

## Table of Contents

1. Introduction
  - a. Root Domain
  - b. List of Endpoints
  - c. Security
  - d. Technical Support
2. Archive Instances
3. Default Behaviors
4. Authentication Endpoint
  - a. Introduction
  - b. Arguments
  - c. Response
5. Query Endpoint
  - a. Introduction
  - b. Arguments
  - c. Response
6. File Endpoint
  - a. Introduction
  - b. Arguments
  - c. Response
7. Upload Endpoint
  - a. Introduction
  - b. Arguments
  - c. Response
8. Future Versions

# 1. Introduction

The Archive API was developed by Premier One to allow 3<sup>rd</sup> party vendors access to the basic features of the Archive software. In it's first iteration it is very basic and has minimal features. In the future the API is expected to grow in ability and usability. The API consists of 4 endpoints listed below where each endpoints arguments and expected responses are discussed in detail. Each vendor will be given a unique username and password which will be required in order to receive an authentication token to use the API and it's functionalities. The root domain and all endpoints require the use of secure hypertext transfer protocol (https).

## a. Root Domain

The root domain for accessing the API is:

<https://api.tmiarchive.com/>

The root domain has no functionality other than to display the current version of the API and instruct an end user to use the correct endpoint for whatever it is they wish to accomplish. This root domain is the same for every location across the TMI network.

## b. List of Endpoints

The Archive API consists of 4 endpoints:

1. Authentication: <https://api.tmiarchive.com/authentication>

The authentication endpoint is used to generate authentication tokens which must then be used to access the other endpoints of the API.

2. Query Endpoint: <https://api.tmiarchive.com/query>

The query endpoint is used for querying the Archive application for information about and the content of a specific folder in Archive. In this first iteration this endpoint is tied with the Ramquest databases at each TMI location as such the only folders this endpoint currently has access to are those with a Ramquest folder ID which can be found in the Ramquest application.

3. File Endpoint: <https://api.tmiarchive.com/file>

The file endpoint is used in conjunction with the query endpoint to download a file. Currently this file must be specified by filename or file ID. In the future this endpoint will become more restrictive in what files are allowed to be downloaded. As of now there is no way to restrict which files can and cannot be downloaded.

4. Upload Endpoint: <https://api.tmiarchive.com/upload>

The upload endpoint is used to upload files into the API with all the associated properties of that file including Archive keywords and values as well as file type and category. The uploading of files similar to the query endpoint is restricted to folders with a Ramquest folder ID at this time.

## c. Security

The Archive API is built to be as secure as possible. Just by nature of what an API is the root domain and endpoints are exposed to the outside world known as the internet which necessitates being as cautious as possible to avoid a data breach.

As previously mentioned, the API uses the encrypted https protocol as well as a unique username and password for each user. In addition to these basic security measures, we have set the API to only allow whitelisted IP addresses. Vendors must submit a list of IP addresses they wish to use to access the API. Internal IP addresses on the TMI network will already have access to the API. Every 6 months an IP address audit will be done automatically. Any whitelisted IP addresses that have not accessed the API during that time will be removed from the system.

API tokens generated from the authentication endpoint will only be valid for 15 minutes and requests are limited to 10 per minute. To adjust these limits, contact the development team at Premier One.

Vendors working with TMI and the Archive application will initially only be able to target the Archive Demo instance. In production this restriction will be lifted, and full access will be implemented across all TMI Archive instances. During the production the “demo” location argument will be used for all API requests.

## d. Technical Support

For technical support, additional information, or to whitelist additional domains please contact the Premier One development team.

### **Premier One Development Team**

Email: [devteam@premier-one.com](mailto:devteam@premier-one.com)

Phone: (785) 354-8233

Someone from the development team will get back to you ASAP to help resolve any issues with the API you may have. For feature requests or other non-technical feedback or communication please contact the Development Manager.

### **Premier One Development Manager**

Joseph Mioni

Email: [josephm@premier-one.com](mailto:josephm@premier-one.com)

## 2. Archive Instances

TMI has multiple locations and as such has multiple instances of the Archive application. All endpoints except for the authentication require a location argument to work properly. Below is a table of each location and the corresponding argument needed for that location.

Agency/Location	Argument Value
Demo Archive	demo
Dominion Title	dom
Kansas Secured Title	kst
Missouri Secured Title	mst
National Secured Title	nst
Nebraska Title Company	ntc
Oklahoma Secured Title	ost
Rochester Title & Escrow	rte
Secured Title Of KC	skc
Secured Title Of TX	stx
Title Midwest Corporate	tmi

## 3. Default Behaviors

There are several default behaviors for different aspects of the API that are listed below some of these defaults can be overridden others cannot by design.

