



# **Technical Integration Guide for Students' Unions**

Version 4.7    22-Feb-2017

## Document History

Version	Description	Author	Date	Status
0.1	Initial version for review.	Charles Bernasconi	06-Mar-2012	Draft
0.2	Initial draft for external release	Charles Bernasconi	19-Mar-2012	Draft
0.3	Added error messages	Charles Bernasconi	05-Apr-2012	Draft
0.4	Added outline of SOAP scheduled upload	Charles Bernasconi	20-Apr-2012	Draft
1.0	Published for circulation incorporating final editing, design UI templates & finance extracts	Doug Tomlinson	14-May-2012	Issue
1.1	Updated Section 3.2 and 3.4	Shoba Menon	23-Jul-2012	Issue
1.2	Added FE study type	Charles Bernasconi	17-Sep-2012	Issue
1.3	Made course end_date mandatory, added validations for names. Added CSV details.	Charles Bernasconi	03-Oct-2012	Issue
1.4	Changed Scotland ISO 2 code from SD to SW as duplicate with Sudan	Charles Bernasconi	24-Oct-2012	Issue
1.5	Added section on CSV upload	Aliya Aziz	06-Dec-2012	Draft
2.0	Updated CSV upload section	Aliya Aziz	18-Mar-2013	Draft
3.0	Updated the Shibboleth section	Shoba Menon	20-Mar-2014	Issue
4.0	Added the REST upload	Shoba Menon	15-Jun-2015	Issue
4.1	Modified the REST upload	Shoba Menon	18-Aug-2015	Issue
4.2	Updated section 3.1 overview	Shoba Menon	11-Sep-2015	Issue
4.3	Updated section 3.4 and 3.5	Shoba Menon	7-Jan-2016	Issue
4.4	Updated the endpoint protocol for the SOAP/REST(json)	Shoba Menon	18-Apr-2016	Issue
4.5	Updated the fields and validation for student data feed	Shoba Menon	25-Jul-2016	Issue
4.6	Updated with the rate limit details	Shoba Menon	11-Aug-2016	Issue
4.7	Updated with ISO country code for XC, XA, QO	Vedprakash Singh	22-Feb-2017	Issue

## Contents

<b>1</b>	<b>Introduction.....</b>	<b>4</b>
1.1	Overview	4
<b>2</b>	<b>Streamlining secure student access via Single Sign On .....</b>	<b>5</b>
2.1	Overview	5
2.2	Shibboleth	5
<b>3</b>	<b>Managing student data between university &amp; UnionCloud.....</b>	<b>6</b>
3.1	Overview	6
3.2	Fields in data feed	7
3.3	SOAP Web Service	11
3.4	REST Service	12
3.5	Data Feed Error List	19
	<b>Appendix A – Supported ISO Country Codes.....</b>	<b>22</b>

## 1 Introduction

### 1.1 Overview

UnionCloud is a bespoke, flexible and modular web site development and membership management system for students' unions. Sitting on the NUS Digital infrastructure, NUS' national database and digital platform for the student movement, it allows students' unions to rapidly set up and manage their own web site without requiring technical knowledge.

This document provides information for students' unions and their corresponding institutions on how to configure UnionCloud and connect to the NUS Digital technical platform.

There are two key technical aspects to integration:

- Secure student authentication (especially important for e-voting)

- Student data feeds from institution to NUS Digital

For each of these there are various approaches possible. The approach taken will depend on what kind of system the institution has and data is available, the level of integration required and data sharing agreements in place between the union and institution.

## 2 Streamlining secure student access via Single Sign On

### 2.1 Overview

Many educational institutions provide a Single Sign On (SSO) facility to authenticate students and staff members across a variety of their websites. This is particularly advisable to ensure a secure voting process.

The SSO process can be extended to incorporate NUS Digital by using a third party, commonly used service called Shibboleth. Thus students will not have to use an additional set of credentials to access the union site.

Shibboleth integration is fully supported within the NUS Digital platform. Other SSO mechanisms such as OAuth2 may also be supported in the future but are not supported at the present time.

