REST API

Documentation (Revised

March 2024)

Overview

The MemberSuite REST API allows the client to take their user specific credentials and make calls to their environment to retrieve, update, add, and remove data. The MemberSuite REST API was built on the open source project, Swagger UI, to give the client the smoothest experience while making calls to their environment. REST architecture allows the Swagger UI to provide multiple data formats rather than the sole option of xml, provided by SOAP. JSON has been set as the default format.

Swagger REST Documentation References

Platform:

https://rest.membersuite.com/platform/swagger/ui/index#/

Security:

https://rest.membersuite.com/security/swagger/ui/index#/

CRM:

https://rest.membersuite.com/crm/swagger/ui/index#/

Membership:

https://rest.membersuite.com/membership/swagger/ui/index#/

Subscriptions:

https://rest.membersuite.com/subscriptions/swagger/ui/index

Certifications:

https://rest.membersuite.com/certifications/swagger/ui/index#

Committees:

https://rest.membersuite.com/committees/swagger/ui/index#/

Events:

https://rest.membersuite.com/events/swagger/ui/index#/

Orders:

https://rest.membersuite.com/orders/swagger/ui/index#/

Financial:

https://rest.membersuite.com/financial/swagger/ui/index#

Fundraising:

https://rest.membersuite.com/fundraising/swagger/ui/index#/

Awards:

https://rest.membersuite.com/awards/swagger/ui/index#/

Credentials

Authentication

Credentials are necessary to create an AuthToken which will be used to access the customers environment. The tenantid is also needed to obtain the user specific token for the specific MemberSuite instance.

- Once the Credentials have been entered, the user can click "Try it out!" in Swagger also to receive the Base64 encoded auth token needed to access the user specific environment through the REST endpoint.

How To: Locate Your Association ID (SSO) and Tenant ID (SSO and Non-SSO):

You can determine your Association ID and Tenant ID (Association Key) by logging into the console and navigating to the Association Settings screen.

1. Click on Setup in the Console toolbar

2. Click on Association Settings

3. Make a note of your specific Association's ID and Tenant ID (Association Key)

How To: Obtaining a Bearer Token:

https://rest.membersuite.com/platform/swagger/ui/index#!/AuthToken/AuthToken_LogInUser

**To access our REST API for non SSO API calls a specific type of an account is needed. Please contact our Support Team, or your CSM directly to obtain this for each external developer that needs this for your site.

POST	/platform/v2/loginUser/{tenantId}

Response Class (Status 200)

OK

Model | Model Schema

```
{
  "success": true,
  "data": {},
  "firstErrorMessage": "string",
  "errors": [
    "string"
  ],
  "statusCode": 100
}
```

Response Content Type `application/json`

Parameters

Parameter	Value	Description	Parameter Type	Data Type
tenantId	(required)		path	long
request	(required)		body	Model | Model Schema

```
{
  "email": "string",
  "password": "string"
}
```

Click to set as parameter value

Parameter content type:
`application/json`

| Asynchronous | | Asynchronous operation | header | boolean |

Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	BadRequest		
500	InternalServerError		

Try it out!

Example cURL:

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

   "email": "string",

   "password": "string"

}' 'https://rest.membersuite.com/platform/v2/loginUser/XXXXX'

Response Body: 200 Status Code

The response JSON will have a list of errors (if any). Users should check the **success** field, if that is true, they should expect some data in the **data** property. Else, at least one error is found. The first error gets assigned to **firstErrorMessage** for easy access.

Important: A successful response will give you an idToken string valid for 5 hours that you will use as the authorization to make the other non SSO API calls. For subsequent GET, PUT, POST, PATCH calls for example, be sure to add "Bearer" and a space before the idToken string within the Authorization header of that call you are making.

Common REST Call Examples

CRM

CRM stands for Customer Relationship Management. At its simplest definition, a CRM system allows businesses to manage business relationships and the data and information associated with them.

With CRM, you can store customer and prospect contact information, accounts, leads, and sales opportunities in one central location, so the information is accessible by many, in real time. A CRM can expand to include sophisticated features to help teams collaborate with colleagues and customers, send customized emails, gather insights etc.

GET Individual

https://rest.membersuite.com/crm/swagger/ui/index#!/Individuals/Individuals_Get

Fetches the data related to the provided Individual MSQL.

MSQL can be generated via the MemberSuite console by selecting MSQL from the output format dropdown of an individual advanced search.

Input Parameters:

ID: GUID representing the specific individual Id.

Authorization: Bearer Token

MSQL Example:

select LocalID, LastName, FirstName, EmailAddress, _Preferred_Address_City, _Preferred_Address_State, Membership.ReceivesMemberBenefits, Membership.Type.Name from Individual where (ID='d0b367b1-0006-cf46-5bb3-0b4615706d20') order by SortName, Name

Example Parameter:

Membership.ReceivesMemberBenefits: This field determines if they are a member or not.

true:

If a user has an active Membership - Membership.ReceivesMemberBenefits would be true (Member)

null/false:

If a user has never had a membership - Membership.ReceivesMemberBenefits would be null (Non-Member)

If a user had a membership that expired and are dropped - Membership.ReceivesMemberBenefits would be false (Non-Member)

Example cURL:

curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer [idToken]' 'https://rest.membersuite.com/crm/v1/individuals/[tenantID]?msql=select LocalID, LastName, FirstName, EmailAddress, _Preferred_Address_City, _Preferred_Address_State, Membership.ReceivesMemberBenefits, Membership.Type.Name from Individual where (ID='d0b367b1-0006-cf46-5bb3-0b4615706d20') order by SortName, Name&page=1&pageSize=20'

Be sure to add "Bearer" and a space before the idToken string within the Authorization header, as shown in the example cURL above.

