Student Registration for Lingco.io

We like to pre-register students for the ATTG with the new test provider Lingco.io to streamline the registration process. The main issue with the process implemented by Lingco is that it requires students to register with name, email, and password. These are three opportunities to make a mistake that we like to avoid.

Lingco does not support bulk uploads. However, we reversed engineered the REST APIs used by the Lingco registration page and determined that we can implement a bulk load solution.

# Solution Approach

The solution will work as follow:

1. We extract Sycamore a spreadsheet that contains student first and last name, student.gssb.org email, and a Lingco password (TBD how to construct it).
2. A new Python tool loads the spreadsheet and registers students by issuing POST statements to the Lingco registration URL.
3. Students will login their student.gssb.org email and click the confirmation link.

At his point the student is registered with Lingco and can login to the service using the student.gssb.org email and the provided password.

# Lingco REST API

I found several REST APIs that support the Lingco Registration UI. For our purposes only one POST is required and optionally a GET request can be used to retrieve an ID.

The core or the REST API is a POST to 'https://class.lingco.io/api/users/register’ that submits a JSON object with the following details:

* Student first name
* Student last name
* Student email
* Password
* Language instruction selection (hardcoded to “de”)
* Account type (hard-coded to “student”)
* Missing School URL (hard-coded to [www.gssb.org](about:blank)) und
* Organizations ID (hard-coded to 40427 = unknown organization)

The POST must set headers

* 'Accept: application/json, text/plain, \*/\*'
* 'Content-Type: application/json'

The successful request returns with HTTP status code 201 (created) and a message {"msg":"User successfully created"}.

If the use already exists, the request returns with HTTP status code 409 (conflict) and a message {"msg":"Email already used"}.

The following curl statement creates a user with name “Test3 Test” and email [test3@test.com](about:blank) for missing school with URL www.test.com:

curl -X POST -k -H 'Accept: application/json, text/plain, \*/\*' -H 'Content-Type: application/json' -i 'https://class.lingco.io/api/users/register' --data '{"first\_name":"Test3","last\_name":"Test","language":"de","email":"test3@test.com",

"account\_type":"student","instructor\_code":"","password":"abcd1234",

"missing\_school\_url":"www.test.com","organization\_id":40427,

"registration\_token":"","course\_code":null}'

The spreadsheet will be based on the spreadsheet that we generate for registration and AATG. It already defines the student first name, last name, and gssb student email address.

In addition, we will add a new column called LINGCO\_PWD that will contain the generated password for Lingco.

## Supplemental REST API for Unknown Organisation

The ID for an unknown organization may be variable. I found that the GET [https://class.lingco.io/api/organizations/not\_found](about:blank) returns a JSON object that defines the domain element “ID”, which is currently the number 40427.

It is desirable to use the ID retuned by the eservice instead of hard-coding the value.

curl -X GET -k -H 'Accept: application/json, text/plain, \*/\*' -H 'Content-Type: application/json' -i 'https://class.lingco.io/api/organizations/not\_found’

returns

{"city":"","created\_on":"2021-09-24T12:48:15Z", "default\_courses\_to\_free":false, "domain\_restricted":false, "domains":[], **"id":40427**, "license\_type":null, "name":"No School Found", "short\_name":"", "state":"","zip\_code":""}

# Testing Setup

We need to run some tests to validate that new students can be added and existing students are not re-registered. We do not want to use GSSB student emails for this purpose.

Instead, I have created a new email [developer@gssb.org](about:blank) that we can use as the basis for the test. Password is deve!0p@r. We will deactivate after the tests.

Oskar suggested that we attach additional information after the email address and a “+”.

For example, you can send me an email to [developer+Test1.Test@gssb.org](about:blank) (for student with first name Test1 and last name Test) and it will be delivered to [developer@gssb.org](about:blank). We can create endless number of email addresses for test purposes that are directed in one email box.

My suggestion is that for test purposes we mostly use [www.test.com](about:blank) for the name of the missing school URL to keep the association with GSSB to a minimum.