Figure 4 Postman Environments menu
5. Repeat the same process for G-REX token environment (appendix 2) as well as, where applicable, for G-REX DEMO environment (appendix 4) and for G-REX DEMO - B2C token environment (appendix 5).

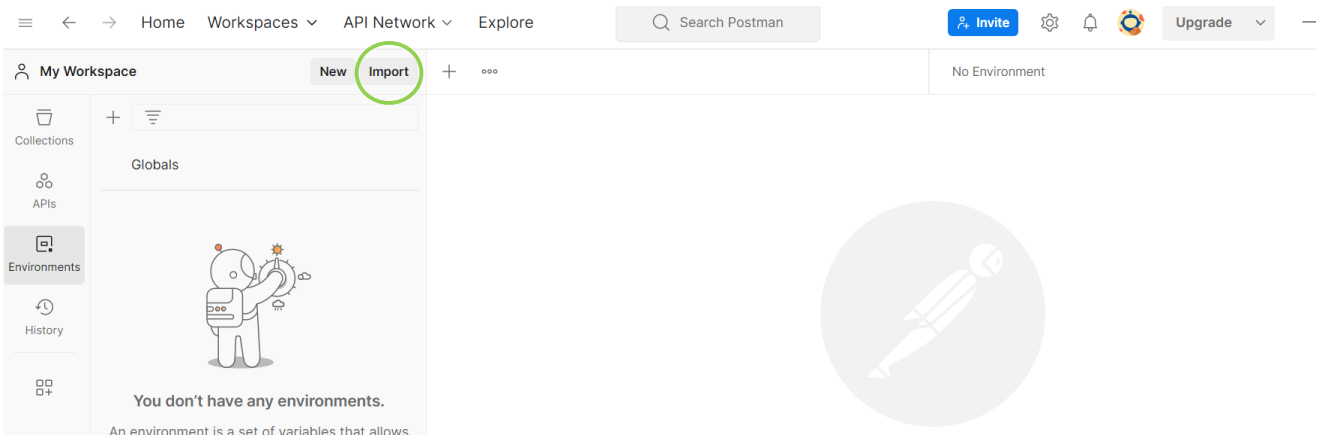

Figure 2 Import to postman

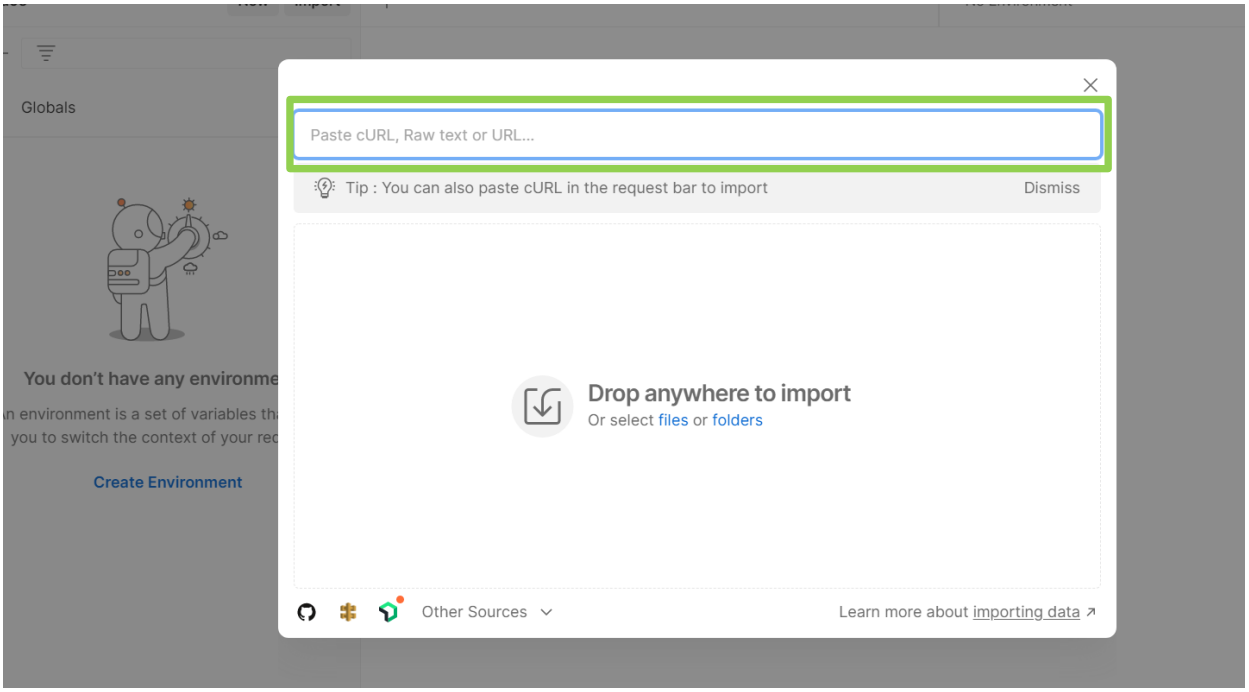

Figure 3 Pasting raw text

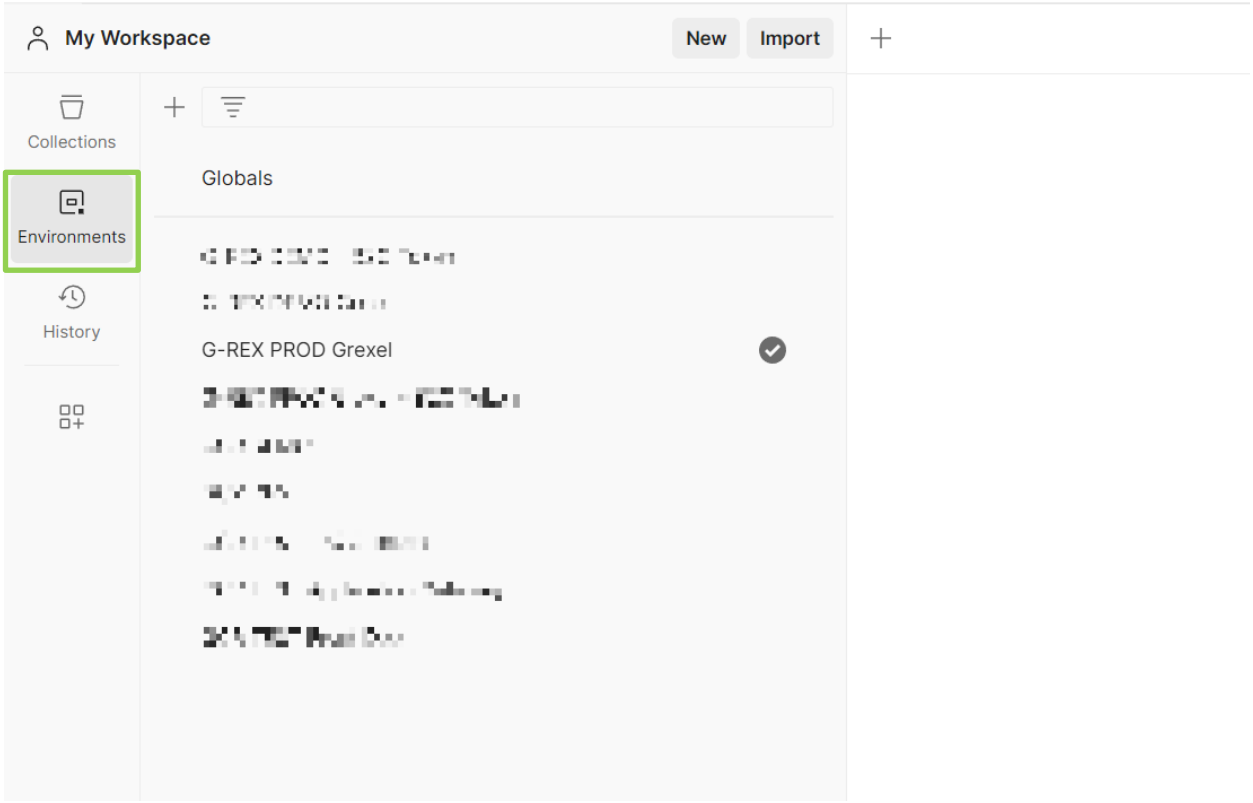

Figure 4 Postman Environments menu

2.2.2 G-REX Token Use cases and B2C token environment

To authenticate to G-REX via postman a collection for Authentication calls is available for import as raw text in Appendix 3: G-REX token use cases for Postman. To use this collection, one also needs to import the B2C Token environment Available as raw text for Import. To import the environments, follow the steps as previously in 2.2.1.

For importing the token use cases, the process is very similar:

1. Copy the raw text from appendix 3.
2. Open Postman and press import (See Figure 2 Import to postman)
3. Paste the raw text into the text field (See Figure 3 Pasting raw text)
4. The use cases will appear in the collection's menu (See Figure 5 Collections menu)
5. The token user cases collection is the same for G-REX production and demo.

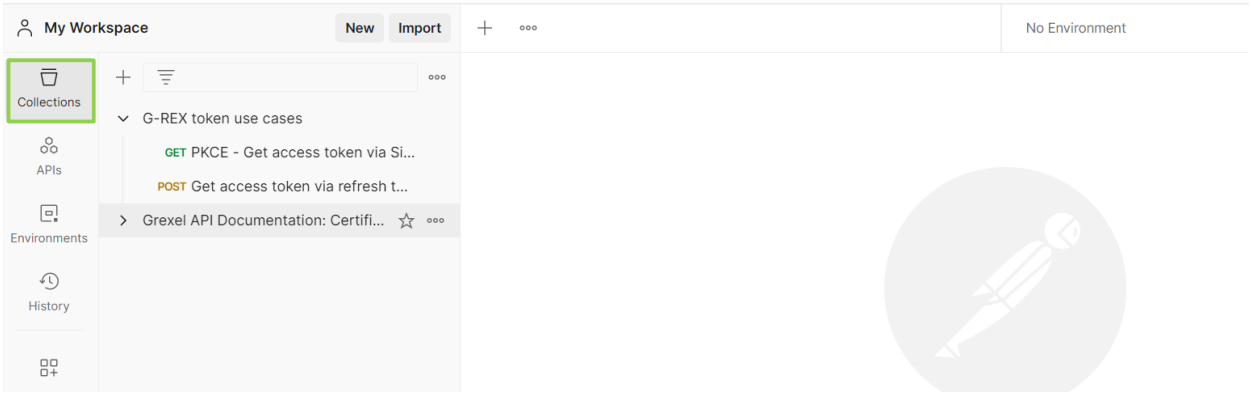

Figure 5 Collections menu

2.2.3 Importing postman collections and API documentation via swagger link

To import the documentation and the postman collection for a certain API into postman, a user can navigate to any API documentation and find the data via the swagger.json link on the top left under the title of the API documentation.

Note: Links to open API documentations in Chapter 5 Open API Documentation.