Response Body: 200 Status Code

PATCH Individual

https://rest.membersuite.com/crm/swagger/ui/index#!/Individuals/Individuals_Patch

Updates the specified field(s) for the specified individual.

Please Note: It is strongly recommended to use the Sandbox environment for the specific MemberSuite instance to test out data writeback calls such as this one. Contact our Support Team or your CSM for more details.

Input Parameters:

 id: GUID representing the specific individual Id.

Authorization: Bearer Token

Payload Example:

{
"id": "63c37506-0006-c199-d948-2326badec23b",
"Prefix": "Mr.",
"FirstName": "Joey",
"LastName": "Tester1",
"Suffix": "IIII",

"Title": "Vice President/Staff",

"EmailAddress": "test@test.com",

"Work_PhoneNumber": "18887774444",

"Mobile_PhoneNumber": "19997775555",

"Home_PhoneNumber": "19997775566",

"Company": "ABC Corp",

"options": [{ "Name": "Employer__c", "Value": "Employer Greg 788889" }, { "Name": "MyBio__c", "Value": "Test Bio 766229" }, { "Name": "Gender__c", "Value": "Female" }, { "Name": "primaryjobsetting__c", "Value": "Office Tester" }, { "Name": "AreaofExpertise__c", "Value": "Doctor" }, { "Name": "role__c", "Value": "Physician" }],

"preferredAddressType": "79e9ba70-000f-ca85-3b9d-0b371d234b26",

"Home_address": {

    "city": "Tampa",

    "country": "US",

    "line1": "123 Papers Street",

    "line2": "112s",

    "postalCode": "33618",

    "state": "FL",

    "stateName": "Florida",

    "__typename": "Address",

},

"Work_address": {

    "city": "New York City",

    "country": "US",

    "line1": "dk arts",

    "line2": "line322",

    "postalCode": "30329",

```
        "state": "NY",

        "stateName": "New York",

        "__typename": "Address",

    }

}
```

Example cURL:

```
curl -X PATCH --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImRHVGV4Q1BHSTk5bWQ1NHkxWFlFLSJ9.eyJ1c
2VyTmFtZSI6ImFwaXVzZXIzNDUiLCJwYXJ0aXRpb25LZXIzIjpbMzU1MjNdLCJjb250ZXh0UGFyYW1l
dGVycyI6eyJpc0FwaVVzZXIiOnRydWUsIm1lbWJlclN1aXRlVXNlcklkIjoiZDBiMzY3YjEtMDAzMi1jM
WFmLTExMzMtMGI0NTBlOGI2ZGE0IiwicGFydGl0aW9uS2V5IjozNTUyMywidXNlclR5cGUiOiJQb3
J0YWwifSwibmlja25hbWUiOiJhcGl1c2VyMzQ1IiwibmFtZSI6IkFQSSBVc2VyIiwicGljdHVyZSI6Imh0
dHBzOi8vcy5ncmF2YXRhci5jb20vYXZhdGFyL2RlMGQyNjYzZTM4NTEyMGNkOWQwYmRkN2Fm
M2IyMmEzP3M9NDgwJnI9cGcmZD1odHRwcyUzQSUyRiUyRmNkbi5hdXRoMC5jb20lMkZhdmF0
YXJzJTJGYXUucG5nIiwidXBkYXRlZF9hdCI6IjIwMjMtMDQtMjdUMTU6MTM6NDIuMDgyWiIsImVt
YWlsIjoiYXBpdXNlcjM0NUB0ZXN0LmNvbSIsImVtYWlsX3ZlcmlmaWVkIjpmYWxzZSwiaXNzIjoiaHR
0cHM6Ly9tZW1iZXItc3VpdGUtcHJ
```

```
vZC51cy5hdXRoMC5jb20vIiwiYXVkIjoiQVRTOW51cnc4ZTBsUFhwZU1yaUZkOTFrQ0JUV2dweFgiL
CJpYXQiOjE2ODI2MDg0MjIsImV4cCI6MTY4MjY0NDQyMiwic3ViIjoiYXV0aDB8NjNmOGU4MGYw
MjQ4NGFiN2JiNjY2OGZjIn0.chNgEUbMrAH6CgGajPXcEBw3HsAuRG9ShJGu1CTwm2xMPHThr_v
QlsAX6Zum_vTJ-D6hZNNW4BVRUSKf6L4T8YOBaeQNZ9xyRYhl-e2D3hY8AiJlk-
g87pBNnRaTCsRgj3nuPB44A9e6xP-DqOUKIqMbv7PtvT-TaGUFdlyMyEQ47jE-
KsEHaJfcNouc_uRQ1gw2Pg7VUe5GqgtmtZofyW_cnv_UgIsP1C1zP-
jlY6MVah2HfiARwNPidCb0CH8UG8tCqTbyuIq_y-8j-
3BaM7n1sSwnYVzbuWgFH6BEm2iy67PqCr2jaJzSmdC9_7ErCMp_5JVi-4QcpY_SC3iCvg' -d '{

"id": "63c37506-0006-c199-d948-2326badec23b",

"Prefix": "Mr.",

"FirstName": "Joey",

"LastName": "Tester1",

"Suffix": "IIII",
```