A user being authenticated via SSO will not necessarily give them access to protected areas of the NUS Digital platform. Once they are authenticated, NUS Digital checks if they are registered, and if not asks them to register. If already registered the user's authorisation rights are determined by what is held in NUS Digital.

Users who are not students of the union can access the local union's website with a 'guest' account, where their own email/password authentication can be used.

### 2.2 Shibboleth

To implement this service, a union administrator / University technical team must provide configuration details required for Shibboleth single sign on at the time of the Union's creation through UnionCloud or before they wish to Go-live with the SSO.

The following manual process is required for the Shibboleth Service provider configuration:

- Request UK Federation to add the endpoints for new Union.

- Configure Shibboleth configuration for new Union.

- Test in our 'UAT' / 'Staging' env with test account if provided by the union.

Once the above manual process is configured, the user will be redirected to a corresponding configured Shibboleth page.

### 3 Managing student data between university & UnionCloud

#### 3.1 Overview

Some pieces of NUS Digital functionality are dependent on the user (i.e. a student) data which is held by the institution. Some other pieces of functionality give an enhanced user experience if this data is available.

eVoting – To enable to determine if a user is eligible to vote in an election NUS Digital needs to know their status. Depending on the type of election this may just be if they are a current student, but for other elections such as course reps, the student's course will be required.

NUS Digital Groups membership – depending on the union's rules, NUS Digital may need to know if a student is eligible for membership. E.g. Some groups may only be open to current students, just post-graduate students, over 18s or just students living in a particular university accommodation.

Events – depending on the event type, NUS Digital may need to know if a student is eligible to attend or buy tickets in a similar way to NUS Digital Groups criteria.

As a minimum, knowing if a user is a current student is required for elections. Three alternative mechanisms are provided to allow the data feed.

The following methods are currently available for data integration:

SOAP Web Server Scheduled Upload – the institution uploads all student records to NUS Digital on a scheduled basis. The details of new users, amends or deletes to existing users can be pushed via this method. It is advisable usually to only push the changed records after the initial student load to NUS Digital as the uploads get processed quicker in this case.

CSV Manual Upload – a union administrator manually uploads a CSV file containing student records/programme data and group memberships via the admin site. The fields required and validations are the same as the SOAP methods. CSV header and first data row is validated. If valid, the file is uploaded. Midnight scheduler picks the uploaded CSV and processes the data. If any row within file is invalid, only that particular row is rejected. After data is processed, error report is generated. Error report contains only those rows that were rejected while data processing. Administrator can login to Unioncloud and check the error report; correct data and upload again. Note that the best way to facilitate error correction is to re-use the error report file, correctly amend and re-upload it so that the upload is processed quicker.

REST Push/Scheduled Upload –The institution would upload all/updated student records to NUS Digital on a scheduled basis. The details of new users, amends or deletes to existing users can be pushed via this method. It is advisable usually to only push the changed records after the initial student load to NUS Digital as the uploads get processed quicker in this case

**Note:** From the system's point of view, a student is searched with the provided university id (id in SOAP request) and institution email address. If a match is found on

either field, then the record is updated, rather than creating a new record. This happens irrespective to the value provided in the record\_type of SOAP request/CSV file/JSON.

## 3.2 Fields in data feed

### 3.2.1 Student data

Regardless of the transfer mechanism, the following fields are sent for each student.

Field name	Required?	Description
id	Y	Unique student identifier. If regex is given during setup, then university id must match this regex.
forename	Y	Special characters ? * ! @ # \$ % ^ & * ( ) < > /   { } [ ] ; , \ : " are not allowed. Accented names are accepted.
surname	Y	Special characters ? * ! @ # \$ % ^ & * ( ) < > /   { } [ ] ; , \ : " are not allowed. Accented names are accepted.
dob	Y	Required. In the format dd/MM/yyyy e.g. 18/03/1990
gender	N	Required if there are gender related eligibility criteria. Accepted input: 'M' – Male 'F' – Female 'N' – Non-Binary 'O' – Other
institution_email	Y	Must be a valid e-mail address
nationality	N	Required if demographic analysis of this is required. 2 or 3 character ISO code. ISO extensions for England, Scotland, Wales, Northern Ireland are supported. See Appendix A for supported values.
domicile_country	N	Country of residence of student when applying for the institution. This can be used as an alternative to nationality if nationality cannot be provided for data protection reasons. Required if demographic analysis of this is required. 2 or 3 character ISO code. ISO extensions for England, Scotland, Wales, Northern Ireland are supported. See Appendix A for supported values.
fee_status	N	Required if there are eligibility criteria based on this. Accepted input: 'UK' – UK fee status 'EU' – EU fee status 'IN' – International fees status
hall_of_residence	N	Can be a code or a name of the hall of

