ATIPXpress

OPEXUS

API User Manual

v11.4.0

October 2023

AX 11.4.0 API User Manual

Notice of Rights

Notice of Liability

The information in this publication is believed to be accurate and reliable. However, the information is distributed by the publisher (AINS, LLC.) on an "As Is" basis without warranty for its use, or for any infringements of patents or other rights of third parties resulting from its use.

While every precaution has been taken in the preparation of this publication, neither the author (or authors) nor the publisher will have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused, directly or indirectly, by the information contained in this publication or by the computer software and hardware products described in it.

Notice of Trademarks

The publisher's company name, company logo, company patents, and company proprietary products are trademarks or registered trademarks of the publisher: AINS, LLC. All other trademarks or registered trademarks are the property of their respective owners.

Non-Disclosure Statement

This document's contents are confidential and proprietary to AINS, LLC. This document cannot be released publicly or outside the purchasing agency without prior written permission from AINS, LLC.

Images in this manual are used as examples and may contain data and versioning that may not be consistent with your version of the application or information in your environment.

Additional Notice

Information in this documentation is subject to change without notice and does not represent a commitment on the part of AINS, LLC.

Notwithstanding any of the foregoing, if this document was produced as a Deliverable or other work for hire under a contract on behalf of a U.S. Government end user, the terms and conditions of that contract shall apply in the event of a conflict.

Contents

1 Introduction

1.1 ATIPXpress API

We've created the ATIPXpress API to allow for integration with external data monitoring applications. This is a standalone API that is included in the ATIPXpress installation package. This API must be installed separately from the ATIPXpress application, and users of the API must be authenticated application users with permissions to access the data being queried.

1.2 About this Manual

This manual covers all actions associated with the API, from installation, to authenticating API users, and instructions on using the included API methods. The following major topics are included:

- *API Installation*: Installing the API
- *Accessing the ATIPXpress API*: Token creation, accessing the API, and viewing logs
- *Request Methods*: Methods to query Request data
- *Audit Methods*: Methods to query user actions and logins

2 API Installation

2.1 Installation Prerequisites

The following prerequisites must be fulfilled to use the API:

- The API is included with ATIPXpress version 11.4.0, and you must be on this latest application version to use the API.
- You must install .Net Core Hosting Bundle v.7.0.12: https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-7.0.12-windows-hosting-bundle-installer

2.2 Unzip the Installation Package

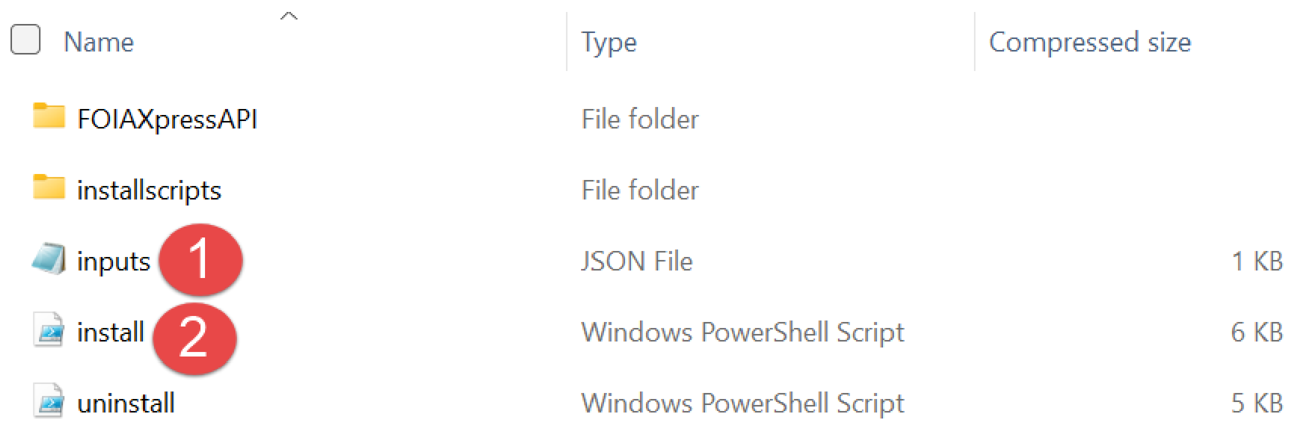
The API installation package is included in the ATIPXpress 11.4.0 application files. After downloading the package, follow the steps below to unzip the file:

1. Locate the downloaded .zip file, then right click it and select **Properties**.
2. On the *General* tab, select **Unblock**, then click **Apply**.
3. Next, unzip the contents of the folder. The contents of the package appear as shown in the following example:

Name	Type	Compressed size
FOIAXpressAPI	File folder	
installscripts	File folder	
inputs	JSON File	1 KB
install	Windows PowerShell Script	6 KB
uninstall	Windows PowerShell Script	5 KB

2.3 About the Installation Package

There are two primary files you'll interact with during installation:



1. *Inputs.json*: Drives the installation of the API components. The file is in a standard json format and needs to remain a valid json file for successful installation.
2. *Install.ps1*: The executable file you will run to deploy the installation package with the parameters configured in the Inputs.json file.