Follow these steps to import the data:

1. Open the API documentation, you will find a link named swagger.json (see Figure 6 below)

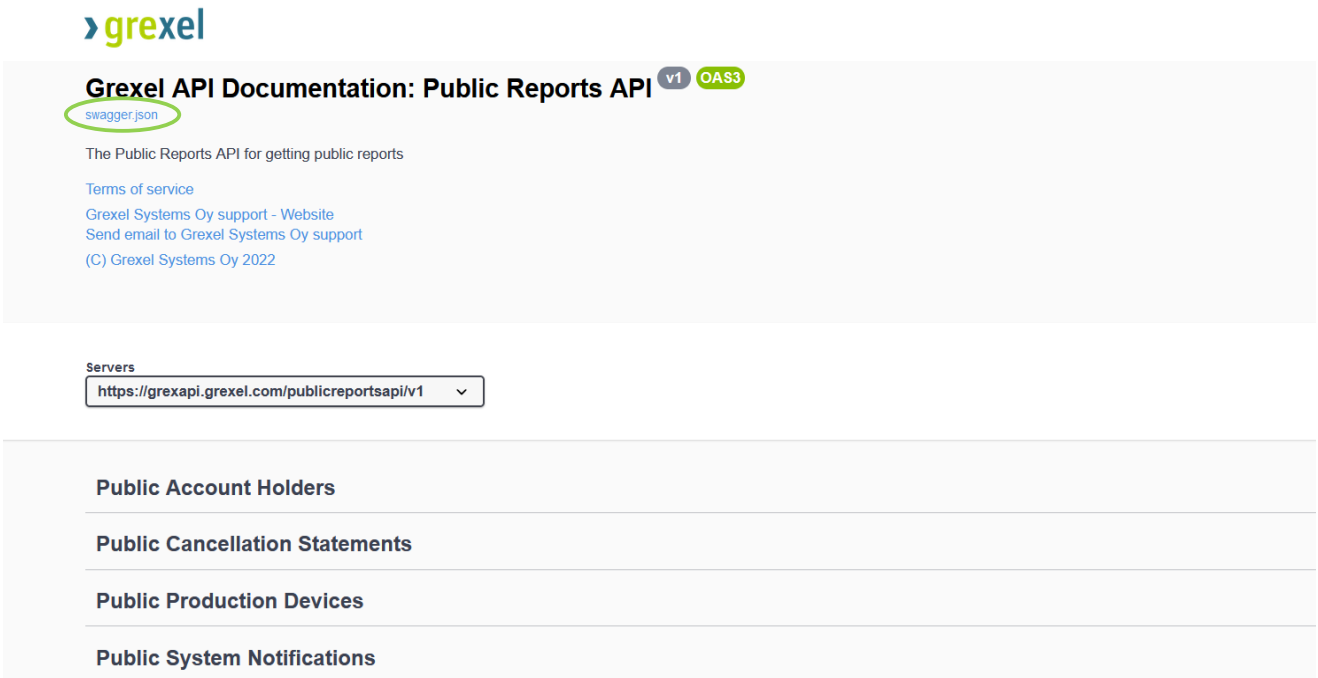

Figure 6 Public reports API, Open API documentation

2. Once you open the link open the “raw data” tab and copy the raw data. For some browsers, there may not be a separate tab for raw data, instead it will be visible as soon as you click the swagger.json link. In this case, copy the raw data by selecting all the text with Ctrl+A and copying it with Ctrl+C.

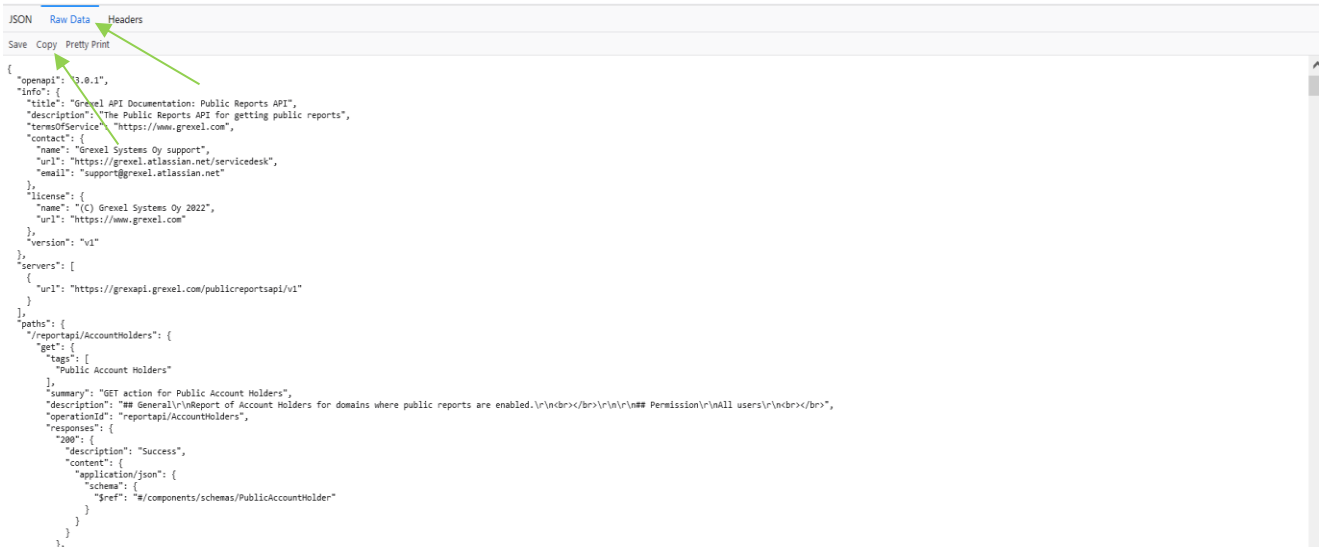

Figure 7 JSON raw text

3. Once copied, open postman, Press the **Import** Button and paste the raw text copied there.

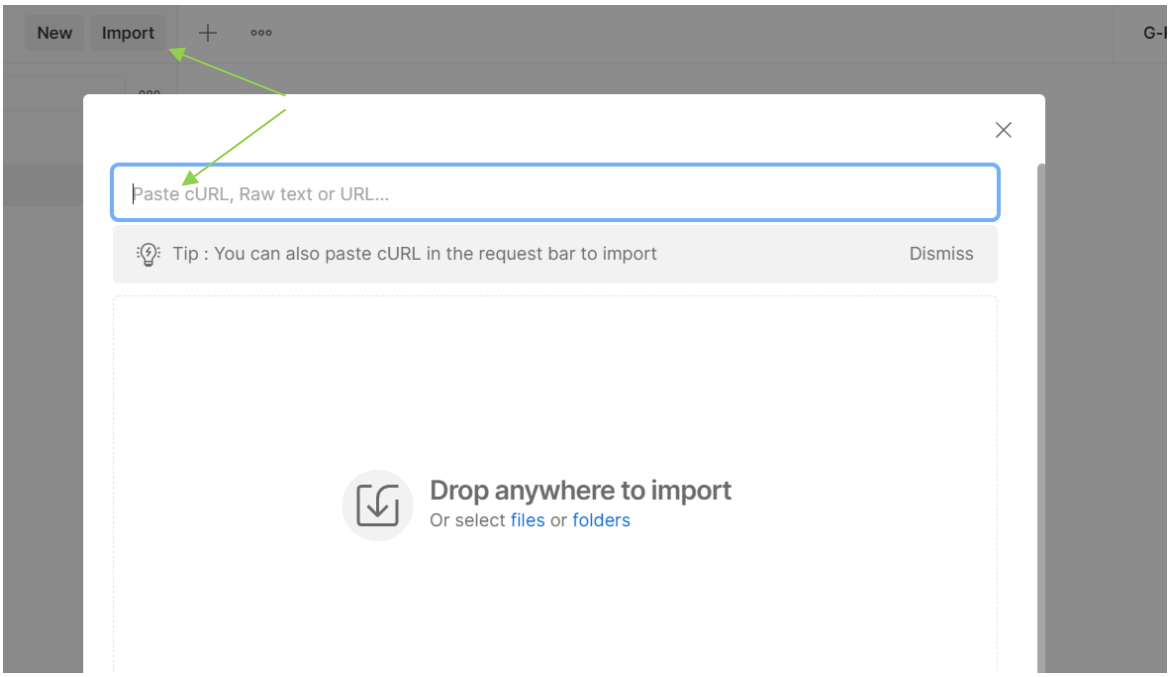

Figure 8 Insert to Postman

Done: The documentation and the collection for that specific API will be available in Postman.

2.2.3.1 Setting the Authorization type

Note: After importing the collections and configuring the environment, the Authorization Type for the collections needs to be set.

Editorial Note: The terms AccessToken and Token are sometimes used interchangeably but refer to the same field.

1. Select the parent of the collection from the collections list (e.g., Grexel API documentation: Management API)
2. Navigate to the Authorization tab.
3. Set the type as Bearer Token
4. Set the Token field to match the variable name in the authentication variables (see Figure 14 and Figure 19) setup (e.g., “{{Token}}”) See Figure 9 below.
5. Press Save

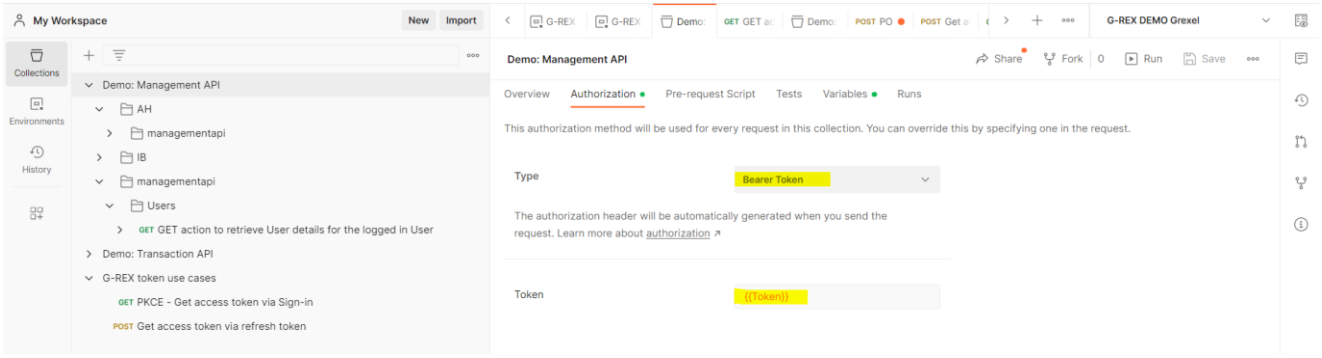

Figure 9 Setting the authorization type for a collection.

3. Retrieving access tokens via the PKCE flow in Postman

Note: This section assumes that Postman has already been configured to your specific authentication requirements. For configuration of Postman, revisit section 2

3.1 Retrieving tokens

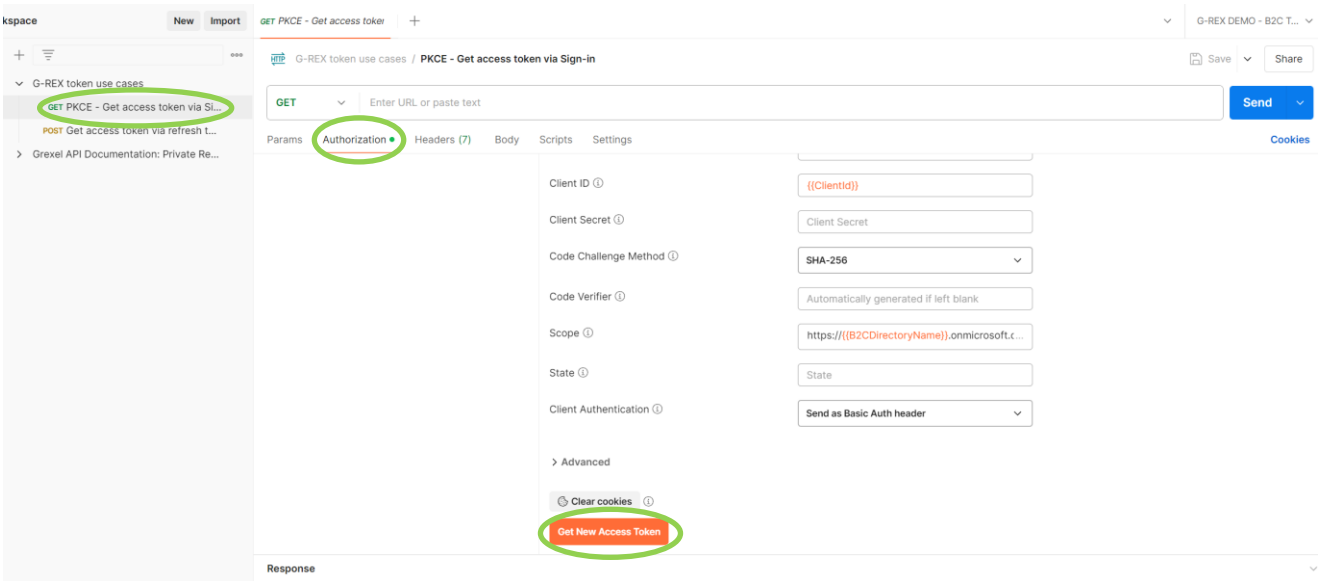

Figure 10 Retrieving access tokens via the PKCE flow in Postman

Once Postman is configured, access tokens should be retrieved by clicking on the *PKCE – Get access token via Sign-in* request on the left side of the window. Navigate to “Authorization”, scroll down and press “Get New Access Token”.