Property	Default Value	Can Be Overridden
Upload File Name	Search / Exam Notes	Yes
Upload File Type	Search / Exam Notes	No
Upload File Category	Title	No
Query Include File List	false	Yes
Query Include Folder Keywords	false	Yes

## 4. Authentication Endpoint

### a. Introduction

To use the authentication endpoint, you will need to craft a custom http request in whatever development language you are using. For this documentation guide PostMan is being used to test generate and receive the requests and responses. PostMan allows you to easily generate the code needed for a variety of development languages. The API will work with any tool that can generate and receive http requests and responses. The authentication endpoint uses POST. For the authentication endpoint all arguments must be passed in the http request body. The content type is x-www-form-urlencoded. The endpoint will issue an authentication token that will be needed to access the other endpoints successfully. The token is valid for 15 minutes.

#### Quick Reference

URL: <https://api.tmiarchive.com/authentication>

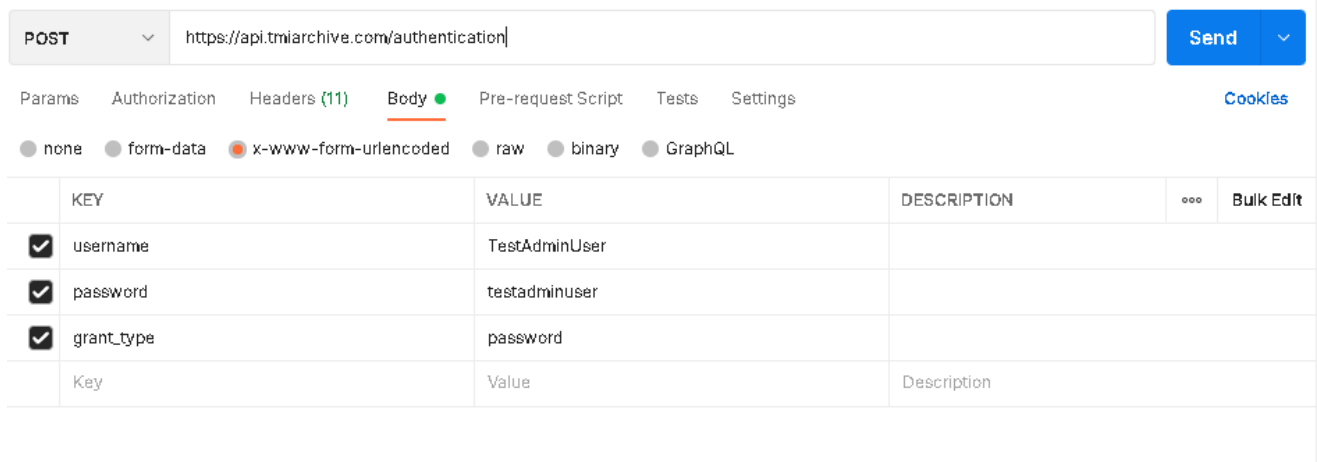
Request Type: POST

Body Content Type: application/x-www-form-urlencoded

### b. Arguments

Argument	Value	Required	Location
username	Unique to vendor.	Yes	Body
password	Unique to vendor.	Yes	Body
grant_type	password	Yes	Body

Sample:



The screenshot shows a Postman interface for a POST request to `https://api.tmiarchive.com/authentication`. The request body is configured as `x-www-form-urlencoded` and contains the following parameters:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	TestAdminUser	
<input checked="" type="checkbox"/> password	testadminuser	
<input checked="" type="checkbox"/> grant_type	password	
Key	Value	Description

## c. Response

The expected response is an authentication token. If you have any invalid arguments the endpoint will return an error.

Sample:

The screenshot shows a REST client interface with the following details:

- Body tab selected, showing a JSON response in "Pretty" format.
- Status: 200 OK, Time: 189 ms, Size: 814 B.
- Response content:
 

```

1 {
2   "access_token":
3     "VdZktqX3NMbcTnayovyjv8qeSIPiojdwISG_iozwI5cdv5r6_38ffBmAd2oq6JvdepSSJ0ssA83dlr7_wJbXWPPxjp9s0nQXha9p-rzDYH1M
4     RbCG6NpMmnWiqPoG-AaK7Wt5C5sCa_11P4CJFZxli4i0str2xzKQ2JY0N14Nf0df0b8ovtzeTxoP7nkh4oxG1IunGZPCF09i6Kokgb18dg8Mc
5     lmdcxewzw7XnkD0K0g9YZ3zR5oqoXxZd7Qn09jB_zq82uoG6rYALH29gMyd0T_Cf5dfK1q7MsZ-v05547DFJ6kYg_9tp3xtvGh5PURhb-Sbfr
6     MZY65N9CgXK1jb2w",
7   "token_type": "bearer",
8   "expires_in": 899
9 }
      
```

## 5. Query Endpoint

### a. Introduction

The query endpoint is used to get information about a folder that exists in Archive. The request must be authenticated by including the authentication token in the request header. Content type is application/json. The query endpoint uses GET and at a minimum requires a folder ID as an argument.