		residence. Required if there are eligibility criteria based on this.
programme_id	N	Unique identifier for the student's course of study. Ideally as granular as possible. Required if NUS Digital is to be used for educational rep elections. E.g. course rep, department rep etc. If provided the student is placed in a user_group corresponding to the programme. A separate feed of programmes is provided as mentioned in 3.2.2 .
study_type	N	Required if there are eligibility criteria based on this Accepted input: 'FE' – Further education 'UG' – Undergraduate 'PG' – Postgraduate 'PGT' – Postgraduate taught 'PGR' – Postgraduate research 'CPD' – Continual professional development 'UGM' - Undergraduate masters 'MPH' - Master of philosophy 'TES' - Temporary Exchange Socrates
programme_level	N	E.g. 1,2,3. Indicates 1 <sup>st</sup> year, 2 <sup>nd</sup> year etc. Required if there are eligibility criteria based on this. Must be >=0.
start_date	N	Course start date. In the format dd/MM/yyyy e.g. 01/09/2013
end_date	Y	Course end date. Must be in the future. In the format dd/MM/yyyy e.g. 05/05/2016
record_type	Y	Required to determine the action to be performed for the student data Accepted input: 'New' – New student record. Regarded as 'update' if the student already exists in the system 'Update' – Updating existing student record. Regarded as 'new' if the student doesn't exist in system 'Temp_delete' – Temporary deletion of student record – just moves the student from Confirmed list to Deleted list 'Permanent_delete' – Permanent deletion of student records – Soft delete of student. All membership expires with immediate effect.  A student is identified by their university ID or their university e-mail address.
alternate_email_address	N	Any email other than the mentioned email
library_card	N	Library Card Number of the student or Mifare number of the student card. eg. L54321. If regex is given during setup, then value for this field must match with this regex.



department	N	Department Code. eg. EL, PS
erasmus	N	Denotes whether the student is a temporary exchange student from another country. Valid values for this field are Y or N
ethnicity	N	Ethnicity of Student. This is useful for demographic analysis when using Analytics. eg. African, European, etc
finalist	N	Denotes whether the student is in the final year of the course. Valid values are Y or N
mode_of_study	N	Valid values are 'Full-Time' or 'Part-Time'
placement	N	Denotes whether the student is either studying abroad or on a year in industry as part of their course. Valid values for this field are Y, N, R, P
address	N	Address of the student. Free form text.
postcode	N	Valid UK Post codes. eg: SK102NK
additional_identities	N	Applicable to only SOAP/JSON upload

### 3.2.2 Programme data

This optional data feed or CSV upload can specify additional attributes for programme (i.e. course data). Without this data, the user\_group names that students are placed into for programmes will use the programme\_id for the name e.g. "A234". Once this is supplied (which can be done retrospectively) the names will change to use the programme\_name field e.g. "BSc Maths".

Field	Required?	Description
id	Y	Unique programme identifier
programme_name	Y	e.g. French with Dentistry, AS-Level History
jacs_code	N	Optional JACS code
school_name	N	e.g. School of Dentistry
faculty_name	N	e.g. Faculty of Medicine
study_type	N	<b>'FE'</b> – Further education <b>'UG'</b> – Undergraduate <b>'PG'</b> – Postgraduate <b>'PGT'</b> – Postgraduate taught <b>'PGR'</b> – Postgraduate research <b>'CPD'</b> – Continual professional development <b>'UGM'</b> - Undergraduate masters <b>'MPH'</b> - Master of philosophy <b>'TES'</b> - Temporary Exchange Socrates
record_type	Y	Required to determine the action to performed for the programme data <b>'New'</b> – New programme record. Regarded as 'update' if the programme (identified by id) already exists in the system <b>'Update'</b> - Updating existing programme record. Regarded as 'new' if the programme doesn't exist in system <b>'Delete'</b> - Soft delete of the programme

### 3.2.3 Group join data

This optional data feed (happens via fresher fair android app only) or CSV upload can specify which groups a student has joined. A group membership may or may not require next of kin to be filled in.