2.4 Inputs.Json Settings for Installation

The Inputs.json file located in the root directory of the extracted deployment package zip file drives the installation of the different components of the ATIPXpress API. The file is in a standard json format and needs to remain a valid json file post updates prior to the installation.

The settings are grouped into sections, each addressing settings for your installation. This section describes each part of the Inputs.Json file.

Use the “InstallType” section to determine if this is a new installation, or an upgrade for an existing system:

```
{
  "inputs": {
      "InstallType": {
        "New": "Y",
        "Upgrade": "N"
      },
```

Setting	Description
New	Determine whether this is a new installation. For new installations this should be a “Y” value.

Setting	Description
Upgrade	Determine whether this is an upgrade for an existing application. For new installations this should be a "N" value.

Use the "API" section has settings related to the Collaboration Portal application and scheduler installations. A value must be set for all values in this section to successfully install the application:

(!!) Notes:

- **For the SSLCert and AppPool fields, use the same values provided for ATIPXpress installation**
- **Any files paths specified must have two backslashes where a path in windows explorer requires a single backslash.**

```
"API": {
  "Name": "FOIAXpressAPI",
  "InstallLocation": "C:\\Program Files",
  "BackupLocation": "C:\\backup",
  "SSLCert": {
    "SSLPort": "443",
    "CertPath": "<<Certificate Path>>",
    "CertPassword": "<<Certificate Password>>"
  },
  "AppPool": "<<AppPoolName>>"
```

Setting	Description
Name	Name for the API program file.
InstallLocation	Specify a path for the API installation. Note this must be a file path for the .json file the execute.
BackupLocation	Specify a path for the API backup location. Note this must be a file path for the .json file the execute.

Setting	Description
SSLCert	The following lines contain the SSL Certificate information. No data is required on this line. This is the grouping for SSL Certification settings to be used for the installation.
SSLPort	The network port designated for your application's secure communications using SSL.
CertPath	Specify a path where the .pfx certificate file is located. Note this must a file path for the .json file to execute.
CertPassword	Enter the password associated with the certificate linked in the "CertPath" field.
DNSName	Enter the Domain Name System (DNS) name for the domain associated with the application.
AppPool	Enter the name of the application pool associated with your ATIPXpress API.

The "DB" section contains the settings for installing or upgrading your API database. In case of a new installation, an agent account will be created.

(!!) Note: Database details should be the same as used for the ATIPXpress installation.

```
"DB": {
  "DBServer": "<<SQL Server Name>>",
  "DBName": "<<SQL DB Name>>",
  "DBUserId": "<<DB UserId>>",
  "DBUserPassword": "<<DB Password>>"
}
```

Setting	Description
DBServer	Name assigned to the server responsible for storing, retrieving, and managing API data.
DBName	Name given to the specific database used for ATIPXpress API.
DBUserID	Database User ID to be used for this installation. The DBUserId should have admin rights with the ability to create a new database or database object on the specified Database server.
DBUserPassword	Password associated with the account used in the “DBUserID” field.

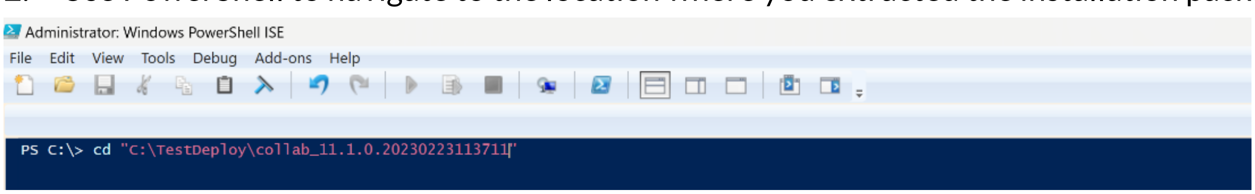
2.5 Install the API

Before you are ready to run the Install.ps1 file, review the steps below to verify you’ve completed all the steps to this point:

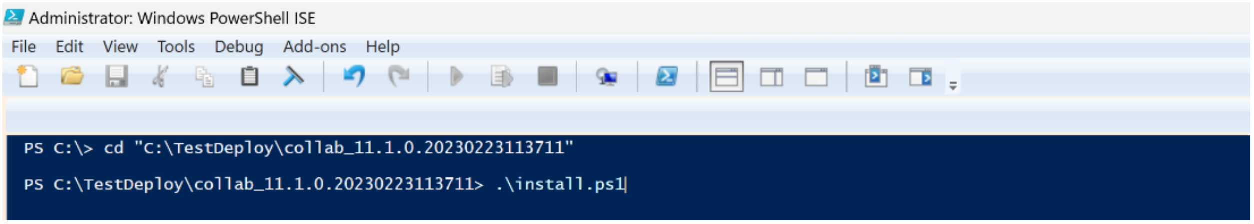
- Make sure you have connectivity to the servers where the software is being installed.
- Download, verify, and extract the API Installation package.
- Configure the Inputs.json file for your installation.

