SedonaAPI - Using Postman with GraphQL

12/28/2023 3:33 pm EST

The SedonaAPI is used for our new Angular client but also can be used for third-party access. We do provide some REST API calls similar to AlarmBiller but also have added a new technology called GraphQL. This provides a way to expose queries that the front end can use to dynamically pull only what it needs in each call. Because underneath it uses HTTPS calls to an IIS server, it allows very quick testing through several tools. This document covers one called POSTMAN which is a free tool for testing APIs.

Importing Demo Collection

Click on the Import button on the top left part of the POSTMAN interface. Drag and drop the collections from the .postman folder in the SedonaAPI repo for the most current collection.

Once this collection is imported, click on the Sedona-X GraphQL API Tests collection.

Change the environment to Sedona-X.

Click on the Environments and select the Sedona-X one.

Update the Variables settings for url, username, and password with your correct values for your Sedona-X. You may need to hit the Reset All button if there are current values already in place.

You are now set up and ready to begin working with API calls. You can rename the collection to better describe the system you are hitting.

Getting Authorized with OAUTH

The first thing that must happen before any API calls can be made is to get an OAUTH token for your user. We do this from the collection Authorization tab so that all subsequent call can inherit this token. Sedona-X uses OAuth 2.0 as the authorization token. This is added to the HTTP request as a Bearer token in the request header.

Now you can configure and get a new token. Make sure it is set for Password Credentials as the Grant Type.

Hit the Get New Access Token button and you should get an OAUTH token back.

Click the Proceed button and then click the Use Token button.

Not you have an OAuth Token.

Make sure to hit the Save button to store that token for the rest of your API calls.

Once you have this working you can move on to the next steps.

Making GraphQL Calls

Now that you have a valid OAUTH token, we can hit the /graphql endpoint and make requests. GraphQL is extremely nice in that you don't have to manage a huge list of API endpoints. There is just one URL. There is also a difference in the terms that are used from REST APIs. We don't use GET/POST/PUT/etc to do work. Everything in GraphQL is a POST. The body determines the functionality and is either a QUERY (get data) or a MUTATION (put data). An easy one to start with is requesting the list of Items in the Sedona-X database. You probably will not need to request this in practice, but I use it just to make sure the basic connection is working.

Open the Items Request. This already has the parent Auth setup to be used under the Authorization tab.

Because this is now through GraphQL, the request does require a body. It should return a JSON array object showing that the connection is working to the API including sending in the OAUTH token.

You can now work through the other Queries and Mutation calls.

Because GraphQL allows you to do basic SQL-like things from the front end, you can change the query body to return just what you need. Try getting just the ItemCode and Name. Now try getting the costAmount and laborUnits.

API Documentation

Our REST APIs use Swagger to self-document the calls. GraphQL does this slightly differently. It is called a Schema and can be accessed from the following URL.

https://{YOURSERVER}/graphql/schema

This will let you download a schema file that can be imported into Postman.

Click on the Import button and then drag/drop the schema file. You probably will want to uncheck the Generate collection button.

You might want to rename your Schema in the APIs section as well if you have a lot.

To get the schema to work properly, I had to delete the "repeatable" keyword on the @authorize directive

Now, you can use this in your Query/Mutation to have auto-complete type of functionality. In your request body, you should have a schema drop-down. You may have to hit the refresh button.

But now you should have auto-complete.

You can hit Ctrl+Space to start the autocomplete without typing.