Field	Required?	Description
univ_id	Y	Unique student identifier
group_name	Y	The group name e.g. Hiking Club
membership_type_category	Y	Membership type of group e.g. Silver
next_of_kin_name	N	If next of kin is required, this field is mandatory
next_of_kin_relationship	N	Relationship to student. If next of kin is required, this field is mandatory.
next_of_kin_address	N	Address of the next of kin. If next of kin is required, this field is mandatory.
next_of_kin_mobile_no	N	Either mobile, home or work number is required, if next of kin is required
next_of_kin_home_no	N	Either mobile, home or work number is required, if next of kin is required
next_of_kin_work_no	N	Either mobile, home or work number is

		required, if next of kin is required
create_user_email	Y	Email address of the person adding the group membership, e.g. admin@uni.ac.uk

### 3.3 SOAP Web Service

NUS Digital exposes a SOAP web service to which student and programme data can be pushed from an institutions systems.

WSDL for the SOAP web service could be accessed via  
[https://<union domain>/api/soap/union\\_ws/wsdl](https://<union domain>/api/soap/union_ws/wsdl)

For e.g, if the union URL is fibchester.unioncloud.org, then the WSDL is available at  
[https://fibchester.unioncloud.org/api/soap/union\\_ws/wsdl](https://fibchester.unioncloud.org/api/soap/union_ws/wsdl)

Messages will contain multiple records, but will have a maximum number to mitigate any performance problems. Above that limit multiple messages must be sent.

Records will be of the following types:

New record

Amended record

Temporary delete – For student data, this is just a status change, to indicate the person is no longer a student (but may become a student again), and NUS can retain the data. Not applicable for Programme Data.

Permanent delete – For student data, this is where the record should be anonymised so the person can no longer be identified. e.g. the person has requested to leave or not share their data with the union, or the DPA retention period for holding the data has elapsed. Not applicable for Programme Data.

Delete – For programme data, this is where the programme is marked as inactive. Not applicable for Student Data.

For each record sent there is a response which indicates either:

Success

Failure, include a list of one or more failure reasons (see below)

The institution should have a mechanism to mark that a record that needs synchronising has been sent to NUS Digital and that a successful acknowledgement has been received. If an acknowledgement is not received the message should be re-sent.

Pre-requisites for SOAP upload:

1. Register the IP address(s) (comma separated list) via Union admin screen. These IP are of Institution servers from where the SOAP requests are made.
2. Add the developer key and password in the SOAP header. There is Union admin screen to add the developer details. This is provided by Unioncloud.

### 3.4 REST Service

This web service accepts the student/programme data in JSON format.

The API can be accessed at:

<https://union domain/api/json/upload/students>

<https://union domain/api/json/upload/programmes>

For e.g. for fibchester union, the URL would be

<https://fibchester.unioncloud.org/api/json/upload/students>

<https://fibchester.unioncloud.org/api/json/upload/programmes>

REST web service is designed to upload student and programme data. Unioncloud authenticates the API requests using the **auth\_token** in the request header. Unioncloud accepts the JSON requests, processes the requests and generates the response synchronously. Unioncloud accepts 100 student/programme records per request.