Once the above prerequisites are complete, follow the steps below to deploy the API installation package.

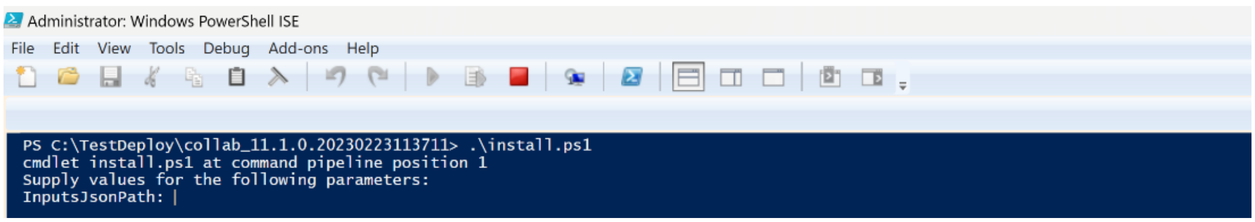
1. Open Windows Powershell ISE.
2. Use PowerShell to navigate to the location where you extracted the installation package:



3. Type .\install.ps1 for either install or upgrade



4. If the inputs.json file is in same location as the extracted installation files, simply click enter to execute the installation. Alternatively, if you have file in different location provide this file location, then execute the script.



5. The installation procedure executes.

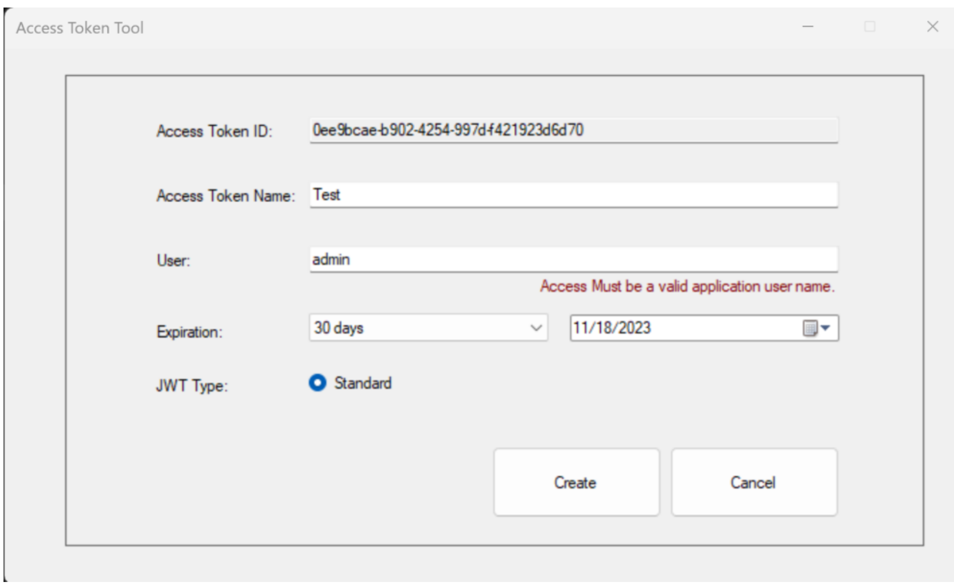
3 Accessing the ATIPXpress API

3.1 Authentication using Access Token Tool

Each API user must be authenticated before they are able to use the API. To provision API access, we've created an Access Token Tool which you'll use to create tokens for each API user. These tokens are time limited and must be provisioned again after expiration. This expiration period is configurable, and we recommend a 30-day expiration period for these tokens.

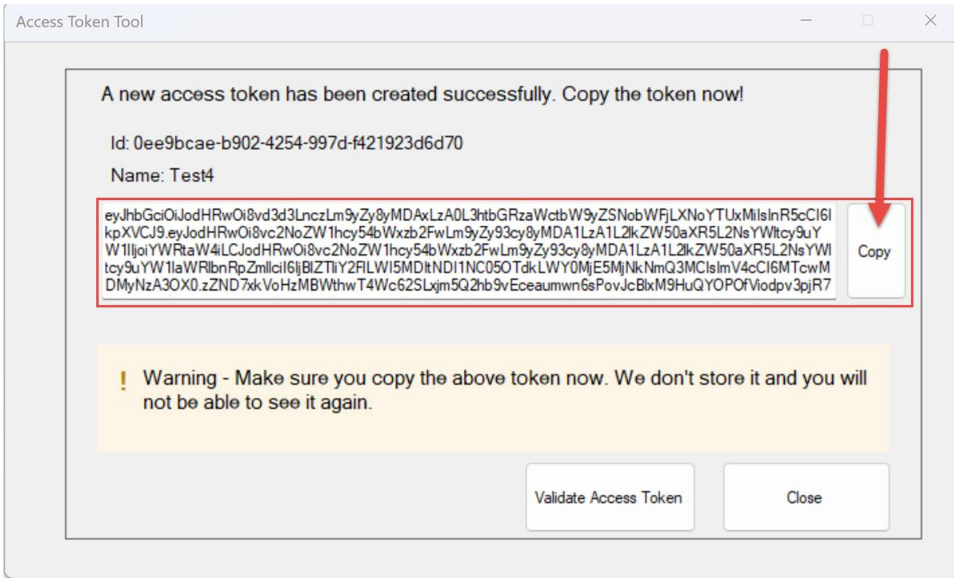
Follow the steps below to create access tokens for your API users:

Within the ATIPXpress application files, run the **Access Token Tool**. The tool appears as shown in the following example:



1. You'll use the fields on this screen to configure tokens for your API users. These are described below:
    a. The *AccessToken ID* is a random GUID which is assigned to each token. This is prepopulated.
    b. Enter a name in the *Access Token Name* field, which should help reference back to the token's assignee.
    c. In the *User* field, enter the application username for the user who will be assigned this token. This must be a valid application username, and the user must have application permissions to view the request data they will be querying using the API.
    d. Enter an Expiration time for the token. The token will expire after the listed date, and a new token must be provided for that user to access the tool.
2. Click the **Create** button to create the token with the provided details.

3. After clicking **Create**, the following screen appears, displaying the newly created token. You can manually copy the token text, or click **Copy** to copy the token to your clipboard.



4. Save the copied token to a secure location, where you can retrieve it to authenticate this user.

(!!) Note: The token is only displayed once on this screen and is not retained anywhere after this window is closed. You must save the token now.

3.2 How to Access the ATIPXpress API

If you have been provided with a valid access token, you can access the ATIPXpress API from your desktop search bar. Search for **ATIPXpress API** in your program files and select this program to open the ATIPXpress API application.

The methods available in the API are described in the *Request Methods* and *Audit Methods* sections of this document.

(!!) Note: You must have Request Type or Audit permissions within the application in order to use the Request and Audit methods, respectively.

3.3 API Logs

You can view logs of actions taken in the API. These are available in the following location: *<ATIPXpress_Install_Location>\Logs\ATIPXpress\API*

If the default location is chosen, the logs should be located at:

C:\Program Files\ATIPXpress\Logs\ATIPXpress\API

4 Request Methods

The following sections provide details and steps to use the Request methods to query request data.

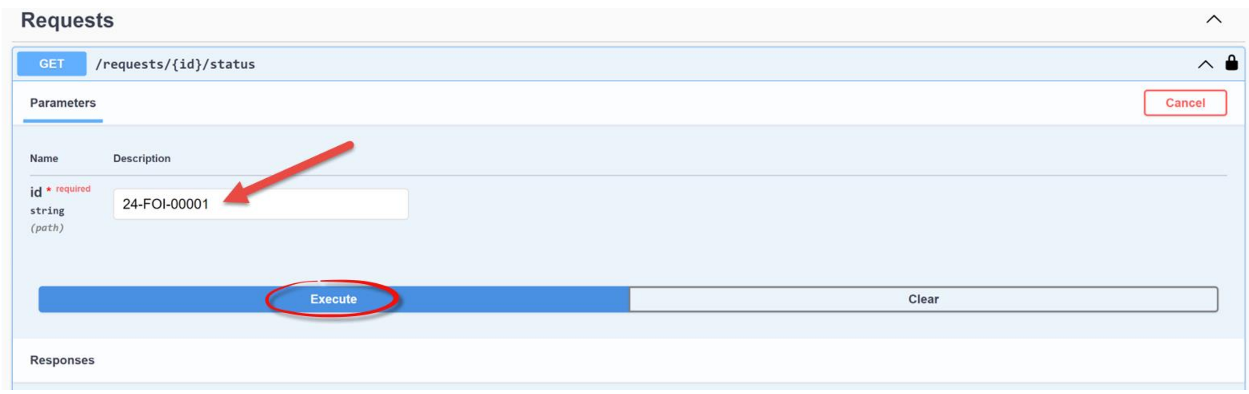
4.1 Get Request Status

You can use the Get Request Status method to retrieve the status of a specific request. This requires the exact ID of the request you are querying.

(!!) Note: You must have Request Type permissions within the application to access this method.

To use the Get Request Status function:

1. Under *Requests*, locate the *id string* field and enter the Request ID exactly as it appears in the application, then click **Execute**:



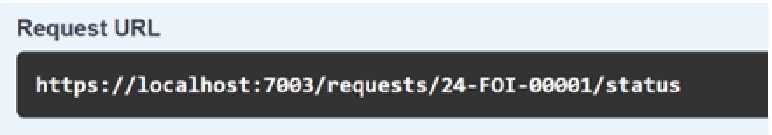
2. After clicking **Execute**, the request data appears in the *Response Body* as shown below.