Sample of Request Header:

The screenshot shows a REST client interface with the following details:

- Headers tab selected, showing 6 hidden headers.
- Table of headers:

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Content-Type	application/json				
<input checked="" type="checkbox"/> Authorization	Bearer gFLEinhGvST5CEPFrmDit0uVWZGUo8p...				
Key	Value	Description			

#### Quick Reference:

URL: <https://api.tmiarchive.com/query>

Request Type: GET

Header Content-Type: application/json

## b. Arguments

As this endpoint uses GET the arguments are submitted via URL parameters. The keywords arguments determines if you would like to include the folder keywords and values associated with the Archive folder. The documents keyword determines if you would like to include a list of documents currently in the associated Archive folder. For both of these arguments the default is false so the arguments only need to be included if being set to true.

Argument	Value	Required	Location
<b>folderid</b>	Unique Ramquest folder ID	Yes	URL Param
<b>keywords</b>	true or do not include	No	URL Param
<b>documents</b>	true or do not include	No	URL Param

Sample:

GET <https://api.tmiarchive.com/query?folderid=PWT0382&keywords=true&documents=true> Send

Params **●** Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> folderid	PWT0382			
<input checked="" type="checkbox"/> keywords	true			
<input checked="" type="checkbox"/> documents	true			
Key	Value	Description		

## c. Response

The response will be in JSON format with 3 possible sections: Folder Information, Folder Contents, and Folder Keywords. If the folder ID is not found in Archive it will query Ramquest and import the folder first. If the folder ID is not found in Ramquest the endpoint will return an error.

### Folder Information:

ArchiveFolderID: The internal ID used by Archive.

FolderName: The name of the folder in Archive.

RQFolderID: The folder ID used by Ramquest.

DrawerID: The ID of the drawer that contains the folder.

Drawer: The name of the Drawer that contains the folder.

### Folder Contents:

ArchiveFileID: The file ID used in Archive.

Description: The name of the file in Archive.

Path: The actual path to the physical file.

DrawerID: The ID of the drawer containing the file.  
 Drawer: The name of the drawer containing the file.

#### Folder Keywords:

keywordID: The ID of the keyword used by Archive.  
 Keyword: The name of the keyword.  
 KeywordValue: The assigned value of the keyword.

#### Sample:

```

Body Cookies Headers (11) Test Results Status: 200 OK Time: 440 ms Size: 21.44 KB Save Response
Pretty Raw Preview Visualize JSON
1
2   "FolderInformation": [
3     {
4       "ArchiveFolderID": 40996,
5       "FolderName": "Alex Johnson, Sr. & RaeAnne Johnson",
6       "RQFolderID": "PwT0382",
7       "DrawerID": 79,
8       "Drawer": "Purchase"
9     }
10  ],
11  "FolderContents": [
12    {
13      "ArchiveFileID": 132544,
14      "Description": "40-Year Search 08301120",
15      "Path": "\\P1DEV-RQ\Webtemp\ccDev\PwT0382\4443066_ASK_Service_8532867_1567164012.pdf",
16      "DrawerID": 79,
17      "Drawer": "Purchase"
18    }
19  ]
  
```

## 6. File Endpoint

### a. Introduction

The file endpoint is used to download a file from Archive. The authentication must still be included in the header for this request to work. The file ID will be found using the query endpoint under the Folder Contents section.

#### Quick Reference:

URL: <https://api.tmiarchive.com/file>  
 Request Type: GET  
 Header Content-Type: application/json

### b. Arguments

Argument	Value	Required	Location
fileid	Unique file ID	Yes	URL Param

## Sample:

GET <https://api.tmiarchive.com/api/file?fileid=1325445> Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	fileid	1325445			
	Key	Value	Description		

## c. Response

The response is simply the raw file data. It is up to the vendor to save as a file with an appropriate file name. File type can be determined from the file header or by using the file name received from the query endpoint.