"Title": "Vice President/Staff",

"EmailAddress": "test@test.com",

"Work_PhoneNumber": "18887774444",

"Mobile_PhoneNumber": "19997775555",

"Home_PhoneNumber": "19997775566",

"Company": "ABC Corp",

"options": [{ "Name": "Employer__c", "Value": "Employer Greg 788889" }, { "Name": "MyBio__c", "Value": "Test Bio 766229" }, { "Name": "Gender__c", "Value": "Female" }, { "Name": "primaryjobsetting__c", "Value": "Office Tester" }, { "Name": "AreaofExpertise__c", "Value": "Doctor" }, { "Name": "role__c", "Value": "Physician" }],

"preferredAddressType": "79e9ba70-000f-ca85-3b9d-0b371d234b26",

"Home_address": {

    "city": "Tampa",

    "country": "US",

    "line1": "123 Papers Street",

    "line2": "112s",

    "postalCode": "33618",

    "state": "FL",

    "stateName": "Florida",

    "__typename": "Address",

},

"Work_address": {

    "city": "New York City",

    "country": "US",

    "line1": "dk arts",

    "line2": "line322",

    "postalCode": "30329",

    "state": "NY",

```
    "stateName": "New York",

    "__typename": "Address",

  }

} ' 'https://rest.membersuite.com/crm/v1/individuals/patch/63c37506-0006-c199-d948-
2326badec23b '
```

Be sure to add "Bearer" and a space before the idToken string within the Authorization header, as shown in the example cURL above.

Response Body: 200 Status Code

Memberships

GET Memberships

https://rest.membersuite.com/membership/swagger/ui/index#!/Memberships/Memberships_Get

Fetches the data related to the provided Memberships MSQL.

MSQL can be generated via the MemberSuite console by selecting MSQL from the output format dropdown of a members advanced search.

Authorization: Bearer Token

MSQL Example:
select Owner.Name, Type.Name, PrimaryChapter.Name, JoinDate, ExpirationDate, ReceivesMemberBenefits from Membership order by Owner.Name, ReceivesMemberBenefits DESC, ExpirationDate DESC

Example Parameter:

ReceivesMemberBenefits: This field determines if they are a member or not.

true:

If a user has an active Membership - receivesMemberBenefits would be true (Member)

null/false:

If a user has never had a membership - receivesMemberBenefits would be null (Non-Member)

If a user had a membership that expired and are dropped - receivesMemberBenefits would be false (Non-Member)

Example cURL:

curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer [idToken]''https://rest.membersuite.com/membership/v1/memberships/[tenantID]?msql= select Owner.Name, Type.Name, PrimaryChapter.Name, JoinDate, ExpirationDate, ReceivesMemberBenefits from Membership order by Owner.Name, ReceivesMemberBenefits DESC, ExpirationDate DESC &page=1&pageSize=20'

Be sure to add "Bearer" and a space before the idToken string within the Authorization header, as shown in the example cURL above.

Response Body: 200 Status Code

Events

GET Event Registrations

https://rest.membersuite.com/events/swagger/ui/index#!/EventRegistrations/EventRegistrations_Get

Fetches the data related to the provided Event Registrations MSQL.

MSQL can be generated via the MemberSuite console by selecting MSQL from the output format dropdown of an event registrations advanced search.

EventRegistration GUID below in MSQL is the unique Event GUID (in the URL on the event 360 page in MS)

Authorization: Bearer Token

MSQL Example:

select LocalID, Individual.LocalID, Event.Code, BadgeName, BadgeOrganization, Individual.FirstName, Individual.LastName, Individual.Designation, Individual.Nickname, Individual.EmailAddress, CreatedDate, Individual.PreferredAddressType.Name, Individual._Preferred_Address_Line1, Individual._Preferred_Address_Line2, Individual._Preferred_Address_City, Individual._Preferred_Address_StateName, Individual._Preferred_Address_PostalCode, Individual._Preferred_Address_Country, Individual.PreferredPhoneNumberType.Name, Individual._Preferred_PhoneNumber from EventRegistration('b9e368ec-0078-c5ca-269d-0b3eac79f2e1') where (CancellationDate IS NULL AND OnWaitList=0) order by Name

Response Body: 200 Status Code

POST Event Registrations

https://rest.membersuite.com/events/swagger/ui/index#!/EventRegistrations/EventRegistrations_Post

Adds an event registration record in MS for the specified individual.

Input Parameters:

Owner: GUID representing the specific individual Id.

Authorization: Bearer Token

Payload Example:

```
{
"Owner": "3ec42f0c-0006-c4ed-7834-0b4512794a90",
"Event": "3ec42f0c-0078-cd62-d948-027d6f6be3bf",
"Fee": "3ec42f0c-007e-c6c1-d948-6875df6fe3bf",
"CreatedDate": "2023-01-10T14:17:12-05:00",
"DateApproved": "2023-01-10T14:17:12-05:00",
"Approved": "true",
"Notes": "Total Payment: $3174, Guest Count: 2",
}
```

Response Body: 201 Status Code

Public Saved Searches

Retrieval and Execute Results

Lookup Applicable Public Saved Searches List
https://rest.membersuite.com/platform/swagger/ui/index#!/DataSuite/DataSuite_ExecuteSearch

Fetches the data related to the provided MSQL.