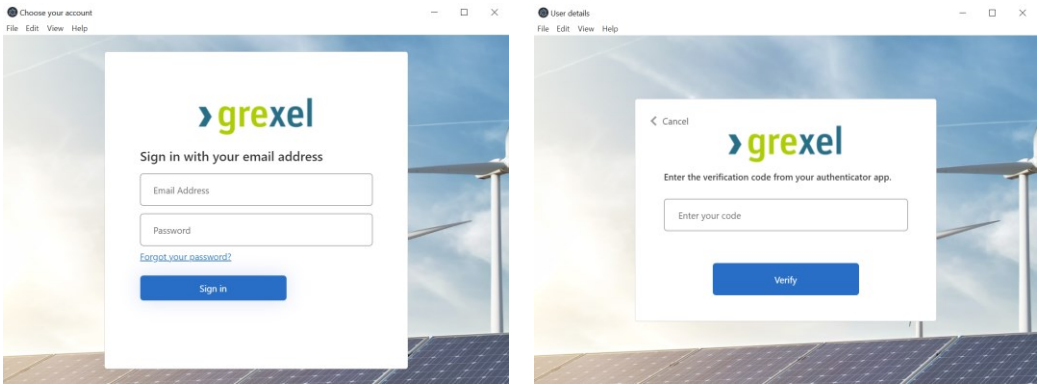

Figure 11 Sign-in window for access token retrieval

A pop-up window will appear prompting you to sign in with your G-REX credentials. This will be followed by a request for a code from the Microsoft Authenticator app. For information on the Multi-factor Authentication process and set up, please consult the G-REX UX User Manuals.

After successfully authenticating, press “Proceed” to retrieve the Access Token and Refresh Token.

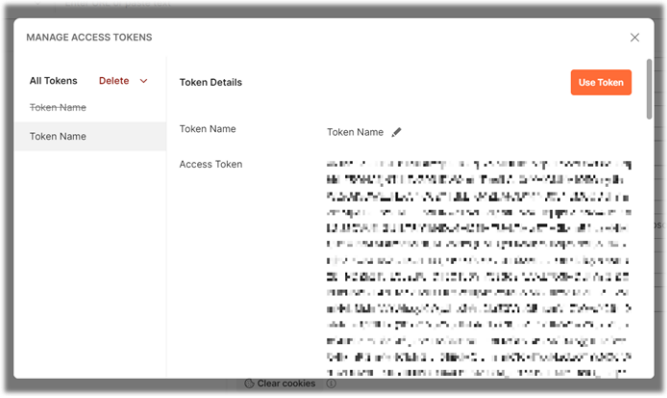

Figure 12 View and copy access token

Figure 13 View and copy refresh token

3.1.1 Adding a Refresh Token to an environment

To add an access or refresh token to the environment, the token should be copied to be inputted into the **G-REX Environment** by pressing the variables icon in the top right of Postman (circled in the below figure). Among the variables there is a space for inputting the value of the Refresh Token. The field may be empty or contain a previously used token value. Paste the Token / Refresh Token into the relevant field. For additional info about extending the refresh token lifetime, see chapter 3.2.1 Refresh token extension.

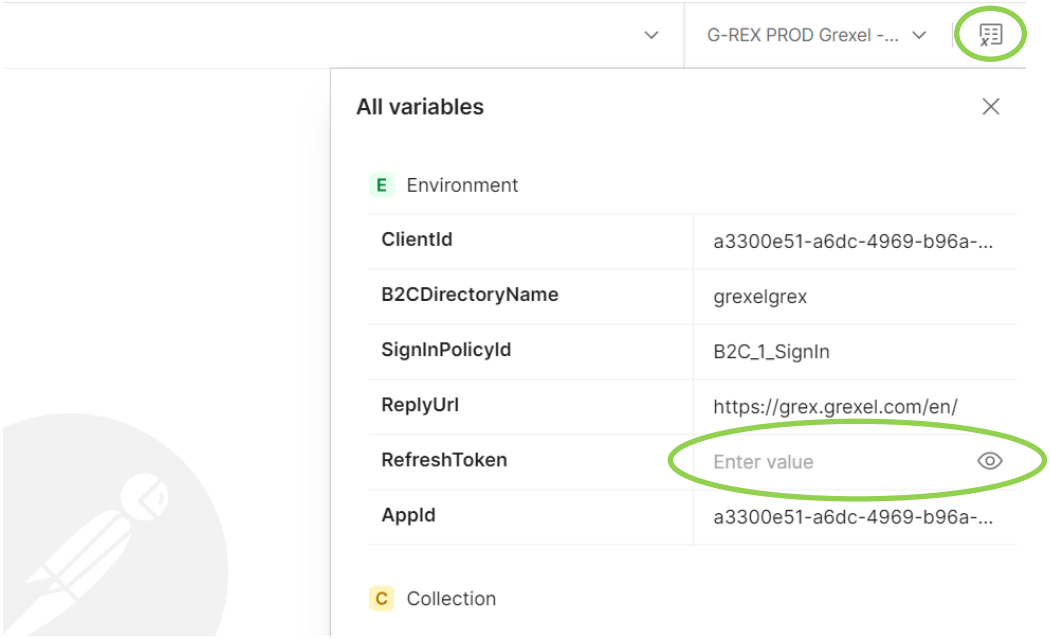

Figure 14 Adding a refresh token to an environment

3.2 Obtaining Access Tokens using a valid Refresh Token

If a valid Refresh Token is inputted in the corresponding environment variable field, an Access Token can be retrieved by using the request “POST: Get access token via refresh token” circled in the image below.

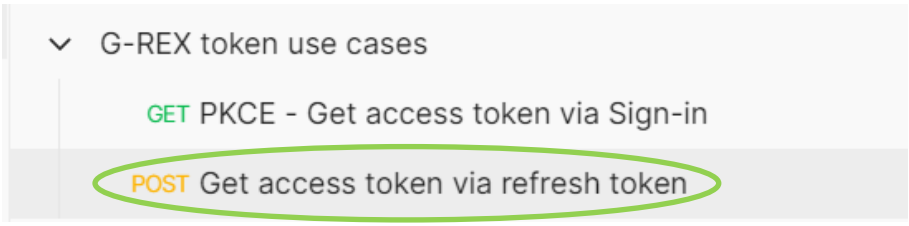

Figure 15 Obtaining Access Tokens using a valid Refresh Token

Note: Make sure you are in the B2C token environment before making the call to get the access token (See Figure 16 below)

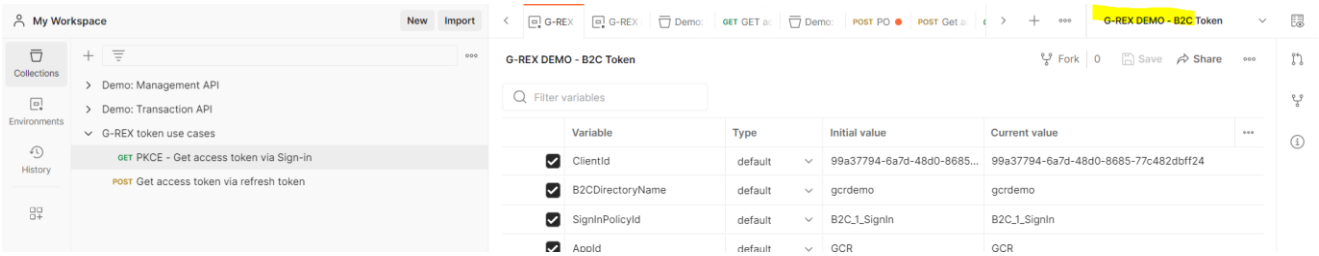

Figure 16 Selecting B2C Token environment for obtaining access token

If the Refresh Token variable is valid and the environment is otherwise correctly configured, then this request can be sent with no other changes. After pressing “Send,” the Access Token will be returned. This access token should be copied and pasted according to Chapter 3.1.

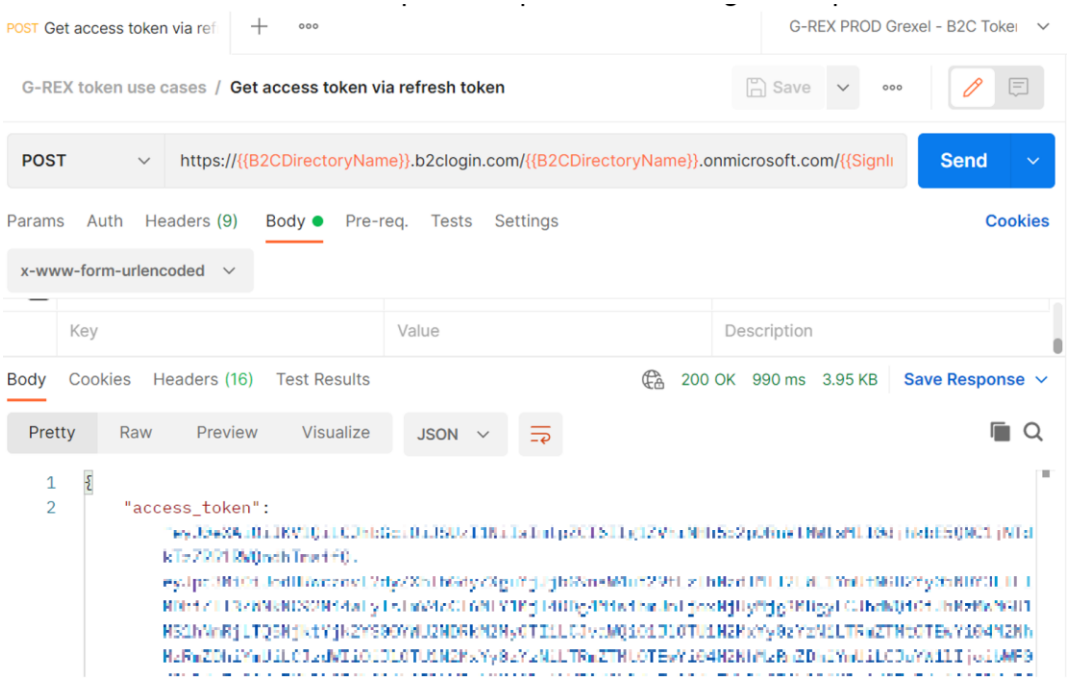

Figure 17 Obtaining Access Tokens using a valid Refresh Token (2)

3.2.1 Refresh token extension

The refresh token lifetime can be extended by changing the ReplyURL – see below:

- For a refresh token with max 24-hour lifetime (recommended for single-page-applications (SPAs)): **https://eexfrance.grexel.com/en/**
- For a refresh token with the maximum configured lifetime: **https://gcrfr.b2clogin.com/oauth2/nativeclient**

By using the alternative ReplyURL (https://gcrfr.b2clogin.com/oauth2/nativeclient), the refresh token lifetime will be 7 days, rather than 24 hours.

In addition, there is a 90-day sliding window enabled on the refresh token. This means that if you are to fetch a new refresh token within the 7-day lifetime of your existing refresh token, no repeat sign-in/MFA will be required. This can be done for up to 90 days, after which sign-in/MFA will be required again.

Note: For security measures, the longest supported time that a refresh token can be renewed without manual intervention (sign in/MFA) is 90 days. When using the alternate ReplyURL, a new refresh token can be fetched without sign in/MFA within the 7-day lifetime of your existing token, and this process can be repeated for up to 90 days, after which manual sign-in/MFA is required again. After authentication, the 90-day sliding window restarts.

3.3 Using tokens

To complete authorization and use the access token to make API requests, the token needs to be copied and pasted into the right environment.

Start by selecting the right environment.