Pre-requisites for REST upload:

1. Register the IP address(s) (comma separated list) via Union admin screen. These IP are of Institution servers from where the REST requests are made.
2. Add the auth\_token in the request header. Generation of auth\_token is specified in the below section. Please refer *Steps to generate the authentication token(auth token)*

---

**JSON example for student uploads:**

```
{
  "data": [
    {
      "id": "U0053",
      "forename": "John",
      "surname": "Peter",
      "dob": "22/08/1917",
      "gender": "male",
      "institution_email": "john.peter@univ.ac.uk",
      "nationality": "United Kingdom",
      "domicile_country": "Nigeria",
      "fee_status": "IN",
      "hall_of_residence": "hog",
      "programme_id": "22",
      "study_type": "UG",
      "programme_level": "2",
      "start_date": "",
      "end_date": "09/06/2017",
      "record_type": "",
      "alternate_email_address": "",
      "library_card": "",
      "department": "",
      "erasmus": "",
      "ethnicity": "",
      "finalist": "",
      "mode_of_study": "",
      "placement": "",
      "address": "",
      "postcode": "",
      "additional_identities": [
        { "provider": "facebook", "id": "2s45sss6sB8" },
        { "provider": "lusu_number", "id": "DssEsssMO123" }
      ]
    },
    {
      "id": "32423",
      "forename": "Jane",
      "surname": "Patinson",
      "dob": "22/08/1912",
      "gender": "female",
      "institution_email": "jane@univ.ac.uk",
      "nationality": "United Kingdom",
      "domicile_country": "Nigeria",
      "fee_status": "IN",
      "hall_of_residence": "hog",
      "programme_id": "22",
      "study_type": "UG",
      "programme_level": "2",
      "start_date": "09/06/2019",
      "end_date": "09/06/2020",
      "record_type": "",
      "alternate_email_address": "",
      "library_card": "",
      "department": "",
      "erasmus": ""
    }
  ]
}
```

```
"ethnicity": "", "finalist":  
"", "mode_of_study": "",  
"placement": "",  
"address": "",  
"postcode": "",  
"additional_identities": [  
  { "provider": "facebodok", "id": "dssasda"},  
  { "provider": "lusu_number", "id": "2ssq13"}  
]  
},  
{  
  "id": "U0044",  
  "forename": "Smith",  
  "surname": "Jones",  
  "dob": "22/08/1999",  
  "gender": "male",  
  "institution_email": "smith.jones@univ.ac.uk",  
  "nationality": "United Kingdom",  
  "domicile_country": "Nigeria",  
  "fee_status": "UK",  
  "hall_of_residence": "hog",  
  "programme_id": "22",  
  "study_type": "UG",  
  "programme_level": "2",  
  "start_date": "09/06/2019",  
  "end_date": "09/06/2020",  
  "record_type": "update",  
  "alternate_email_address": "",  
  "library_card": "",  
  "department": "",  
  "erasmus": "Y",  
  "ethnicity": "",  
  "finalist": "",  
  "mode_of_study": "",  
  "placement": "",  
  "address": "",  
  "postcode": "",  
  "additional_identities": [  
    { "provider": "facebook", "id": "2Vs3ssss4sssss256128"},  
    { "provider": "lusu_number", "id": "AssssSsssssssV1ss22"}  
  ]  
}  
}]}
```

---

**Sample student upload's response:**

```
{
  "meta": {"Summary": {
    "Total": 9,
    "Failure": 5,
    "Success": 4
  }},
  "data": [
    {
      "institution_email": "john.peter@univ.ac.uk",
      "status": "Success",
      "error": null,
      "uid": 1284045,
      "id": "209"
    },
    {
      "institution_email": "jane@univ.ac.uk",
      "status": "Failed",
      "error": [
        {
          "error_code": "ERR102",
          "error_message": "INVALID: user forename can't be blank"
        },
        {
          "error_code": "ERR102",
          "error_message": "INVALID: user forename only alphanumeric characters
and [ '-' ] allowed"
        }
      ]
    },
    {
      "uid": 1284038,
      "id": "201"
    },
    {
      "institution_email": "smith.jones@univ.ac.uk",
      "status": "Failed",
      "error": [
        {
          "error_code": "ERR114",
          "error_message": "INVALID: course_finishing_year cannot be null or blank"
        }
      ],
      "uid": 1284042,
      "id": "205"
    }
  ]
}
```

**JSON example for programme upload:**

```
{"data": [
  {
    "id": "CHEM021",
    "programme_name": "PGCert Env Monitor & Cntrl FT",
    "jacs_code": "AX001",
    "study_type": "PG",
    "faculty_name": "Mr. James",
    "school_name": "Science & Land-based Stds",
    "record_type": "update"
  }
]
```

```
    },  
    {  
      "id": "LAW_020",  
      "programme_name": "PGDip Adv Litig (in-house) DL",  
      "jacs_code": "AX002",  
      "study_type": "PG",  
      "faculty_name": "Ms. Jane",  
      "school_name": "Nottingham Law School",  
      "record_type": "update"  
    }  
  ]  
}
```