Authorization: Bearer Token

MSQL Example with Name Filter (Public Filter is Required):

select Name, ID, ApplicableType, CreatedDate, CreatedBy.Name from SavedSearch where (IsPublic=1 AND Name LIKE '%Active Members%') order by Name

MSQL Example with CreatedDate Filter (Public Filter is Required):

select Name, ID, ApplicableType, CreatedDate, CreatedBy.Name from SavedSearch where (IsPublic=1 AND CreatedDate>='2/1/2024 12:00:00 AM') order by Name

Important Note: The developer can filter by the Name field as shown in the MSQL above in order to obtain the unique id GUID(s). MemberSuite Clients could use a certain keyword in the name of each Public Saved Search for example to make it easier for a developer to filter on that keyword in the Name filter, and only pull that list of applicable Public Saved Searches and their ids that are needed for the integration.

Example cURL:

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImd0eSI6WyJwYXNzd29yZCJdLCJraWQiOiJNYWx1WnF4c01OdEU3WDBwd1ZkcnY3T1lBTkUifQ.eyJhdWQiOiJhYjA3MTFjNC0yMDEzLTQ0MmMtYmY1Ny0wODhiOWYyNTg1ZTIiLCJleHAiOjE3MDk3NTM1OTEsImlhdCI6MTcwOTcxNzU5MSwiaXNzIjoiaHR0cHM6Ly9ncm93dGh6b25lLmZ1c2lvbmF1dGguaW8viwic3ViIjoiZDBiMzY3YjEtMDAzMi1jMTI1LTdkZTgtMGI0NTRjN2Y2NjFiIiwianRpIjoiZDE0MWFhNzAtZDZhMS00Nzk0LWI2ZTUtMTdkMjlkMzU4ZjM2IiwiYXV0aGVudGljYXRpb25UeXBlIjoiUEFTU1dPUkQiLCJlbWFpbCI6ImdyY3Rlc3RAdGVzdC5jb20iLCJlbWFpbF92ZXJpZmllZCI6dHJ1ZSwicHJlZmVycmVkX3VzZXJuYW1lIjoiZ3JjdGVzdEB0ZXN0LmNvbSIsImF0X2hhc2giOiJkX1NOOVlY4cXhVcksxdjZIbE5tNzVnViIiwic2NvcGUiOiJvZmZsaW5lX2FjY2VzcyBvcGVuaWQiLCJhdXRoX3RpbWUiOjE3MDk3MTc1OTEsInRpZCI6ImJmMzkwOWI0LWJjZGUtNGI4OC1iOTlkLWVkODk1NDAxMDU0NyIsImNvbnRleHRRYXRhbWV0ZXJzIjp7ImlzQXBpVXNlciI6ZmFsc2UsIm1lbWJlclN1aXRlVXNlcklkIjoiZDBiMzY3YjEtMDAzMi1jMTI1LTdkZTgtMGI0NTRjN2Y2NjFiIiwicGFydGl0aW9uS2V5IjozNTUyMywidXNlclR5cGUiOiJwb3J0YWwifSwidXNlck5hbWUiOiJncmN0

ZXN0QHRlc3QuY29tIiwidXNlcklkIjoiZDBiMzY3YjEtMDAzMi1jMTI1LTdkZTgtMGI0NTRjN2Y2NjFiIi
wiYXBwbGljYXRpb25JZHMiOlsiYWIwNzExYzQtMjFhMy00NDJjLWJmNTctMDg4YjlmMjU4NWUyIl
0sInBhcnRpdGlvbktleXMiOlszNTUyM119.Q8uNtAqOCyjtsmzRfTrFIDhCQ4Z6JX0CD8svSngs-
FRTP6KZumdRzOVPwrDXkryFWdrUA8UBrRNx_gcZf5B2xJZy24JhLd1hWS3iMH8hj7lOl-
P4L8txBelf7DifHEYjrD436DPzSabR6ob3W3tzDZi_rQDiasKC2WYnNPVuWF-
k2W6bjmqrByX7Arq8zlj4cnOqhqLmreyH7Sii4uj7Rdv0Mdsk73VFG4_L1PwAZZ-
sH9XroKKp7F7yt1iXEpkl8zhkXWY8Wedw--
tB5igECyeFPUvx2FCz82PLbhah0EcXBqVDSHlDgqBhqFTdZWjm9sOPFtrDY0RYfVXD

OfV59A' -d '{
"msql": "select Name, ID, ApplicableType, CreatedDate, CreatedBy.Name from SavedSearch
where (IsPublic=1 AND Name LIKE '%Active Members%') order by Name"
}' 'https://rest.membersuite.com/platform/v2/dataSuite/executeSearch'

Be sure to add "Bearer" and a space before the idToken string within the Authorization header,
as shown in the example cURL above.

Response Body: 200 Status Code

Example Response:
{
"success": true,
"data": [
{
"roW_NUMBER": 1,
"name": "Public Saved Active Members",
"id": "d0b367b1-001e-c919-bbd4-0b46864fccb9",
"applicableType": "Individual",
"createdDate": "2024-03-06T04:31:13",
"createdBy.Name": "gcarr"
}
],
"firstErrorMessage": "",
"errors": [],
"statusCode": 200
}

Important Notes:

Using the unique GUID from the id field above for each returned Public Saved Search from the MSQL lookup, use the next call below to execute that specific Public Saved Search and return the record results. Only one Public Saved Search ID can be run per request, however multiple requests one after the other against this endpoint are allowed.

The initial unique id GUID generated for each Public Saved Search at the time of creation in the Console does not change, even if that specific Public Saved Search is edited by the MemberSuite Client to add more fields, as long as the Public Saved Search is saved as an overwrite and the same name is kept.