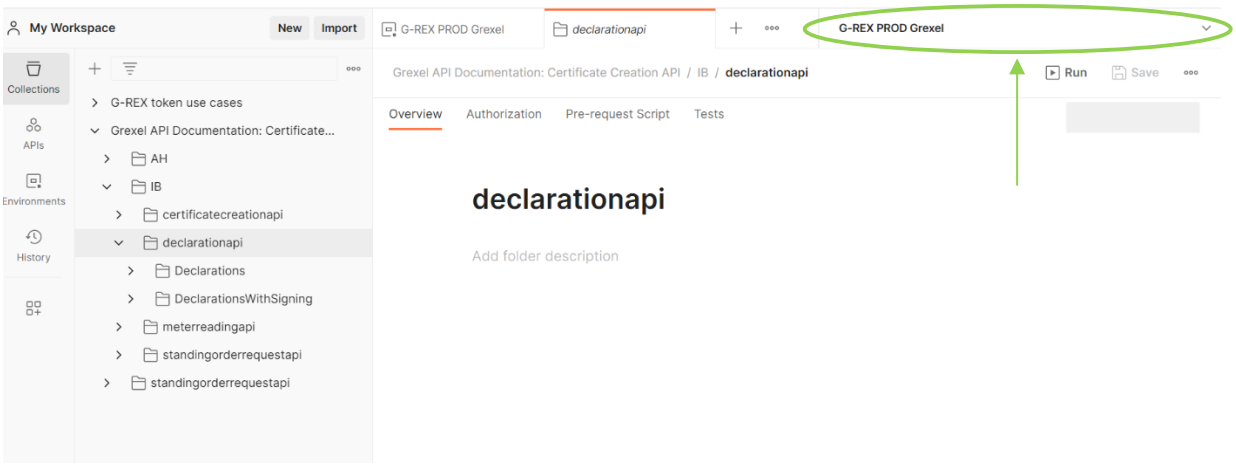

Figure 18 Environment selection

Once the right environment is selected, open the environment quick look from the top right of the screen, and paste the token previously retrieved (see Retrieving tokens) into the **Value** field of the **Token** Variable (see Figure 19 Pasting the token). Alternatively, you can see the same environment variables by clicking the correct environment in the **Environments** list accessed from the left side menu of Postman (see Figure 19 Pasting the token).

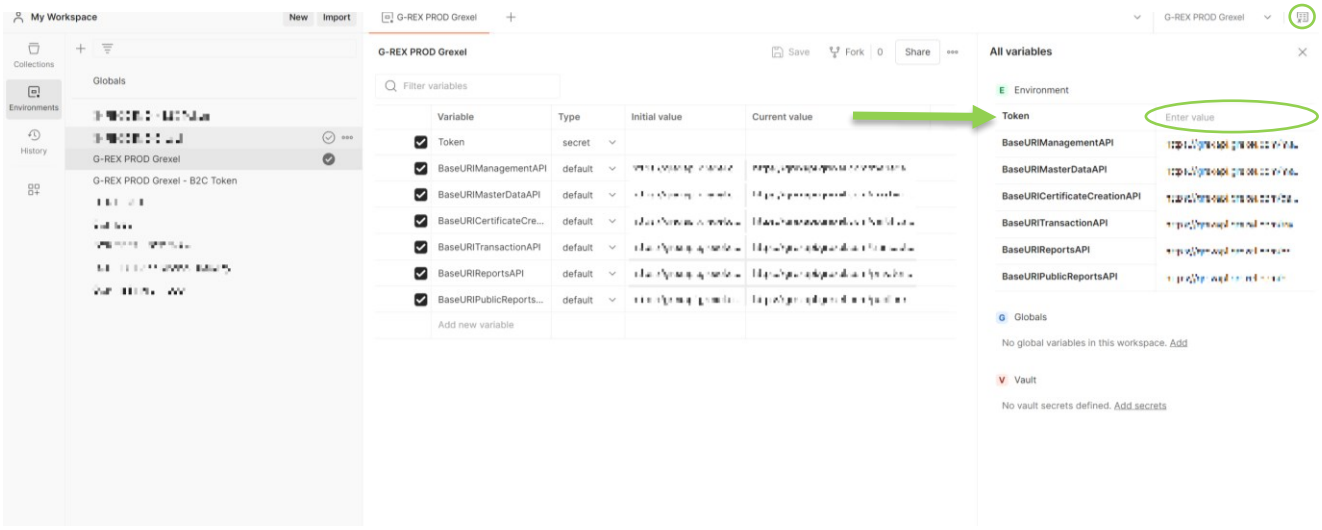

Figure 19 Pasting the token

4. Retrieving Access token from the UX

The access token for making API calls can be retrieved from the User Interface of G-REX via the Developer tools of the browser the application is opened on.

To retrieve the access token from the UX follow these steps:

1. Login to the application (If this is a first-time login, see section 1.4 of the G-REX user manual).
2. Open the developer tools (see Figure 20 below) from the browser's application menu or click Fn+F12 or Ctrl+Shift+i.

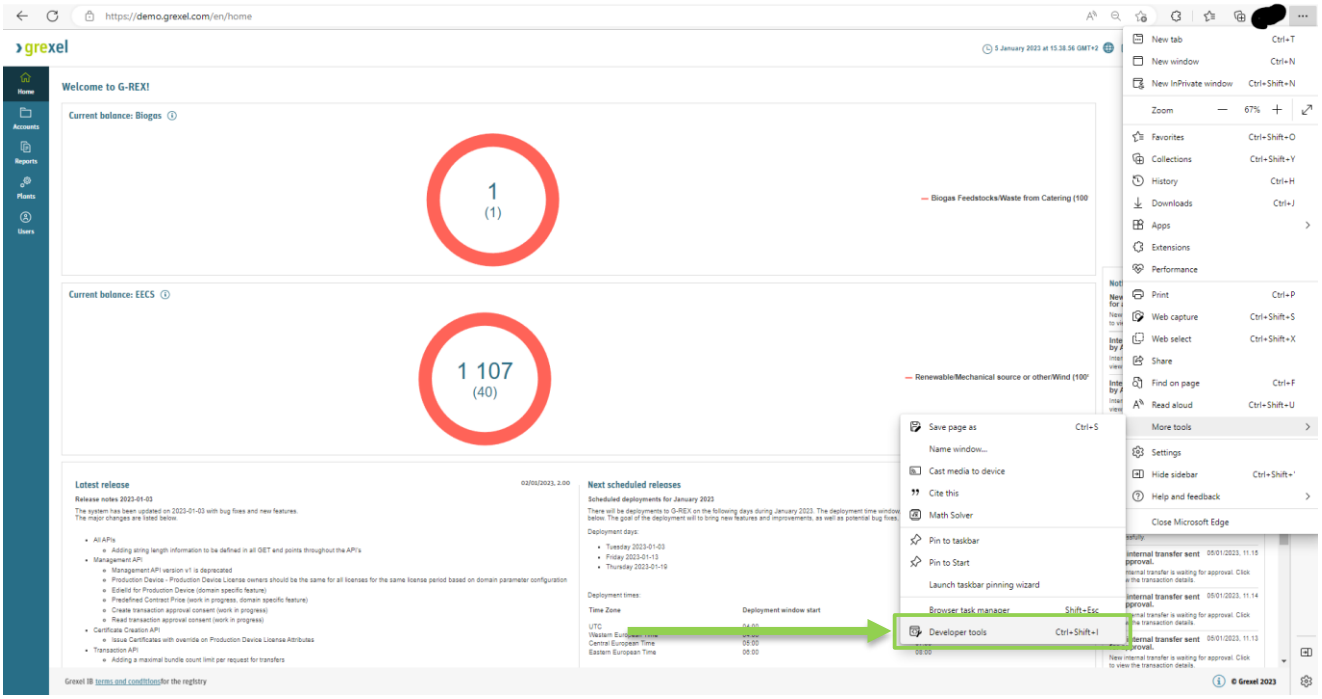

Figure 20 Developer tools

3. Next go to the Application tab and choose Session storage (Depending on the browser, the session storage could be directly under a storage tab).
4. Copy the access_token key from the table key table.
5. And copy the access token value (Highlighted in Figure 21 below)

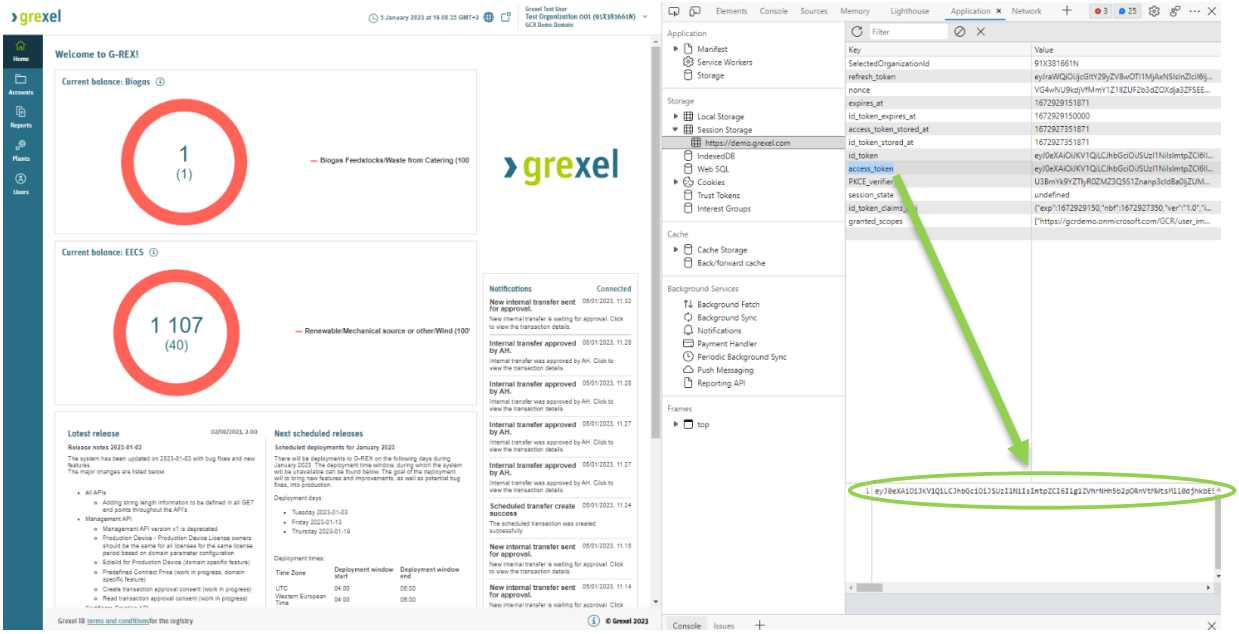

Figure 21 Access token from session storage

6. Add the access token to the postman environment, the token should be copied into the G-REX Environment (**not** the B2C token environment) by pressing the **Variables** icon in the top right of Postman. In the list of variables there is the **Token** variable, paste the copied Access Token into the relevant value field. The field may be empty or contain a previously used token. Regardless, selecting the field and pasting the value into it will be sufficient.

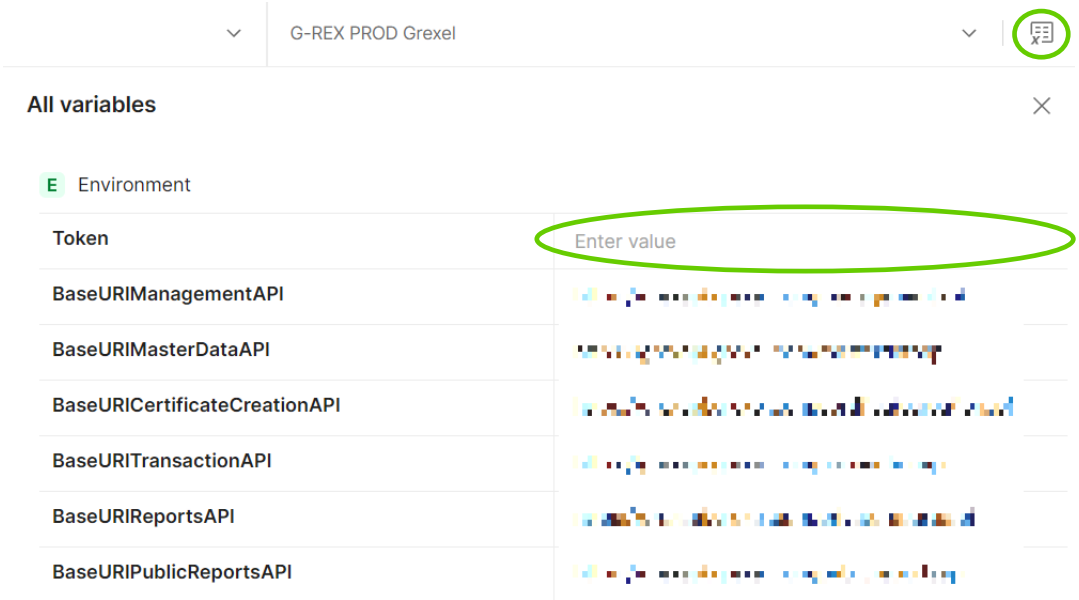

Figure 22 Acces token variable

5. Open API Documentation

After all necessary setup described in previous chapters, to start utilizing G-REX APIs please import the needed collections by accessing the below links and importing those to Postman according to instructions in 2.2.3 Importing postman collections and API documentation via swagger link.

