WE HELP SCHOOLS **WORK SMART**

SENTRAL API

Quickstart Guide

Version 1.0

04/12/2019

SENTRAL

Document Control

Title: ***Sentral API Quickstart Guide***

Issue: 1.0

Issue Date: 04/12/2019

Document Approval

Version No.	Date	Updated By	Change History
1.0	04/12/2019	James Hamilton	Initial Document

TABLE OF CONTENTS

Document Control 2
1. Accessing The API 4
1.1. Deployment 4
1.2. URL 4
1.3. Versioning 4
1.4. Authentication 4
1.5. Tenant Key 4
1.6. Example CURL Request 4
2. Using the API 5
2.1. API Format 5
2.2. Schema 5
2.3. Pagination 5
2.4. Includes 5

1. Accessing The API

1.1. Deployment

The API is available as part of the regular deployment of the Enterprise version of Sentral. Any site running on this version can have access to the API upon request. As the API is treated as a module of Sentral it will need to be enabled before it can be accessed.

1.2. URL

The API is accessed via HTTP calls to the /restapi/ path on the same host as the rest of Sentral. As with all access to Sentral, the API can only be accessed via secure HTTP calls. From there the version of the API is specified and then the specific endpoint is appended to the URL. e.g. https://demo.sentral.com.au/restapi/v1/sentral/schema.

1.3. Versioning

The API is designed to support multiple concurrent versions. There is a contract that the endpoints will never have breaking changes within a specific version. New endpoints may be added, and existing endpoints may have new attributes or functionality added within the same version. Any changes that will remove or fundamentally change the functionality of the endpoint will be deferred until the next version of the API. The version of the API to access is done via the URL, i.e. the first sub path after the /restapi/ defines the version. In the example URL https://demo.sentral.com.au/restapi/v1/sentral/schema the /v1/ sub path defines that this is a call to version 1 of the API.

1.4. Authentication

Authentication is actioned via API Keys. Once provisioned, an API key must be provided for all requests to the API. This is done by providing the API as the X-API-KEY request header.

1.5. Tenant Key

In addition to the API Key, a tenant key must be provided as the X-API-TENANT request header. This key is the same key as what is used in the URL when accessing tenanted modules within Sentral. E.G. when accessing a tenants attendance module the URL would be https://demo.sentral.com.au/s-ABCDE/attendance/, in this case the tenant key is ABCDE. When accessing data for tenanted modules, the tenant key is used to determine which tenant to access. Note that the tenant key is still required even when accessing non tenanted modules such as Enrolments

1.6. Example CURL Request

CURL or a similar program for making HTTP requests (such as Postman) can be used to test access to the API. The following is an example request made to the API. To use this against a real Sentral installation the base URL, and the two headers will need to be replaced with actual values

```
curl -H "X-API-KEY: 354fe888-16af-4a84-af95-94e3640e25b5" -H "X-API-TENANT: ABCDE" https://demo.sentral.com.au/restapi/v1/sentral/schema
```

2. Using the API

2.1. API Format

The API is formatted using the JSON API specification. This specification is documented at https://jsonapi.org/ .

2.2. Schema

The endpoints available for the API are defined at the /restapi/v1/sentral/schema endpoint. This endpoint is automatically updated whenever new endpoints are created, or existing endpoints are updated. Alternatively the schema can be viewed in a more human readable format at https://development.sentral.com.au/ . This documentation is periodically updated with the latest version of the api.

2.3. GET requests

When making GET requests to the API. The resultant response will be a JSON object as defined in the JSON API specification with certain sections presented. The data requested will be available in the data section of the response. Depending on the call made and any parameters sent, additional sections may be presented. Example responses for specific endpoints are available in the API schema.

2.4. Pagination

Calls to endpoints that will return multiple results are automatically paginated to ensure that responses are delivered in a timely manner. Whenever a response is received, the count attribute in the meta section of the response will return the total amount of records. The additional records can be retrieved by using the limit and offset GET parameters to make additional calls to the endpoint. The limit parameter controls how many records to return. The offset parameter controls how many records to skip. In addition, the response includes the URL attributes for the first, last and next call in the sequence in the links section of the payload. Attempting to retrieve too many results at once using the limit parameter will return an error that describes the current server limit. Attempting to retrieve records after the maximum number of records will return an empty data set. Following is an example CURL request to get the second page of 10 enrolment student records.

```
curl -H "X-API-KEY: 354fe888-16af-4a84-af95-94e3640e25b5" -H "X-API-TENANT: ABCDE"
"https://demo.sentral.com.au/restapi/v1/enrolments/student?limit=10&offset=10"
```

2.5. Includes

Sometimes it is necessary to retrieve related data for an endpoint. Rather than making additional calls, specific endpoints define available includes that can be added to the return payload by specifying the include GET parameter. Multiple includes can be specified by using commas to separate the data to include. The included data is then presented in the payload under the included section. Available includes for a specific endpoint are defined in the API Schema. Following is an example CURL request to get the related primary enrolment and flag data for an enrolment student in one call.

```
curl -H "X-API-KEY: 354fe888-16af-4a84-af95-94e3640e25b5" -H "X-API-
TENANT: ABCDE"
"https://demo.sentral.com.au/restapi/v1/enrolments/student?include=pri
maryEnrolment,flags"
```

2.6. POST and PATCH requests

When making POST and PATCH requests, the API expects the data to be presented in the JSON API format. This will mean sending the data as a JSON object with certain sections and attributes defined. The exception to this is when binary data is to be sent, for example, medical change request care plan files. In these cases the binary data should be sent as multipart/form-data requests. Example request payloads for specific endpoints are available in the API schema.

End of Document