## 7. Upload Endpoint

## a. Introduction

The upload endpoint allows vendors to upload a file into a specified folder. The folder ID is the Ramquest folder ID. The drawer ID will also be needed, the drawer ID can be found in the query endpoint when getting the folder information. Currently the file category and file type are automatically assigned as indicated in Section 3. The file name can be overridden by including the filename argument.

**Quick Reference:**

URL: <https://api.tmiarchive.com/upload>

Request Type: POST

Header Content-Type: application/json

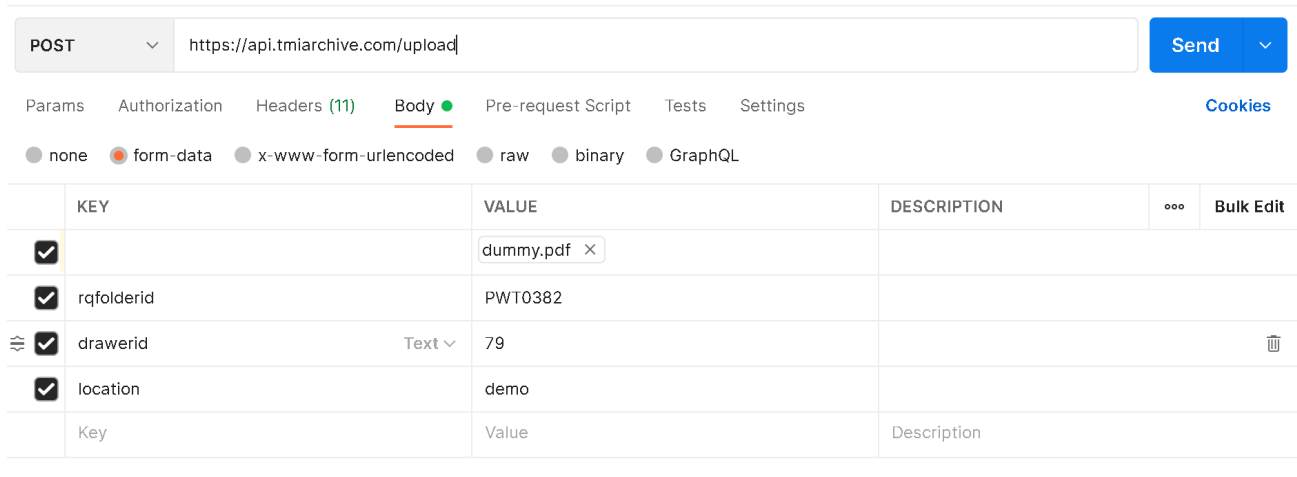
Body Content-Type: multipart/form-data

## b. Arguments

Argument	Value	Required	Location
<b>file</b>	A file	Yes	Body
<b>rqfolderid</b>	Ramquest folder ID where the file should be uploaded.	Yes	Body
<b>drawerid</b>	The drawer the Ramquest folder is located in.	Yes	Body

<b>location</b>	The TMI Archive instance being targeted.	Yes	Body
<b>filename</b>	Optional file name to override default file name.	No	Body

Sample:



POST `https://api.tmiarchive.com/upload` Send

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings Cookies

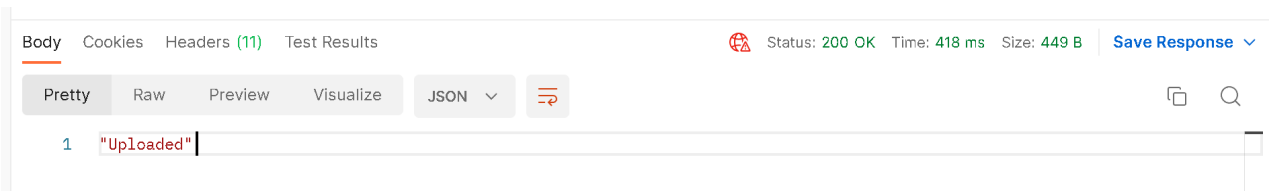
none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>		<code>dummy.pdf</code> <span>×</span>			
<input checked="" type="checkbox"/>	rqfolderid	PWT0382			
<input checked="" type="checkbox"/>	drawerid	Text <code>79</code>			<span>🗑</span>
<input checked="" type="checkbox"/>	location	demo			
	Key	Value	Description		

## c. Response

The endpoint will either return uploaded or an error message.

Sample:



Body Cookies Headers (11) Test Results Status: 200 OK Time: 418 ms Size: 449 B Save Response

Pretty Raw Preview Visualize JSON 🔍

```
1 "Uploaded"
```

## 8. Future Versions

File in this section.