Note: When making calls in Postman, please make sure the call is NOT generated for the B2C token environment but instead from the G-REX PROD/DEMO environment.

G-REX Open API Documentation can be accessed through the links below:

Certificate Creation API: https://eexfranceapi.grexel.com/certificatecreationapi/v1/

Management API: https://eexfranceapi.grexel.com/managementapi/v2/

Master Data API: https://eexfranceapi.grexel.com/masterdataapi/v2/

Private Reports API: https://eexfranceapi.grexel.com/privatereportsapi/v1/

Public Reports API: https://eexfranceapi.grexel.com/publicreportsapi/v1/

Transaction API: https://eexfranceapi.grexel.com/transactionapi/v1/

For G-REX DEMO environments:

Certificate Creation API: https://eexfrancedemoapi.grexel.com/certificatecreationapi/v1/

Management API: https://eexfrancedemoapi.grexel.com/managementapi/v2/

Master Data API: https://eexfrancedemoapi.grexel.com/masterdataapi/v2/

Private Reports API: https://eexfrancedemoapi.grexel.com/privatereportsapi/v1/

Public Reports API: https://eexfrancedemoapi.grexel.com/publicreportsapi/v1/

Transaction API: https://eexfrancedemoapi.grexel.com/transactionapi/v1/

5.1 Referring to the OpenAPI Documentation

5.1.1 Landing page

All the G-REX API links in the previous section follow the same format. The landing page of the Management API landing page is shown in the image below, and its format is identical to that of all G-REX APIs. At the top of the page, a brief description of the API functionality is provided, along with links to Grexel's Terms of service, website, and support portal.

Below this, the user can see the API servers, which do not need to be changed.

The remainder of the landing page shows the sub-categories of API requests, which can be expanded by pressing the arrow on the right side of the page.

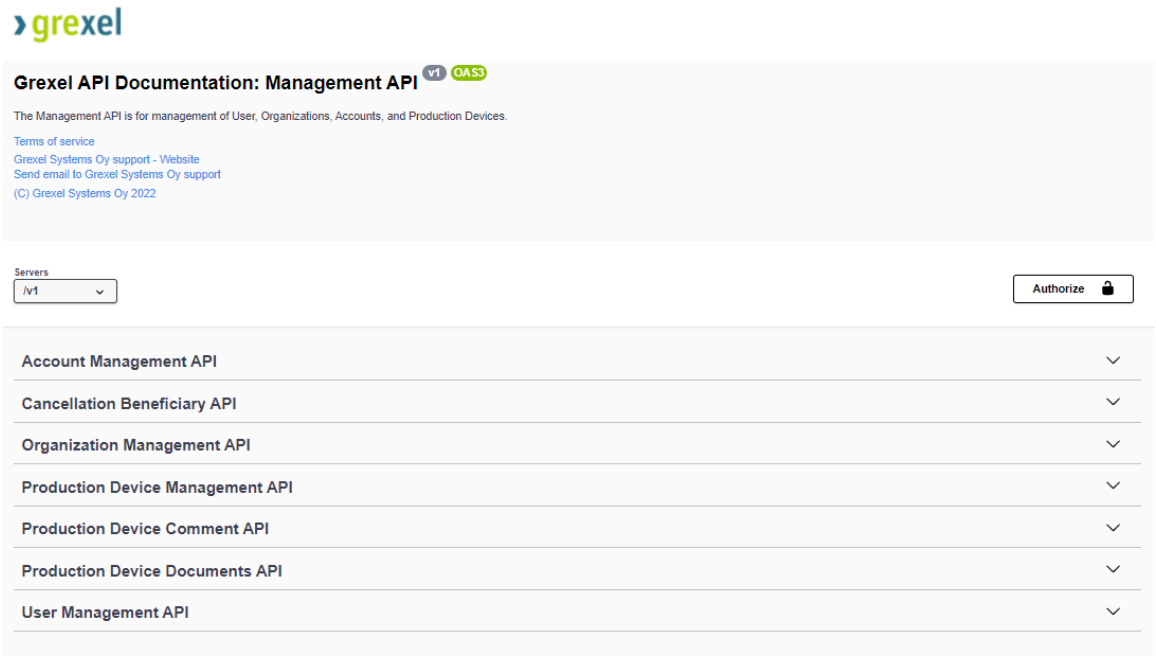

Figure 23 Management API landing page

5.1.2 API sub-categories

The image below shows an example of an expanded sub-category, with the API requests belonging to the sub-category shown below.

Figure 24 API sub-categories

API requests in G-REX fall under the following types:

- **GET**: Used for retrieving information from the system
- **PATCH**: Used for amending existing information in the system
- **POST**: Used for creating new information to the system
- **PUT**: Used for inputting new and/or amending existing information in the system
- **DELETE**: Used for removing information from the system

The type of request is denoted by the color-coded box on the left side of each request line. The request line also includes a description of the function of each request. On the right of each line is a downwards arrow, which can be used to expand the request.

5.1.3 API Requests

An expanded API request is intended to provide all the information needed for submitting a request through postman with G-REX APIs.

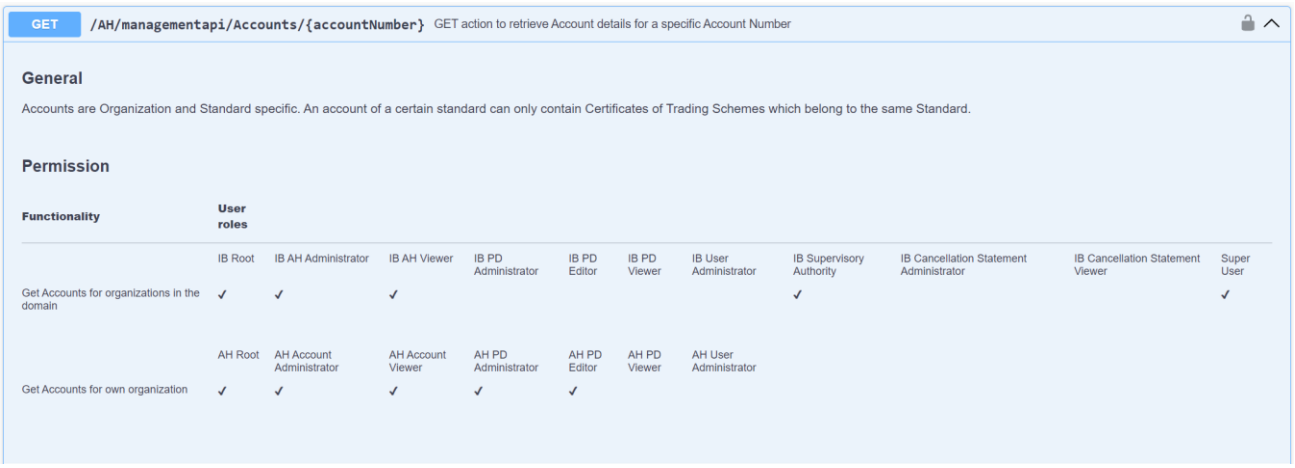

Figure 25 API requests

The image above shows the first part of the API request. The “General” sub-heading contains generic information which should be kept in mind when making a request. This is followed by the “Permission” sub-heading which indicates which type of G-REX users can submit the request. Finally, “Validation and responses” explains the checks which take place in the API and the expected result of a successful request.

The API request form continues by listing any required parameters for submitting the request. These parameters vary for each request. Often, no parameters are required to be submitted. In the image above, the GET Account request is shown, where “accountNumber” is a required parameter. At the bottom of the above image, the user can see an example of the schema of a submitted request. This shows the required input format of the request.

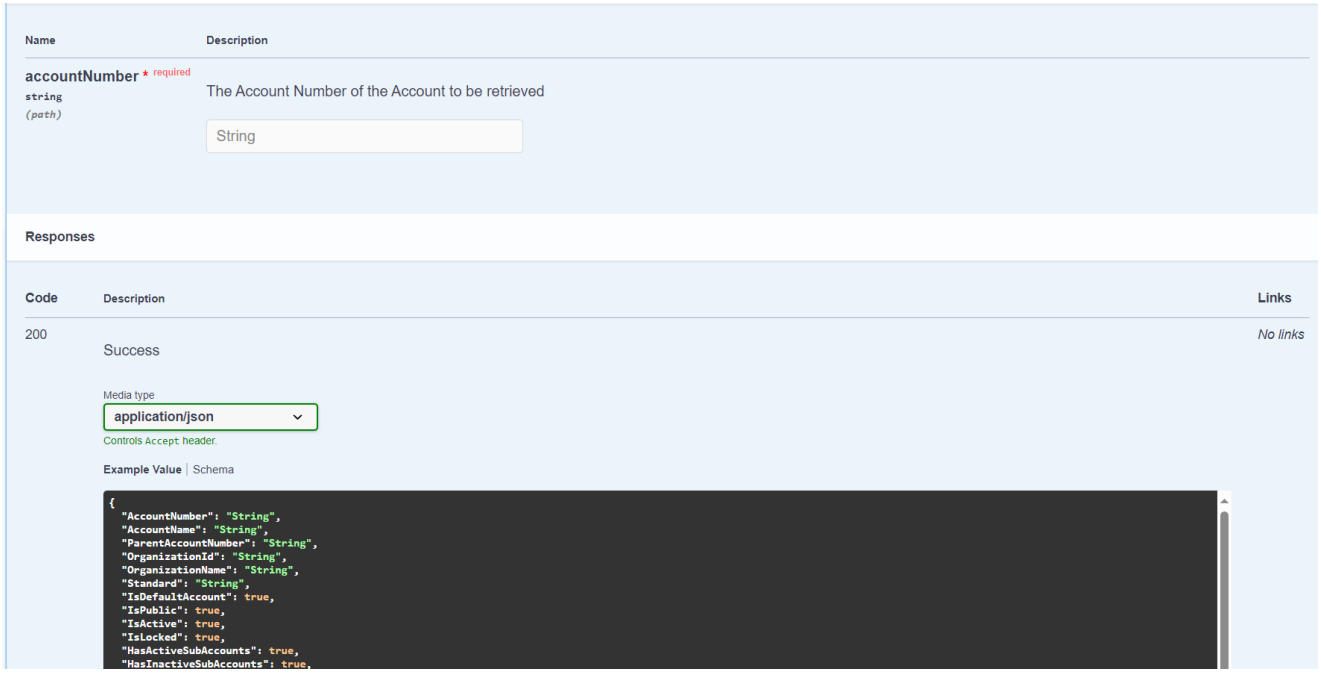

Figure 26 Get Account request example

5.1.4 Making an API request

Note: A user can build their own requests. However, there is no need as the importable collections already cover the possible requests that can be made via API.

To make an API call in Postman, follow the actions listed in sections 2.1.2 - 3.1 in this document.

Once you have these set up and have authorized the user, switch to the environment you have created by choosing it from the list of environments (See Figure 27 below).