3. There are options to **(A) Copy** this to your clipboard, or **(B) Download** the response body.

To use this function via URL, use the following string with the application URL in place of <AX-api-url>, and replacing the {id} with the request ID you are querying:

<AX-api-url>/requests/**{id}**/status

An example URL is shown below:



4.2 Get Request

You can use the Get Request API to query for data on a specific request. This requires the exact Request ID.

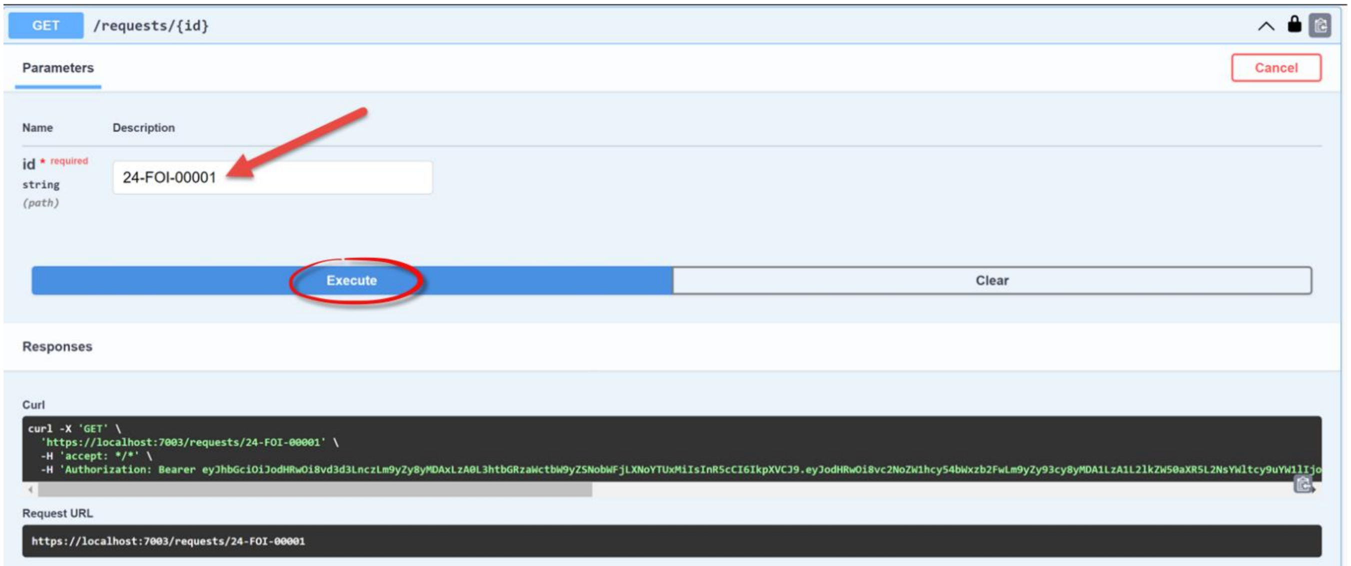
(!!) Note: You must have Request Type permissions within the application to access this method.

This method returns the following Request details:

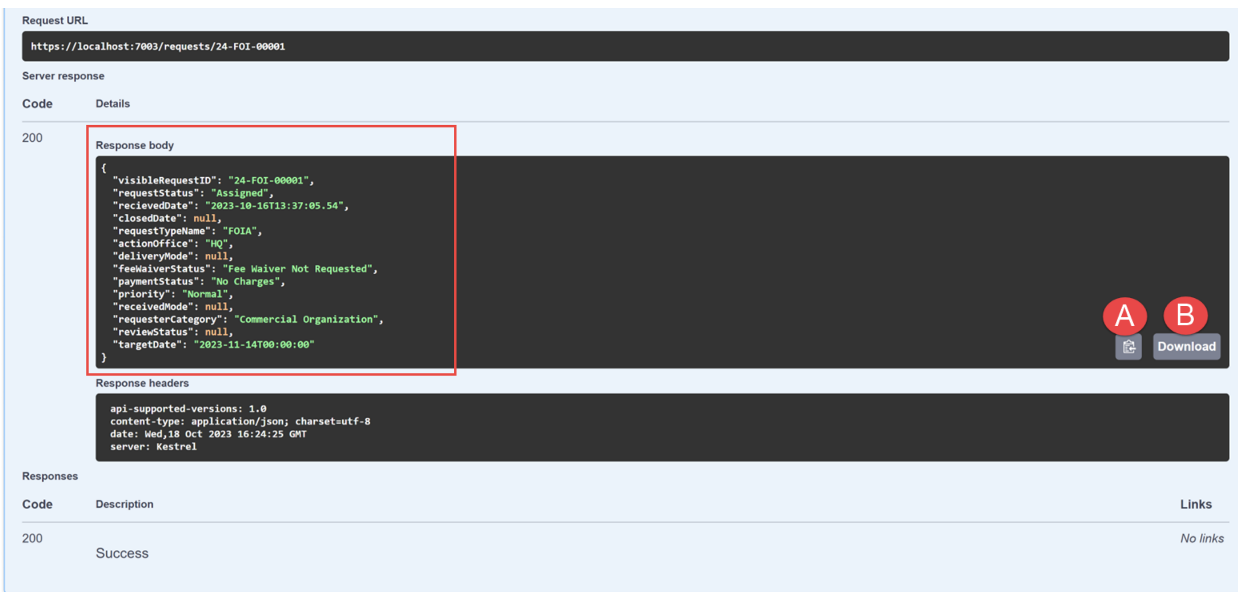
- *visibleRequestID*: ID of the request
- *requestStatus*: Status of the request
- *receivedDate*: Date when the request was received.
- *closedDate*: Date when the request was closed. It is null if request is active
- *requestTypeName*: Request Type
- *actionOffice*: Action Office
- *deliveryMode*: Mode of delivery of response to the requester
- *feeWaiverStatus*: Fee waiver status of the request
- *paymentStatus*: Payment Status of the request
- *priority*: Priority of the request
- *receivedMode*: How the request was received.
- *requesterCategory*: Category to which the requester belongs to
- *reviewStatus*: Review status of the request
- *targetDate*: Target date to close the request

