Cirrus Data API

After getting your Client ID and Client Secret from Springbrook you can request a token and utilize the API. At the start of each session, you'll need to request a new access token if you don't already have an unexpired token. Springbrook recommends caching access tokens for re-use and only requesting new tokens when necessary.

Once you have a token, you'll be ready to call the API to access your data.

Requesting a Token

Make a `POST` request to the token endpoint `https://springbrook-prod.us.auth0.com/oauth/token` with the JSON body below:

```
{
    "client_id": "{Client ID}",
    "client_secret": "{Client Secret}",
    "audience": "https://cirrus-data-api.prod.springbrooksoftware.com",
    "grant_type": "client_credentials"
}
```

Be sure to replace `{Client ID}` and `{Client Secret}` with the values you received from Springbrook. You'll also want to make sure you're sending a `Content-Type` request header with the value `application/json`.

Token Response

A successful token request will result in a status code 200 OK response with a JSON body like the example below:

```
{
  "access_token": "{Token}",
  "expires_in": 86400,
  "token_type": "Bearer"
}
```

The value returned in the `access_token` field, represented by `{Token}` in the example above, is what you'll use to make requests to the Cirrus Data API.

Calling the Data API

The Data API is located at `https://data.springbrooksoftware.com`. Each route you see in this document assumes this URI as the base. For example, the route `/api/documentation/entities` should be called as `https://data.springbrooksoftware.com/api/documentation/entities`. With every request you make to the Data API you'll need to provide an `Authorization` header and a `shardId` header.

Authorization Header

The `Authorization` header should contain a value of `Bearer {Token}`, where `{Token}` is the value received from the token request outlined above.

shardId Header

The `shardId` header should contain the shard of the database you are requesting data

from. This value is provided to you by Springbrook during initial set-up.

Getting Information about Entities
The entities available via the API can be retrieved by the *Entities Documentation* endpoint, while the properties each entity has can be retrieved via the *Entity Properties Documentation* endpoint.

Entities Documentation
Request
`GET` `/api/documentation/entities`
Response
```
[
  "Ap1099Prints",
  "ApChecks",
  ...
]
```
Entity Properties Documentation
Request
`GET` `/api/documentation/entities/{entity}`
Response
```
[
  "ApCheckId",
  "ApVendorId",
  ...
]
```

Getting Data from the API
Make **`GET`** requests to `/api/{entity}` where `{entity}` matches one of the values returned via the *Entities Documentation* endpoint to retrieve entity data from the database.

Filtering Results
Entity data can be filtered by specifying the `$filter` query parameter and a filter string.
Example 1
```
/api/glJournalEntries?$filter=fiscalYear eq 2021
```
Example 2
```
/api/glJournalEntries?$filter=system eq 'PR'
```

Selecting Properties
Selecting specific entity properties can be accomplished by specifying the `$select` query parameter and the desired properties.
Example

```
/api/glJournalEntries?$select=fiscalYear,fiscalPeriod,journalEntry
```

Sorting Results
Results can be sorted by specifying the `$orderBy` query parameter and the properties to sort by.
Example
```
/api/glJournalEntries?$orderBy=fiscalYear DESC,fiscalPeriod DESC,journalEntry
```

Limiting the Number of Results
Use the `$top` query parameter to limit the number of results returned.
Example
```
/api/glJournalEntries?$top=1
```

Skipping Results
Use the `$skip` query parameter to exclude a number of results. This can be combined with the `$top` parameter to accomplish pagination.
Example
```
/api/glJournalEntries?$top=10&$skip=1
```