If any existing applicable Public Saved Searches are deleted by the MemberSuite Client in the Console, the related unique id GUID would no longer exist. If a Public Saved Search is edited and saved under a new name, then the Console would generate a new initial unique id GUID, as it would be considered a brand new Public Saved Search.

Run a Unique Public Saved Search by id (GUID):

https://rest.membersuite.com/platform/swagger/ui/index#!/DataSuite/DataSuite_ExecuteSavedSearch

Executes the specific Public Saved Search by unique id and returns the record results

Input Parameters:

ID: GUID representing the specific public saved search Id.

Authorization: Bearer Token

Example cURL:

curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImd0eSI6WyJwYXNzd29yZCJdLCJraWQiOiJNYWx1WnF4c01OdEU3WDBwd1ZkcncY3T1lBTkUifQ.eyJhdWQiOiJhYjA3MTFjNC0yMWEzLTQ0MmMtYmY1Ny0wODhiOWYyNTg1ZTIiLCJleHAiOjE3MDk3NTM1OTEsImlhdCI6MTcwOTcxNzU5MSwiaXNzIjoiaHR0cHM6Ly9ncm93dGh6b25lLmZ1c2lvbmF1dGguaW8viwic3ViIjoiZDBiiMzY3YjEtMDAzMi1jMTI1LTdkZTgtMGI0NTRjN2Y2NjYiIiwianRpIjoiZDE0MWFhNzAtZDZhMS00Nzk0LWI2ZTUtMTdkMjlkMzU

4ZjM2IiwiYXV0aGVudGljYXRpb25UeXBlIjoiUEFTU1dPUkQiLCJlbWFpbCI6ImdyY3Rlc3RRdGVzdC5j
b20iLCJlbWFpbF92ZXJpZmllZCI6dHJ1ZSwicHJlZmVycmVkX3VzZXJuYW1lIjoiZ3JjdGVzdEB0ZXN0L
mNvbSIsImF0X2hhc2giOiJkX1NOVlY4cXhhVcksxdjZIbE5tNzVnIiwic2NvcGUiOiJvZmZsaW5lX2FjY2V
zcyBvcGVuaWQiLCJhdXRoX3RpbWUiOjE3MDk3MTc1OTEsInRpcCI6ImJmMzkwOWI0LWJjZGUtN
GI4OC1iOTlkLWVkODk1NDAxMDU0NyIsImNvbnRleHRQYXJhbWV0ZXJzIjp7ImlzQXBpVXNlciI6Zm
Fsc2UsIm1lbWJlclN1aXRlVXNlcklkIjoiZDBiMzY3YjEtMDAzMi1jMTI1LTdkkZTgtMGI0NTRjN2Y2NjFii
wicGFydGl0aW9uS2V5IjozNTUyMywidXNlclR5cGUiOiJwb3J0YWwifSwidXNlck5hbWU

iOiJncmN0ZXN0QHRlc3QuY29tIiwidXNlcklkIjoiZDBiMzY3YjEtMDAzMi1jMTI1LTdkkZTgtMGI0NTRj
N2Y2NjFiIiwiYXBwbGljYXRpb25JZHMiOlsiYWIwNzExYzQtMjFhMy00NDJjLWJmNTctMDg4YjlmMj
U4NWUyIl0sInBhcnRpdGlvbktleXMiOlszNTUyM119.Q8uNtAqOCyjtsmzRfTrFIDhCQ4Z6JX0CD8sv
Sngs-FRTP6KZumdRzOVPwrDXkryFWdrUA8UBrRNx_gcZf5B2xJZy24JhLd1hWS3iMH8hj7lOl-
P4L8txBelf7DifHEYjrD436DPzSabR6ob3W3tzDZi_rQDiasKC2WYnNPVuWF-
k2W6bjmqrByX7Arq8zlj4cnOqhqLmreyH7Sii4uj7Rdv0Mdsk73VFG4_L1PwAZZ-
sH9XroKKp7F7yt1iXEpkl8zhkXWY8Wedw--
tB5igECyeFPUvx2FCz82PLbhah0EcXBqVDSHlDgqBhqFTdZWjm9sOPFtrDY0RYfVXDOfV59A'
'https://rest.membersuite.com/platform/v2/dataSuite/executeSavedSearch/d0b367b1-001e-c919-bbd4-0b46864fccb9'

Be sure to add "Bearer" and a space before the idToken string within the Authorization header, as shown in the example cURL above.

Response Body: 200 Status Code

Example of Successful Response:

```
{
"success": true,
"data": [
{
"roWNUMBER": 1,
"localID": 10183,
"firstName": "Wynter",
"lastName": "Atkinson",
"title": null,
"emailAddress": "Wat@membersuite.com",
"primaryOrganization.Name": "MemberSuite Inc.",
```

"favoriteColorc": null,
"PreferredAddressLine1": "8410 Senectus Street",
"PreferredAddressLine2": null,

"PreferredAddressCity": "Oklahoma City",
"PreferredAddressState": "OK",
"PreferredAddressPostalCode": "63526",
"PreferredAddressCountryName": "UNITED STATES",