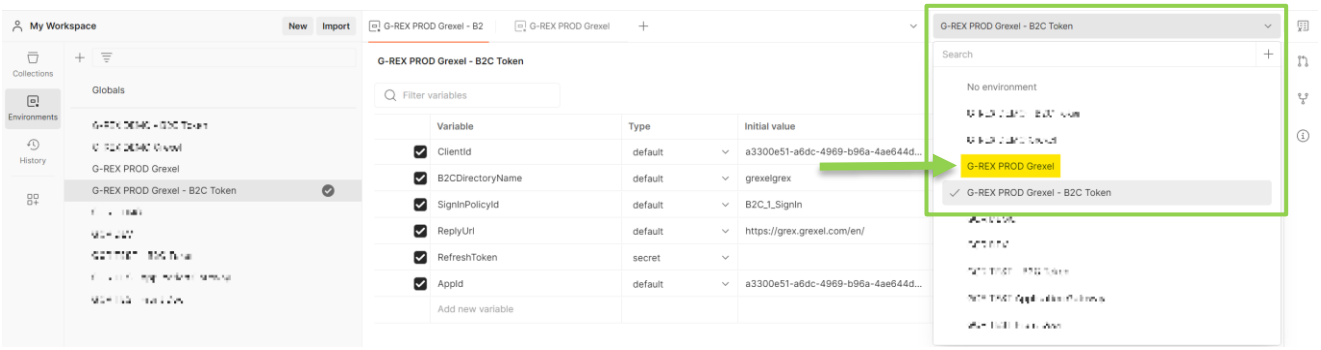

Figure 27 Selecting environments in Postman

Then, press the Plus button (See Figure 28 below) for sending a new request.

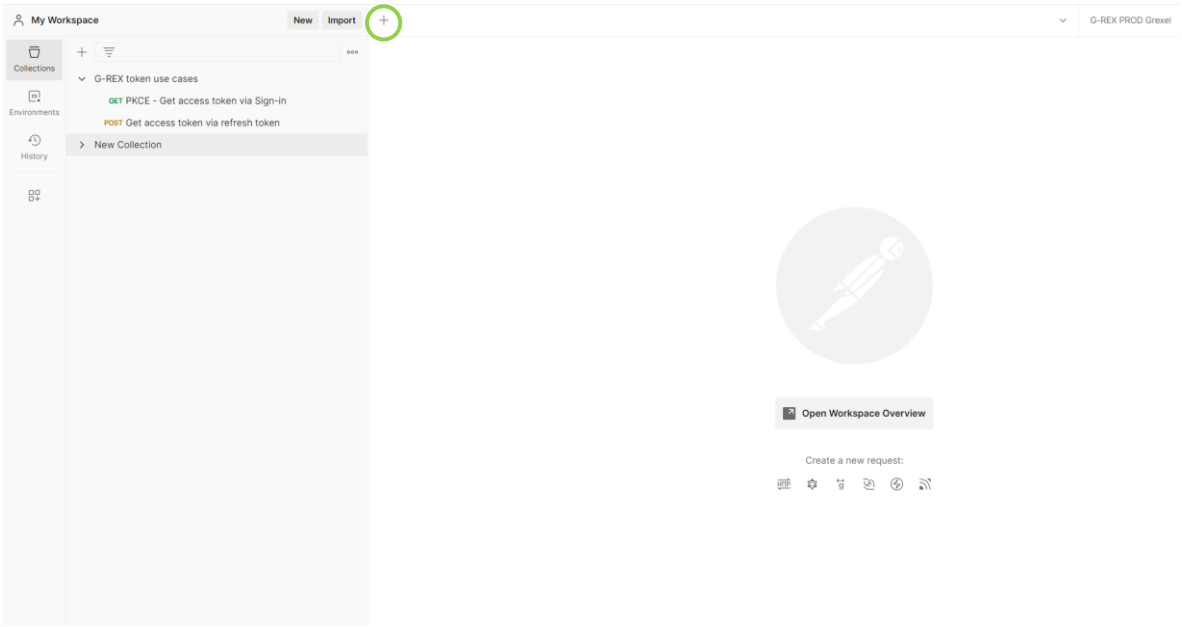

Figure 28 New collection

Next, press the link to the correct documentation and Copy the URL of the API you wish to make the call to (from section 6 in this document), and:

1. Paste it in the enter request URL field (for this example we would be using the Management API URL, Highlighted in yellow)
2. Change the request type if it is different from the API call you wish to make (see oval)
3. Add the subcategory (highlighted green) to the end of the URL.
4. Add filters to your query that you would like to see in the query parameters (see rectangle)
5. Press send (see arrow).

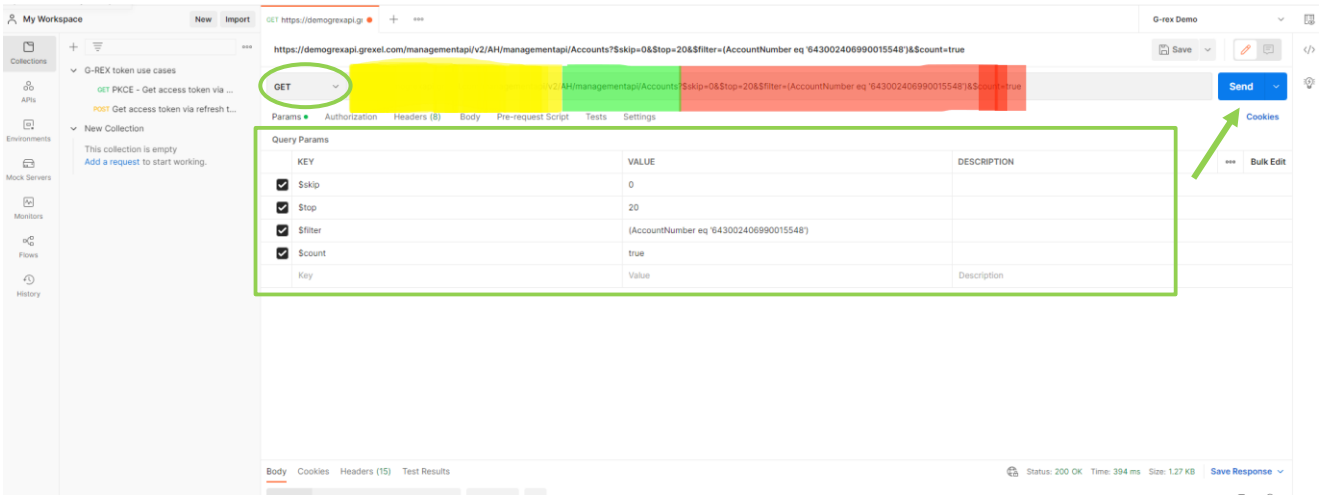

Figure 29 Sending an API request

From the image below one can see a successful response to the request sent and the values of the response.

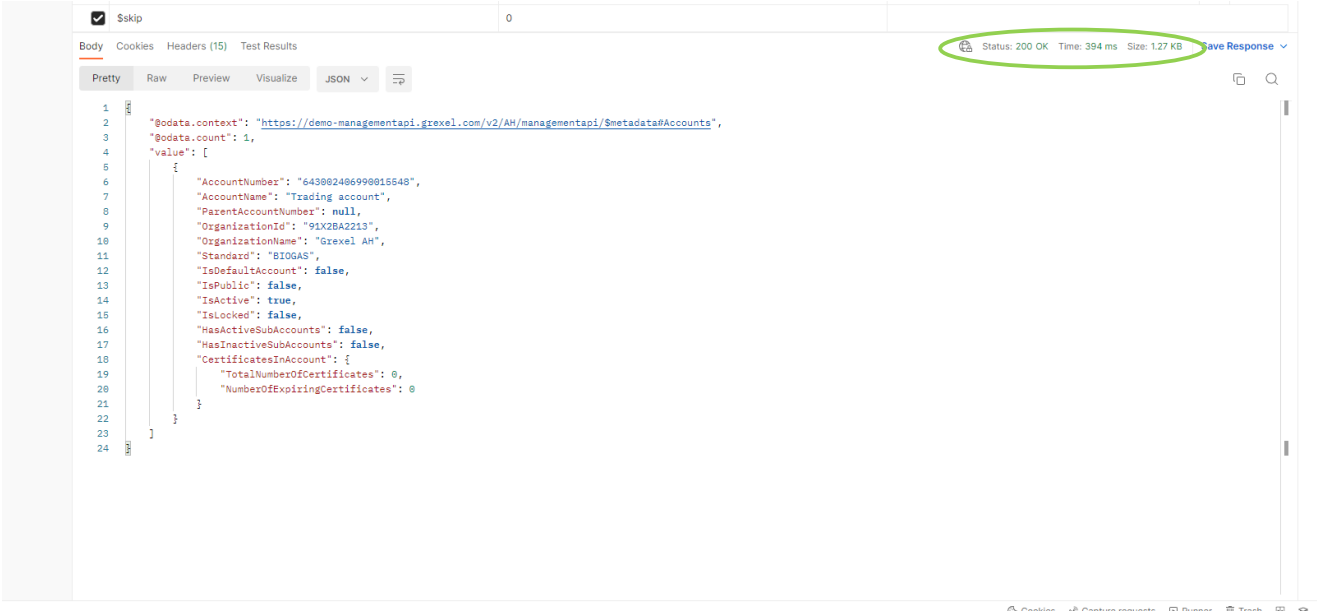

Figure 30 Successful reponse to an API request

Once a request has been sent, a “Server response” will be returned, which will indicate whether the request was submitted to G-REX successfully.

At the bottom of the expanded API request field, the User is provided with a list of all possible response codes for the request. In the image below, an example of a successfully submitted request is shown, where the response code is 200, as well as possible error codes with code 400 and 500. In the expanded API request, all other possible codes which can be returned by a request are shown and explained similarly to those in the below image.

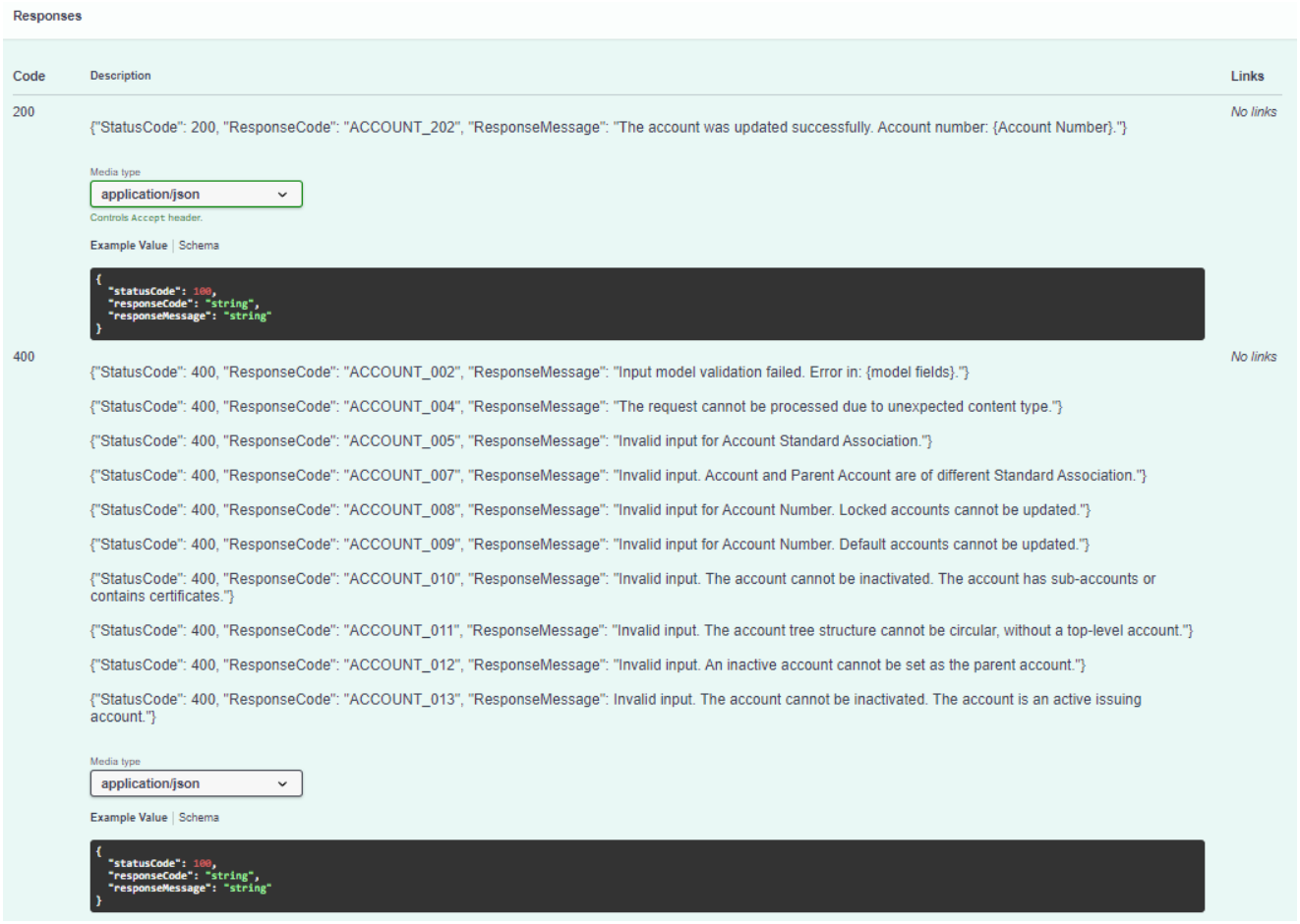

Figure 31 Response codes to API requests

Note: If a request is unsuccessful with the HTTP response '403: Forbidden' message, this can often indicate that some element of the request has been blocked due to invalid data input which is triggering firewall blockings in the system. If possible, the request can be reattempted with problematic data removed. Data which triggers these blockings varies widely but is most often special characters (e.g. "/") or combinations of special characters (e.g. "https://").