To use the Get Request Status function:

1. Under *Requests*, locate the *id string* field and enter the Request ID exactly as it appears in the application, then click **Execute**:



2. After clicking **Execute**, the request data appears in the *Response Body* as shown below.



3. There are options to **(A) Copy** this to your clipboard, or **(B) Download** the response body.

4.3 Get Requests

You can use the Get Requests API to query for requests with matching attributes.

(!!) Note: You must have Request Type permissions within the application to access this method.

4.3.1 Get Requests Parameters

The parameters available within the Get Requests method are outlined in the following tables.

Request Status parameter options:

Parameter	Attribute
Need	On Hold-Need Info/Clarification
CWithheld	On Hold-Fee Related
DAdded	Documents Added
DD	Documents Delivered
DF	Delivery Failed
DP	Delivery Pending
Amended	Amended
Assigned	Assigned
Canceled	Canceled
Closed	Closed
Completed	Disposition Accepted
Received	Received
Screened	Perfected
ReqforDocs	Request for Docs Sent
RvwPending	Review Pending

Parameter	Attribute
RvwAprvd	Review Approved
RvwDisApvd	Review Disapproved
DAddRvwLog	Documents Added to Review Log

Request Type Name Parameters:

Parameter	Example
Request Type name, as found in Administration > Request Management	“ATIP”, “ATIP,APP”

Action Office parameter options:

Parameter	Example
Office Codes, as found in Administration > Organization Setup	“HQ”, “HQ,DC”

4.3.2 Using the Get Requests Method

Follow the steps below to use the Get Requests method:

1. Under *Post/requests*, use the *Request body* to customize the query. Beside each parameter you are using, enter the Request data that you would like returned in the results:

POST `/requests`

Parameters

No parameters

Request body

```
{
  "pageSize": 100,
  "pageNumber": 1,
  "visibleRequestIDs": [
    "24-FOI-00001", "24-FOI-00002"
  ],
  "recievedDateFrom": "2023-01-01",
  "recievedDateTo": "2024-01-01",
  "closedDateFrom": "2023-01-01",
  "closedDateTo": "2023-01-01",
  "requestStatus": [
    "Assigned", "Closed"
  ],
  "requestTypeName": [
    "FOIA"
  ],
  "actionOffice": [
    "HQ"
  ]
}
```

2. The pageSize and pageNumber fields are required and configure the formatting for the returned results. *pageSize* indicates the number of items returned per page, and *pageNumber* dictates the current page number.
3. If you are using the *receivedDateFrom* or *closedDateFrom* fields, you must also use the *receivedDateTo* and *closedDateTo* fields, respectively.

(!!) Note: Dates follow UTC format and will take the hours and minutes as 00:00 unless specified. To fetch values for a specific date, either specify the time or use the following date. For example, to fetch values for 11/1/2023, use either 2023-11-01T23:59Z or 2023-11-02.

4. For details on each parameter, see the *Get Request Parameters* section. Remove any parameters you are not using in the query.
5. After executing the method, the output returns all Requests matching the parameters used in the Request Body.

5 Audit Methods

You can use the Audit Methods to query for User Actions and User Logins in the application.