"membership.JoinDate": "2015-01-09T00:00:00",
"membership.ExpirationDate": "2018-12-31T00:00:00",
"membership.ReceivesMemberBenefits": true,
"dateOfBirth": "1981-01-04T00:00:00",
"classType": "INDIV",
"communicationsLastVerified": null,
"dateOfBirthMonth": 1,
"company": null,
"designation": null,
"id": "d0b367b1-0006-cbd5-f291-0b39829bd9e0"
},
{
"roWNUMBER": 2,
"localID": 10239,
"firstName": "Greg",
"lastName": "Carr",
"title": null,
"emailAddress": "gctest123@greg.com",
"primaryOrganization.Name": null,
"favoriteColorc": "Blue",
"PreferredAddressLine1": null,
"PreferredAddressLine2": null,
"PreferredAddressCity": null,
"PreferredAddressState": null,
"PreferredAddressPostalCode": null,
"PreferredAddressCountryName": null,
"membership.JoinDate": "2023-05-14T00:00:00",
"membership.ExpirationDate": "2027-04-30T00:00:00",
"membership.ReceivesMemberBenefits": true,
"dateOfBirth": null,

"classType": "INDIV",
"communicationsLastVerified": null,

"dateOfBirthMonth": null,
"company": null,
"designation": null,
"id": "d0b367b1-0006-cbed-9eb0-0b455d8cb95a"
}
],
"firstErrorMessage": "",
"errors": [],
"statusCode": 200
}

Example Response when Public Saved Search id does not exist:

{
"success": false,
"data": null,
"firstErrorMessage": "no SavedSearch found for savedSearchId d0b367b1-001e-c4c0-652a-0b468657e795",
"errors": [
"no SavedSearch found for savedSearchId d0b367b1-001e-c4c0-652a-0b468657e795"
],
"statusCode": 404
}

Universal MSQL Search

Run Any Generated MemberSuite MSQL Statement

This call allows you to run any generated advanced search MSQL statement. This is especially handy when you have a very complex advanced search that is created in and pulls from unique tables in MemberSuite and prevents you from having to utilize a specific GET call from our available swagger REST object GET calls.

https://rest.membersuite.com/platform/swagger/ui/index#!/DataSuite/DataSuite_ExecuteSearch

Fetches the data related to the provided MSQL.

MSQL can be generated via the MemberSuite console by selecting MSQL from the output format dropdown of an advanced search.

Authorization: Bearer Token

Complex MSQL Example:

select Individual.PrimaryOrganization.Name, Individual.FullName, Individual.Title, Roles from ChildEntities where (Parent='a68f867e-0007-c978-0bd3-0b465a9dab48' AND IsIndividual=1 AND Individual.PrimaryOrganization.Membership.ReceivesMemberBenefits=1 AND Individual.Status IN ('a68f867e-01f3-ca80-f146-0b3d19a154fd') AND Individual.Ind_OptOutofExternalDirectory___c='No') order by Roles

Example cURL:

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImd0eSI6WyJwYXNzd29yZCJdLCJraWQiOiJNYWx1WnF4c01OdEU3WDBwd1ZkcnY3T1lBTkUifQ.eyJhdWQiOiJhYjA3MTFjNC0yMWEzLTQ0MmMtYmY1Ny0wODhiOWYyNTg1ZTIiLCJleHAiOjE3MDk3NjYxODEsImlhdCI6MTcwOTczMDE4MSwiaXNzIjoiaHR0cHM6Ly9ncm93dGh6b25lLmZ1c2lvbmF1dGguaW8vIiwic3ViIjoiYTY4Zjg2N2UtMDAzMi1jNDYyLTY1YzAtMGI0NGFiNmU0OGFiIiwianRpIjoiOTk1NmI2NzItYzI1NC00Y2MzLTgxMTAtYjQ0MWVlMzRjNGEwIiwiYXV0aGVudGljYXRpb25UeXBlIjoiUEFTU1dPUkQiLCJlbWFpbCI6ImFwaXVzZXJAbWF0cml4Z3JvdXBmbV0iwiZW1haWxfdmVyaWZpZWQiOnRydWUsInByZWZlcnJlZF91c2VybmFtZSI6ImFwaXVzZXJAbWF0cml4Z3JvdXBmbV0iwiYXRfaGFzaCI6ImhDcUF4c3RVQld6M2ptN0VQWVJ3VVEiLCJzY29wZSI6Im9mZmxpbmVfYWNjZXNzIG9wZW5pZCIsImF1dGhfdGltZSI6MTcwOTczMDE4MSwidGlkIjoiYmYzOTA5YjQtYmNkZS00Yjg4LWI5OWQtZWQ4OTU0MDEwNTQ3IiwiY29udGV4dFBhcmFtZXRlcnMiOnsiXNBcGlVc2VyIjp0cnVlLCJtZW1iZXJJZTdWl0ZVVzZXJJZCI6ImE2OGY4NjdlLTAwMzItYzQ2Mi02NWMwLTBiNDRhYjZlNDhhZSIsInBhcnRpdGlvbktleSI6MzQ2OTIsInVzZ

XJUeXBlIjoicG9ydGFsIn0sInVzZXJOYW1lIjoiYXBpdXNlckBtYXRyaXhncm91cC5uZXQiLCJ1c2VySWQiOiJhNjhmODY3ZS0wMDMyLWM0NjItNjVjMC0wYjQ0YWI2ZTQ4YWUiLCJhcHBsaWNhdGlvbklkcyI6WyJhYjA3MTFjNC0yMWEzLTQ0MmMtYmY1Ny0wODhiOWYyNTg1ZTIiXSwicGFydGl0aW9uS2V5cyI6WzM0NjkyXX0.g7e_Umgl-5PLsjJU792-VpljbWyCwHfYENuwM9zWLLNfK6lHj4o6BykjKXJRNCLZo6P-lnBOOvq524B-DTObfy43cocj67zcFt4lapp0R_TGJV89W-3vt-