6. PKCE specification for development of own connection

Authentication - Proof Key for Code Exchange (PKCE) flow The GCR authentication utilizes the Microsoft identity platform and OAuth 2.0 authorization code flow, using the Microsoft identity platform implementation of OAuth 2.0 and Open ID Connect. The authentication protocol used is the Authorization Code Grant, specifically the Proof Key for Code Exchange (PKCE).

The following authorization protocols are prohibited due to security concerns:

- Implicit code flow is not to be used in production environments to due to security concerns.
- Resource Owner Password Credentials (ROPC) is not to be used due to security concerns.

For more information on the security of the Authentication: https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/05-Testing_for_OAuth_Weaknesses

The Proof Key for Code Exchange (PKCE) flow is explained here below.

6.1 How to get the access token and refresh token for API use

1. The client requests a token from the authorization host (in practise this is the Azure B2C authorization endpoint).

2. The client requested a specific login flow (in Azure B2C terms this is the User flow), which the end user completes. This includes:

a. Login in via Username and Password

b. Multifactor verification (via SMS/Phone Call). For SMS this is just a small pass code that the end user enters the login flow to verify that the end user requesting the token is the intended user.

3. If steps 1-2 are successful, the host creates a secure connection with the client and issues an authorization token for the request

4. With the authorization token in step 3 the client sends the request to get the access token and refresh token from the authorization host (this is done to the Azure B2C token endpoint)

5. The authorization host returns the tokens (access token and refresh token)

This is step 1-5 in the following picture:

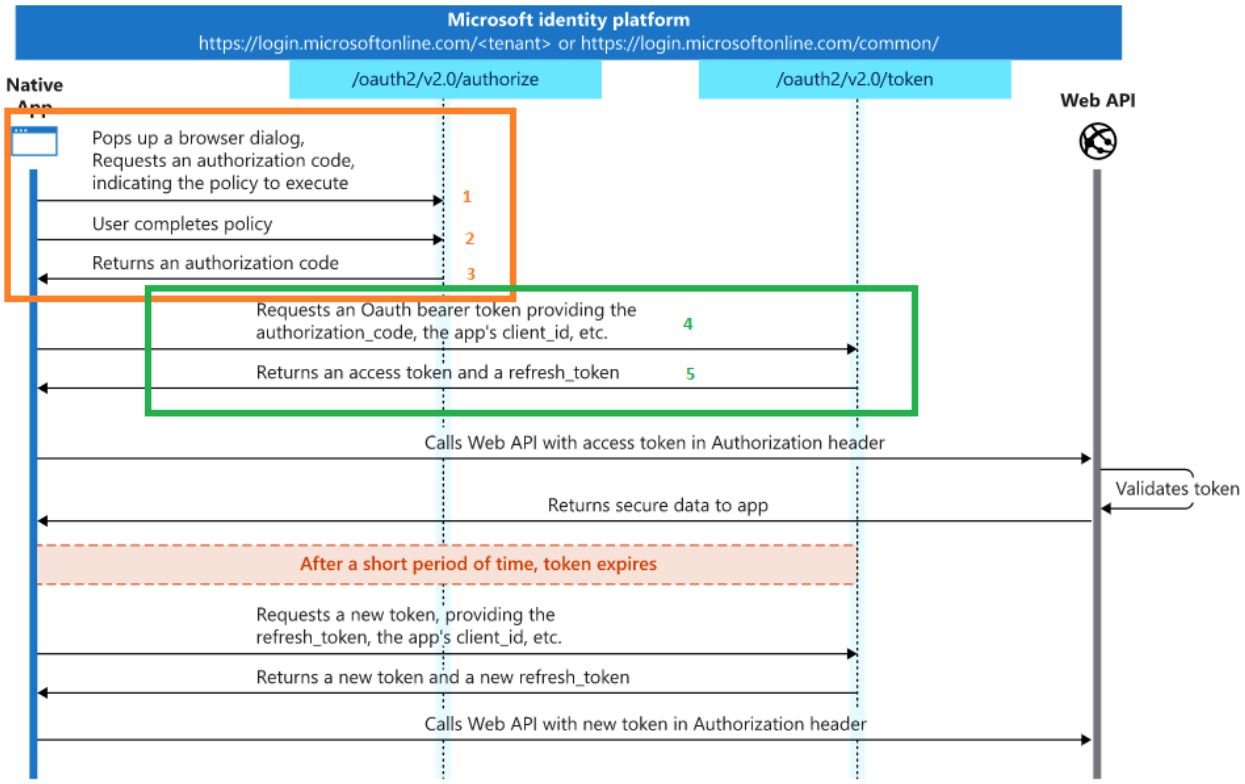

Figure 32

6.2 How to get data from the API

1. The client sends the API call (a HTTPS GET/POST/PATCH/PUT/DELTE request), with the access token in an authorization header, to the G-REX API.

2. The token is validated by the G-REX API, if not valid the call is rejected

a. Token is validated towards the authorization endpoint.

b. Token payload is verified and the G-REX system check that the user exists and have access to request the relevant data.

3. The G-REX API returns secure data to the client if the token and user is valid. If the token or user is not valid the client will get a 401-error code (401 - Unauthorized).

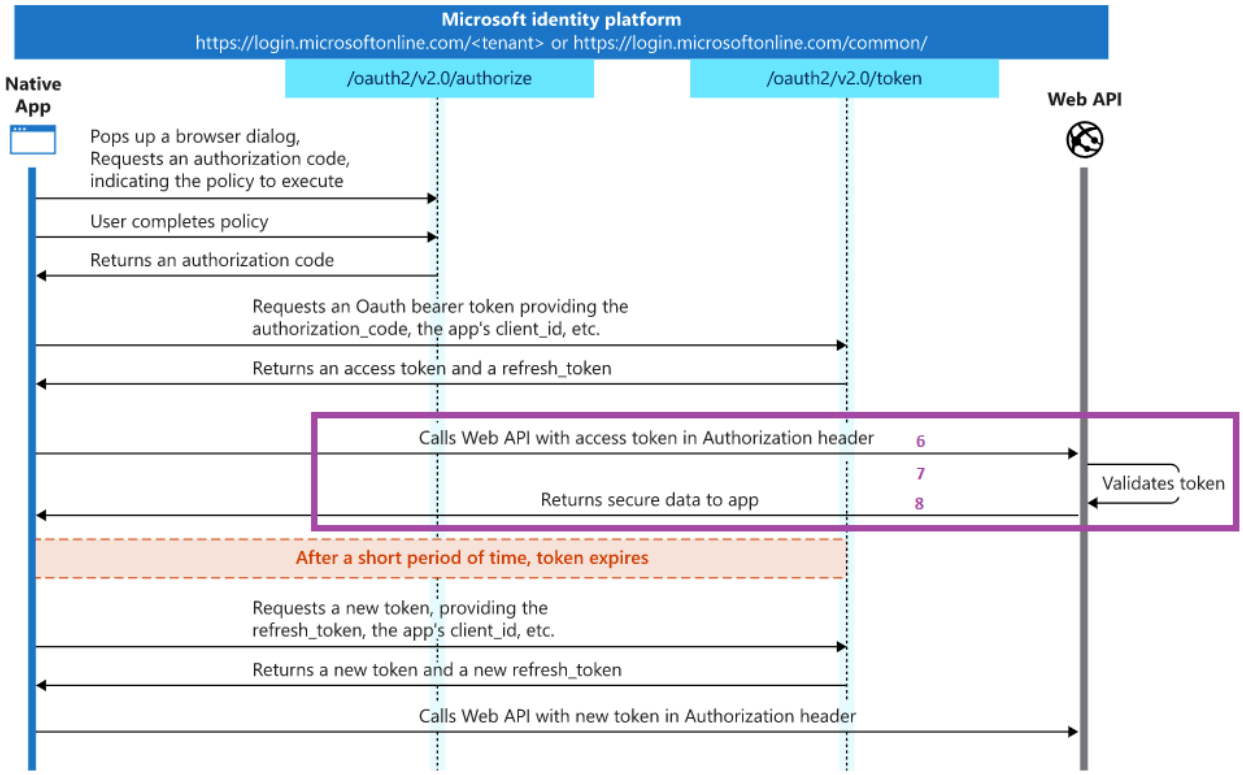

Figure 33

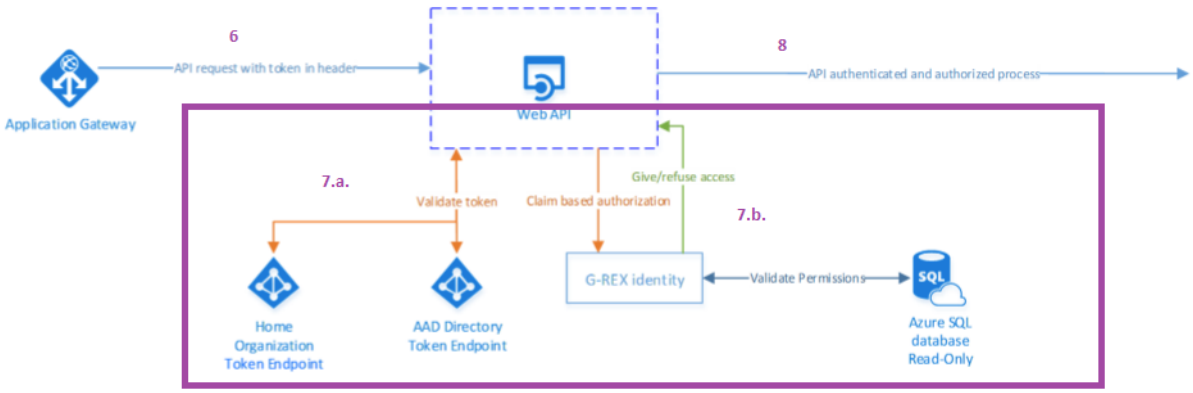

Figure 34

After a configured time (N minutes) the access token will expire. The expiry time is determined by the B2C configuration set in the Sign-In policy.

6.3 How to get a new access, and refresh, token:

At any point the following can be done if there is a valid refresh token.

A valid refresh token means that the lifetime of the refresh token has not expired, or there is no bounded token lifetime that prohibits the refresh – this bounded lifetime can be set to never expire. The refresh token expiry time is determined by the B2C configuration set in the Sign-In policy for mobile and desktop applications. For single page applications the refresh token lifetime is set by default to 24 hours:

https://docs.microsoft.com/en-us/azure/active-directory-b2c/configure-tokens?pivots=b2c-custom-policy#token-lifetime-behavior

A. The client can do a POST request to the authorization host to get a new access and refresh, with the existing refresh token in the request.