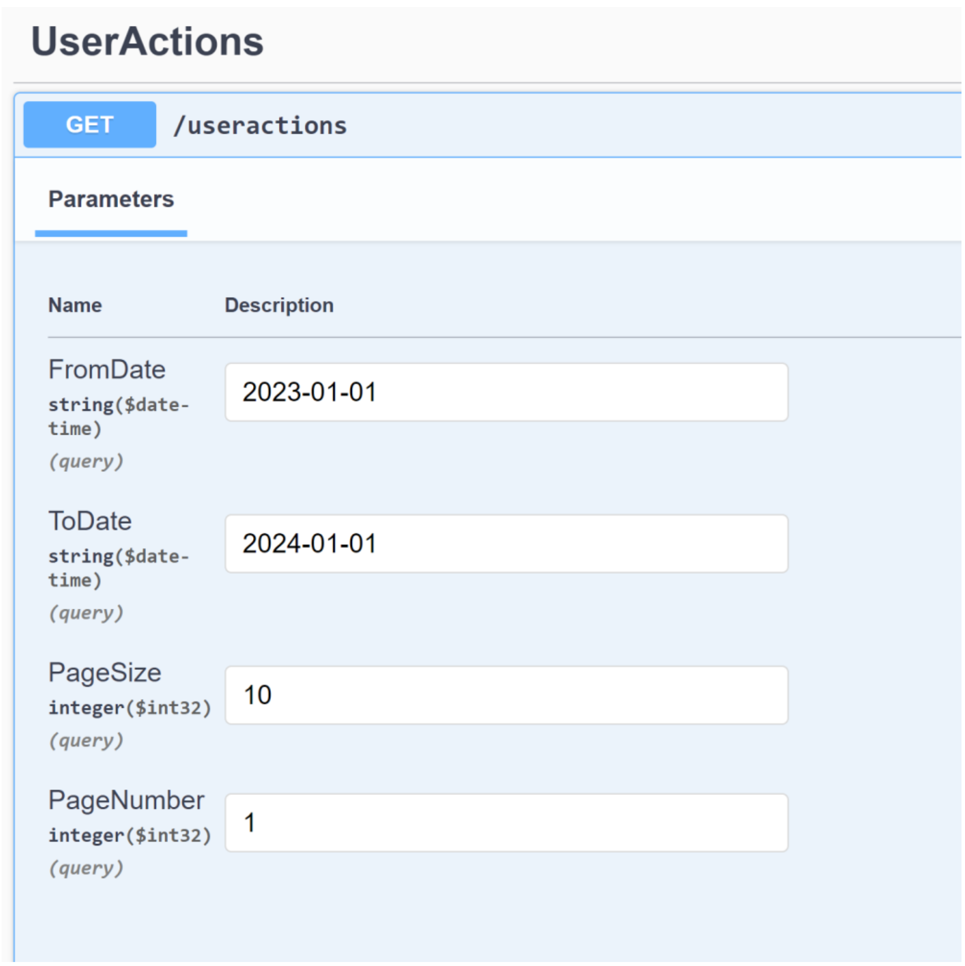
5.1 User Actions

The UserActions method gets all user actions between two dates.

(!!) Note: You must have Audit permissions within the application to access this method.

To audit user actions using the API:

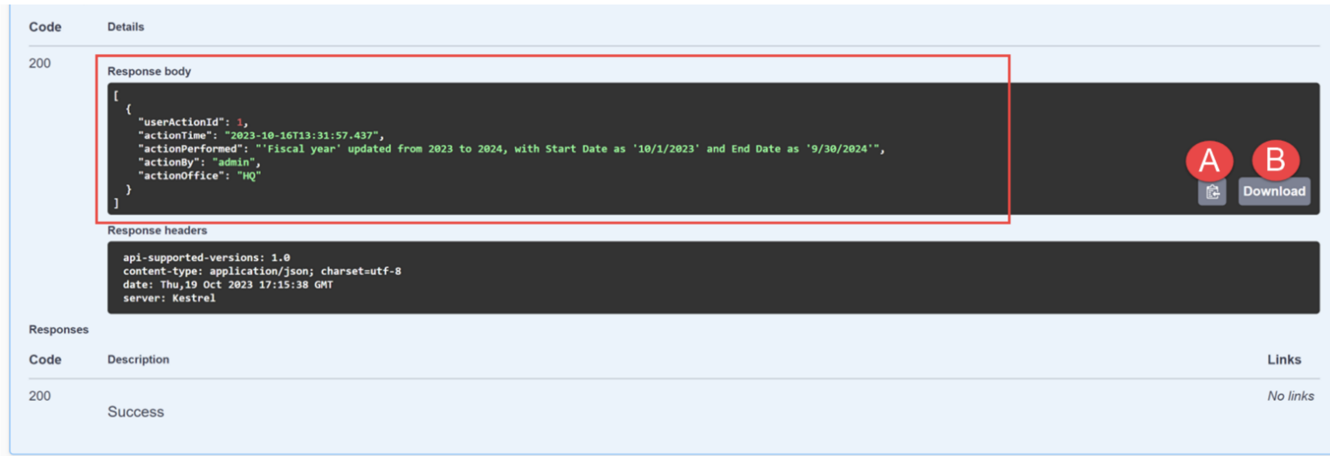
1. Locate the UserActions section of the API, as shown below:



2. Enter the dates you'd like to audit User Actions between using the *FromDate* and *ToDate* fields.

(!!) Note: Dates follow UTC format and will take the hours and minutes as 00:00 unless specified. To fetch values for a specific date, either specify the time or use the following date. For example, to fetch values for 11/1/2023, use either 2023-11-01T23:59Z or 2023-11-02.