6yA6qW2Hltj2QJKbqyLBksUIdwOp5UvJq1WgyD1B7vMZTaduPGLdhU7CSBVV_6CeBpMUQTnpq
oMrB71_w6kGAZnHf_sAm53sF6ImYh527R9Q39tujXGNM5Bx-gs6oPXTdydRlRB-qS75-
1as4oJrUHYg8Lv58mTe5MEeEKlKrZeaLe49ql8bYnVydDs62fbb_-
p4NdPtFybilzWoMxeyINJjL02U0lQQ' -d '{

"msql": " select Individual.PrimaryOrganization.Name, Individual.FullName, Individual.Title, Roles from ChildEntities where (Parent='a68f867e-0007-c978-0bd3-0b465a9dab48' AND IsIndividual=1 AND Individual.PrimaryOrganization.Membership.ReceivesMemberBenefits=1 AND Individual.Status IN ('a68f867e-01f3-ca80-f146-0b3d19a154fd') AND Individual.Ind_OptOutofExternalDirectory__c='No') order by Roles"

}' 'https://rest.membersuite.com/platform/v2/dataSuite/executeSearch?startPosition=0'

Be sure to add "Bearer" and a space before the idToken string within the Authorization header, as shown in the example cURL above.

Response Body: 200 Status Code

REST SSO Options

Option 1: Outside SSO (Third Party Login Page)

Extra info if using PHP

MemberSuite SDK-PHP Repository (Updated 9/18/23 to accommodate new versions of PHP):

https://github.com/membersuite/sdk-php

MS Production REST API Credentials:

Tenant ID/Partition Key: (Required)

If using Postman to test cURLs, please be sure to turn off automatic redirects within the Postman settings.

POST /platform/v2/loginUser/{tenantId}

Pass the User's Login Creds in the payload:

```
{
  "email": "string",
  "password": "string"
}
```

Example cURL:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{
  "email": "string",
  "password": "string"
}' 'https://rest.membersuite.com/platform/v2/loginUser/XXXXX'
```

A successful response will give you three Token Strings.

Then use the three tokens returned in this follow up call:

POST /platform/v2/JWTSSO/{tenantId}

Example cURL:

```
curl --location --request POST 'https://rest.membersuite.com/platform/v2/JWTSSO/[tenantID]' \
--header 'Content-Type: application/x-www-form-urlencoded' \
```

```
--data-urlencode 'idToken=string' \
```
```
--data-urlencode 'refreshToken=string' \
```
```
--data-urlencode 'accessToken=string' \
```
```
--data-urlencode 'nextUrl=https://dev.tradewing-dev.com/api/v1/sso' \
```
```
--data-urlencode 'IsSignUp=false' \
```
```
--data-urlencode 'IsForgotPassword=false'
```

Important Note about IsSignUp in the command above:

If IsSignUp is true = New User SSO

If IsSignUp is false = Existing User SSO

If tokens are valid, the REST API POST call from above will return a URL in the response location field in header that includes a unique tokenGUID for that user:

Example: https://xxxx.users.membersuite.com/auth/jwt-sso?tokenGUID=10553ba0-afcc-4489-90ff-f94b6d25d2e0&nextUrl=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbbQ

The tokenGUID query string value within the response URL shown above is a valid unique tokenGUID for the specific user that attempted to log in that you would need to programmatically capture. Please note, this tokenGUID is only valid for 5 minutes.

Redirecting the specific user to this response URL will create/set an MRP session and forward to the nextURL set above.

The nextURL set above is an external page you create where you would run the additional API calls below to identify the specific user and log them in programmatically to your external site. It is the responsibility of you the developer to decipher the response data and log them into the external site as you see fit by using additional external coding.

Using the valid unique tokenGUID, the specific user can be authenticated on a non-MemberSuite platform by using this endpoint:

GET / JWTSSO

Example cURL:

curl -X GET --header 'Accept: application/json' 'https://rest.membersuite.com/platform/v2/bearerTokenSSO?tokenGUID=590f83e6-1f3e-4704-8066-95bfce312966&partitionKey=XXXXX'

An unsuccessful authentication will return a response with an error. A successful authentication response body should return an idToken String.

From here, the idToken String can be used to retrieve basic user information using this endpoint:

GET /platform/v2/whoami

Example cURL:

curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer [idToken]' 'https://rest.membersuite.com/platform/v2/whoami'

Be sure to add "Bearer" and a space before the idToken String within the Authorization header, as shown in the example cURL above.

An example response looks like:

```
{
  "tenantId": integer,
  "associationId": string (guid),

  "userId": string (guid),
  "email": string,
  "firstName": string,
  "lastName": string,
  "ownerId": string (guid),
  "membershipId": string (guid),
  "receivesMemberBenefits": boolean,
  "paymentProcessor": string,
  "businessUnit": string (guid),
  "merchantAccount": string (guid),
  "baseCurrency": string,
  "username": string
}
```

The receivesMemberBenefits field determines if they are a member or not:

true:

If a user has an active Membership - receivesMemberBenefits would be true (Member)

null/false:

If a user has never had a membership - receivesMemberBenefits would be null (Non-Member)

If a user had a membership that expired and are dropped - receivesMemberBenefits would be false (Non-Member)

More specific information can be returned by searching using the ownerId field from the response as criteria in this call:

GET /crm/v1/individuals/{tenantID}

Example cURL:

curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer [idToken]' 'https://rest.membersuite.com/crm/v1/individuals/XXXXX?msql=[MSQL Statement where ID='ownerId']'

Be sure to add "Bearer" and a space before the idToken String within the Authorization header, as shown in the example cURL above. MSQL can be generated via the MemberSuite console by selecting MSQL from the output format dropdown of an individual advanced search.

Other REST API GET calls can be made by using the same idToken String in this active session as authorization. The idToken String is valid for 1 hour.

Option 2: Reverse SSO (Common)

Extra info if using PHP

MemberSuite SDK-PHP Repository (Updated 9/18/23 to accommodate new versions of PHP):

https://github.com/membersuite/sdk-php

MS Production REST API Credentials:

Association ID: (Required)

Tenant ID: (Required)

Using MemberSuite MRP login page for an external website is Reverse SSO

If using Postman to test cURLs, please be sure to turn off automatic redirects within the Postman settings.

This user flow is used when the MemberSuite portal is used as the single point of sign-on. The user will be directed to login on the MemberSuite portal login page before being forwarded to the nextURL with SSO context. The endpoint to use is:

POST /platform/v2/signUpSSO

Example cURL:

curl --location --request POST 'https://rest.membersuite.com/platform/v2/signUpSSO' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'nextUrl=https://dev.tradewing-dev.com/api/v1/sso' \
--data-urlencode 'IsSignUp=false' \
--data-urlencode 'AssociationId=dcd44b35-0004-c0f8-531c-0b3b287bd8ea'

Important Note about IsSignUp in the command above:

If IsSignUp is true, the user gets sent to the MS Login creation page

If IsSignUp is false, the user gets sent to the MS Login page

The REST API's response should have a Location field in the header that stores a URL. The user's just need to be redirected to this single URL and the system will start the SSO process. From here, the user will be prompted to log into the MemberSuite portal. After the credentials are authenticated, a MemberSuite session will be created before redirecting the user to the provided nextUrl with the SSO context.

Note: The signUpSSO call above only needs to be run once per nextURL passed to obtain that specific Location URL for SSO.

An example response Location URL looks like this:

https://[acronym].users.membersuite.com/auth/portal-login?isSignUp=false&nextUrl=aHR0cHM6Ly93d3cubGF0bG1cy5jb20vYnJlYWtpbmcvYW1hemluZy1uZXXctdGVjaC0x&isReverseSSO=true

The nextURL set above is an external page you create where you would run the additional API calls below to identify the specific user and log them in programmatically to your external site. It is the responsibility of you the developer to decipher the response data and log them into the external site as you see fit by using additional external coding.

When the user is redirected back to the nextURL, there will be a tokenGUID query string parameter added. Please note, this tokenGUID is only valid for 5 minutes.

Example: https://dev.tradewing-dev.com/api/v1/sso?tokenGUID=00d9c818-3306-4eb8-8855-29a0ddd946a4

Using this tokenGUID value, the user can be authenticated on a non-MemberSuite platform by using this endpoint:

GET /platform/v2/regularSSO

Example cURL:

curl --location --request GET 'https://rest.membersuite.com/platform/v2/regularSSO?partitionKey=[tenantID]&tokenGUID=[tokenGUID]'

An unsuccessful authentication will return a response with an error. A successful authentication response body should return a Token String like this:

```
{
    MaiapmkmRRw/BKE4HsvDpT1bHRjfuJ0qW+a0DXITe/kr2SDHcNamIaaB[....]
}
```

From here, the Token String can be used to retrieve basic user information using this endpoint:

GET /platform/v2/whoami

Example cURL:

curl -X GET --header 'Accept: application/json' --header 'Authorization: AuthToken [TokenString]' 'https://rest.membersuite.com/platform/v2/whoami'

Be sure to add "AuthToken" and a space before the Token String within the Authorization header, as shown in the example cURL above.

An example response looks like:

```
{
  "tenantId": integer,
  "associationId": string (guid),
  "userId": string (guid),
  "email": string,
  "firstName": string,

  "lastName": string,
  "ownerId": string (guid),
  "membershipId": string (guid),
  "receivesMemberBenefits": boolean,
  "paymentProcessor": string,
  "businessUnit": string (guid),
  "merchantAccount": string (guid),
  "baseCurrency": string,
  "username": string
}
```

The receivesMemberBenefits field determines if they are a member or not:

true:

If a user has an active Membership - receivesMemberBenefits would be true (Member)

null/false:

If a user has never had a membership - receivesMemberBenefits would be null (Non-Member)

If a user had a membership that expired and are dropped - receivesMemberBenefits would be false (Non-Member)

More specific information can be returned by searching using the ownerId field from the response as criteria in this call:

GET /crm/v1/individuals/{tenantID}

Example cURL:

curl -X GET --header 'Accept: application/json' --header 'Authorization: AuthToken [TokenString]' 'https://rest.membersuite.com/crm/v1/individuals/XXXXX?msql=[MSQL Statement where ID='ownerId']'

Be sure to add "AuthToken" and a space before the Token String within the Authorization header, as shown in the example cURL above. MSQL can be generated via the MemberSuite console by selecting MSQL from the output format dropdown of an individual advanced search.

Other REST API GET calls can be made by using the same Token String in this active session as authorization. The Token String is valid for 1 hour.