B. The authorization host will return a new access token and refresh token.

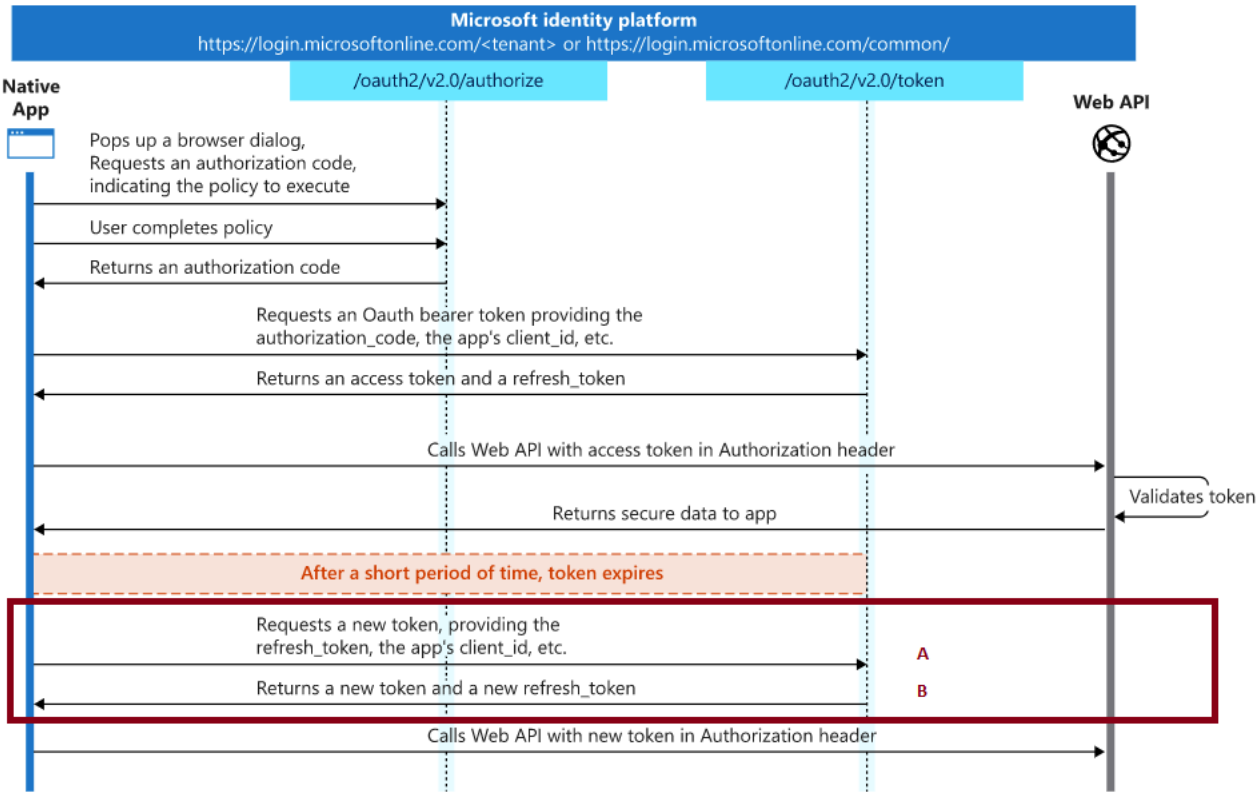

Figure 35

6.4 Microsoft documentation / references / resources:

- https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-auth-code-flow

- https://docs.microsoft.com/en-us/azure/active-directory/develop/tutorial-v2-angular-auth-code
- https://docs.microsoft.com/en-us/azure/active-directory/develop/tutorial-v2-javascript-auth-code
- https://docs.microsoft.com/en-us/azure/active-directory-b2c/configure-tokens?pivots=b2c-custom-policy#token-lifetime-behavior

7. Appendix 1: G-REX Production environment for Postman


```
{
	"id": "4d371554-fb02-4c9a-b905-60dc3f9ab619",
	"name": "G-REX PROD EEX France",
	"values": [
		{
			"key": "Token",
			"value": "",
			"type": "secret",
			"enabled": true
		},
		{
			"key": "BaseURIManagementAPI",
			"value": "https://eexfranceapi.grexel.com/managementapi",
			"enabled": true
		},
		{
			"key": "BaseURIMasterDataAPI",
			"value": "https://eexfranceapi.grexel.com/masterdataapi",
			"enabled": true
		},
		{
			"key": "BaseURICertificateCreationAPI",
			"value": "https://eexfranceapi.grexel.com/certificatecreationapi",
			"enabled": true
		},
		{
			"key": "BaseURIReportsAPI",
			"value": "https://eexfranceapi.grexel.com/privatereportsapi",
			"enabled": true
		},
		{
			"key": "BaseURIPublicReportsAPI",
			"value": "https://eexfranceapi.grexel.com/publicreportsapi",
			"enabled": true
		},
		{
			"key": "BaseURITransactionAPI",
			"value": "https://eexfranceapi.grexel.com/transactionapi",
			"enabled": true
		}
	],
	"_postman_variable_scope": "environment",
	"_postman_exported_at": "2023-09-14T18:31:43.111Z",
	"_postman_exported_using": "Postman/10.17.3"
}
```

8. Appendix 2: G-REX B2C token environment for Postman


```
{
	"id": " bca8a787-322b-4cb6-82fe-0a95f00e3724",
	"name": " EEX FRANCE Grexel - B2C Token ",
	"values": [
		{
			"key": "ClientId",
			"value": "57652e54-8e9d-49b4-bcb8-402c95e8c020",
			"enabled": true
		},
		{
			"key": "B2CDirectoryName",
			"value": "gcrfr",
			"enabled": true
		},
		{
			"key": "SignInPolicyId",
			"value": "B2C_1_SignIn",
			"enabled": true
		},
		{
			"key": "AppId",
			"value": "57652e54-8e9d-49b4-bcb8-402c95e8c020",
			"enabled": true
		},
		{
			"key": "ReplyUrl",
			"value": "https://gcrfr.b2clogin.com/oauth2/nativeclient",
			"enabled": true
		},
		{
			"key": "RefreshToken",
			"value": "",
			"type": "secret",
			"enabled": true
		}
	],
	"_postman_variable_scope": "environment",
	"_postman_exported_at": "2025-07-29T14:26:43.579Z",
	"_postman_exported_using": "Postman/11.56.0"
}
```

9. Appendix 3: G-REX token use cases for Postman


```
{
	"info": {
		"_postman_id": "ca02852b-97b5-4a49-9edc-7b400b69ea74",
		"name": "G-REX token use cases",
		"schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
	},
	"item": [
		{
			"name": "PKCE - Get access token via Sign-in",
			"request": {
				"auth": {
					"type": "oauth2",
					"oauth2": [
						{
							"key": "scope",
							"value": "https://{{B2CDirectoryName}}.onmicrosoft.com/{{AppId}}/user_impersonation openid offline_access",
							"type": "string"
						},
						{
							"key": "grant_type",
							"value": "authorization_code_with_pkce",
							"type": "string"
						},
						{
							"key": "redirect_uri",
							"value": "{{ReplyUrl}}",
							"type": "string"
						},
						{
							"key": "useBrowser",
							"value": false,
							"type": "boolean"
						},
						{
							"key": "authUrl",
							"value": "https://{{B2CDirectoryName}}.b2clogin.com/{{B2CDirectoryName}}.onmicrosoft.com/{{SignInPolicyId}}/oauth2/v2.0/authorize",
```

```
                                "type": "string"
                            },
                            {
                                "key": "accessTokenUrl",
                                "value": "https://{{B2CDirectoryName}}.b2clogin.com/{{B2CDirectoryName}}.onmicrosoft.com/{{SignInPolicyId}}/oauth2/v2.0/token",
                                "type": "string"
                            },
                            {
                                "key": "clientId",
                                "value": "{{ClientId}}",
                                "type": "string"
                            },
                            {
                                "key": "client_authentication",
                                "value": "header",
                                "type": "string"
                            },
                            {
                                "key": "clientSecret",
                                "value": "",
                                "type": "string"
                            },
                            {
                                "key": "addTokenTo",
                                "value": "header",
                                "type": "string"
                            }
                        ]
                    },
                    "method": "GET",
                    "header": [],
                    "url": {
                        "raw": ""
                    }
                },
                "response": []
            },
            {
                "name": "Get access token via refresh token",
```

```
"request": {
        "method": "POST",
        "header": [
            {
                "key": "Content-Type",
                "value": "application/x-www-form-urlencoded",
                "type": "text"
            }
        ],
        "body": {
            "mode": "urlencoded",
            "urlencoded": [
                {
                    "key": "grant_type",
                    "value": "refresh_token",
                    "type": "text"
                },
                {
                    "key": "client_id",
                    "value": "{{ClientId}}",
                    "type": "text"
                },
                {
                    "key": "scope",
                    "value": "https://{{B2CDirectoryName}}.onmicrosoft.com/{{AppId}}/user_impersonation openid offline_access",
                    "type": "text"
                },
                {
                    "key": "offline_access",
                    "value": "",
                    "type": "text",
                    "disabled": true
                },
                {
                    "key": "refresh_token",
                    "value": "{{RefreshToken}}",
                    "type": "text"
                },
                {
```

```
                                        "key": "redirect_uri",
                                        "value": "{{ReplyUrl}}",
                                        "type": "text"
                                    }
                                ]
                            },
                            "url": {
                                "raw":
"https://{{B2CDirectoryName}}.b2clogin.com/{{B2CDirectoryName}}.onmicrosoft.com/{{SignInPolicyId}}/oauth2/v2.0/token",
                                "protocol": "https",
                                "host": [
                                    "{{B2CDirectoryName}}",
                                    "b2clogin",
                                    "com"
                                ],
                                "path": [
                                    "{{B2CDirectoryName}}.onmicrosoft.com",
                                    "{{SignInPolicyId}}",
                                    "oauth2",
                                    "v2.0",
                                    "token"
                                ]
                            }
                        },
                        "response": []
                    }
                ]
}
```

10. Appendix 4: G-REX DEMO environment for Postman


```
{
	"id": "3b6fa718-adeb-4c4e-b26f-abe19c278ce5",
	"name": "G-REX DEMO EEX France",
	"values": [
		{
			"key": "Token",
			"value": "",
			"type": "secret",
			"enabled": true
		},
		{
			"key": "BaseURIManagementAPI",
			"value": "https://eexfrancedemoapi.grexel.com/managementapi",
			"enabled": true
		},
		{
			"key": "BaseURIMasterDataAPI",
			"value": "https://eexfrancedemoapi.grexel.com/masterdataapi",
			"enabled": true
		},
		{
			"key": "BaseURICertificateCreationAPI",
			"value": "https://eexfrancedemoapi.grexel.com/certificatecreationapi",
			"enabled": true
		},
		{
			"key": "BaseURIReportsAPI",
			"value": "https://eexfrancedemoapi.grexel.com/privatereportsapi",
			"enabled": true
		},
		{
			"key": "BaseURIPublicReportsAPI",
			"value": "https://eexfrancedemoapi.grexel.com/publicreportsapi",
			"enabled": true
		},
		{
			"key": "BaseURITransactionAPI",
			"value": "https://eexfrancedemoapi.grexel.com/transactionapi",
			"enabled": true
		}
	],
	"_postman_variable_scope": "environment",
```

```
        "_postman_exported_at": "2023-09-14T18:32:26.621Z",
        "_postman_exported_using": "Postman/10.17.3"
}
```

11. Appendix 5: G-REX DEMO - B2C token environment


```
{
	"id": "e6491240-4dd9-4113-abe1-0f91d53aecc7",
	"name": "EEX FRANCE DEMO Grexel - B2C Token",
	"values": [
		{
			"key": "ClientId",
			"value": "9e818394-185c-47e7-8b93-714714bfcf8f",
			"enabled": true
		},
		{
			"key": "B2CDirectoryName",
			"value": "gcrfrdemo",
			"enabled": true
		},
		{
			"key": "SignInPolicyId",
			"value": "B2C_1_SignIn",
			"enabled": true
		},
		{
			"key": "AppId",
			"value": "GCR",
			"enabled": true
		},
		{
			"key": "ReplyUrl",
			"value": "https://gcrfrdemo.b2clogin.com/oauth2/nativeclient",
			"enabled": true
		},
		{
			"key": "RefreshToken",
			"value": "",
			"type": "secret",
			"enabled": true
		}
	],
	"_postman_variable_scope": "environment",
	"_postman_exported_at": "2023-03-27T09:07:51.592Z",
	"_postman_exported_using": "Postman/10.12.0"
}
```