3. The pageSize and pageNumber fields are required and configure the formatting for the returned results. *pageSize* indicates the number of items returned per page, and *pageNumber* dictates the current page number.
4. Execute the method to view matching results, which display in the *Response body* field. There are options to **(A) Copy** this to your clipboard, or **(B) Download** the response body



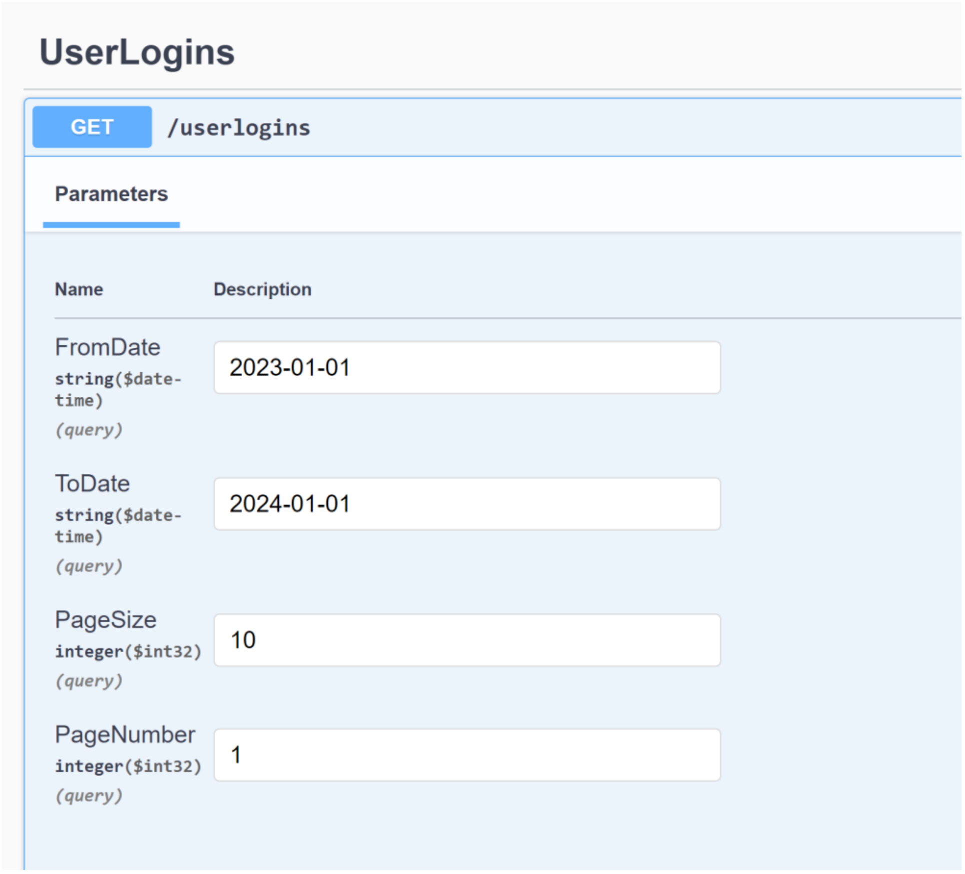
5.2 User Logins

You can use the UserLogins method to gets all user logins between two dates.

(!!) Note: Users utilizing Audit methods must have Audit permissions within the application.

To audit user logins using the API:

1. Locate the UserActions section of the API, as shown below:



2. Enter the dates you'd like to audit User Logins between using the *FromDate* and *ToDate* fields.

(!!) Note: Dates follow UTC format and will take the hours and minutes as 00:00 unless specified. To fetch values for a specific date, either specify the time or use the following date. For example, to fetch values for 11/1/2023, use either 2023-11-01T23:59Z or 2023-11-02.

3. The pageSize and pageNumber fields are required and configure the formatting for the returned results. *pageSize* indicates the number of items returned per page, and *pageNumber* dictates the current page number.
4. Execute the method to view matching results, which display in the *Response body* field. There are options to **(A) Copy** this to your clipboard, or **(B) Download** the response body


Server response
Code
Details
200
Response body
[
{
"sessionID": 1,
"userName": "Admin, Admin",
"actionOffice": "HQ",
"workStation": "::1",
"loginTime": "2023-10-16T13:31:54.64",
"logoutTime": "2023-10-16T14:16:22.55",
"duration": 2668,
"groupID": 1,
"groupName": "Admin",
"loginSuccess": "Success"
},
{
"sessionID": 2,
"userName": "Admin, Admin",
"actionOffice": "HQ",
"workStation": "::1",
"loginTime": "2023-10-17T11:51:28.41",
"logoutTime": "2023-10-17T11:56:40.91",
"duration": 312,
"groupID": 1,
"groupName": "Admin",
"loginSuccess": "Success"
}
]
A
B
Download