**Sample programme upload's response:**

```
{  
  "Summary":  
    [{  
      "Total": 5,  
      "Failure": 1,  
      "Success": 4  
    }],  
  "Failed Records":  
    [{"LAW_PG", {"RECORD_TYPE_REQUIRED": "record_type is mandatory"}}]  
}
```



### Steps to generate the authentication token(auth token)

The sequence of parameters which are used to generate secret hash is:

**email** - This will be union email (UnionCloud will provide this)

**password** - This will be union password (UnionCloud will provide this)

**app\_id** - This will be app name (UnionCloud will provide this)

**date\_stamp** - This will be current date and time in long format

**hash** - This will be app password (hash key) (UnionCloud will provide the app password, using all the parameters, hash is to be generated with SHA-256 encryption)

### **Web service URL:**

<https://<uniondomain>/api/authenticate>

**uniondomain:** Custom URL of the union in UnionCloud

e.g.: <https://fibchester.unioncloud.org/api/authenticate>

UnionCloud expects parameters in the JSON format.

### **Sample Input Request:**

```
{
  "email": "vijay.rawat@arrkggroup.com"
  "password": "admin@123"
  "app_id": "fidelity"
  "date_stamp": "1433203820"
  "hash": "
95fdcb863bcc90fb75dce135aa0317b54f627bc7dc4dd243eda833c62b9d228a"
}
```

***email** - email address of any user within the Unioncloud having 'Allow Access' permission*

***password** - password on Unioncloud of the above user*

***app\_id** - name of the developer created on Unioncloud*

***date\_stamp** - current date time*

***hash** - SHA-256 encrypted hash of email, password, app\_id, date\_stamp and the app secret/app password*

*Join:*

*email << password << app\_id << date\_stamp << app\_password*

*For example, if the data is similar to Sample given above and the app\_secret is 'admin@123', then hash is generated by the following string:*

*vijay.rawat@arrkggroup.comadmin@123fidelity1433203820admin@123*

**Sample Success response:**

```
{ "result": "SUCCESS",  
  "response": {  
    "id": 30,  
    "union_id": 4,  
    "auth_token": "NA84w2keXLvcyN7HG4G5cAdd",  
    "expires": 7200  
  }  
}
```

**Note:** The field 'expires' in the response indicates how long this token will be valid for, it is in seconds.

**Sample Failure response:**

```
{ "result": "FAILURE",  
  "error": {  
    "message": "authentication failed",  
    "code": 402  
  }  
}
```

### 3.5 Data Feed Error List

This lists the errors that can be sent in response to a student or programme data request.

#### 3.5.1 Rate Limits

Rate limits for the APIs work in the following way:

If an API has a rate limit of x requests and the reset interval is y seconds;

Post each request sent, the reset interval of y seconds is the time when the value of remaining requests is reset to x.

Eg. If you send 12 requests with the time between each requests being less than 60 seconds, your next batch of available requests will be available 60 seconds after the last request in the previous batch was sent.

API	max_no_of_requests	reset_interval_in_secs
api/json/upload/students	12	60
api/json/upload/ programmes	20	60

In case, any REST Client, exceeds the rate limit, they would see following JSON in the response body:

```
{ "error_code"=> "403", "error_message" => "API rate limit exceeded"}
```

#### 3.5.2 Student Data Errors

Error Code	Validated field	Error Message	Reason
ERR108	id	"MANDATORY_FIELDS_REQUIRED: id is mandatory"	University id is mandatory
ERR108	id	"INVALID: univ_id ID is already registered with the union"	University id is already registered with the union
ERR108	id	"INVALID: univ_id Invalid University ID"}"	University id is invalid
ERR107	institution_email	"MANDATORY_FIELDS_REQUIRED : institution_email is mandatory"	Institution email id is mandatory
ERR107	institution_email	"INVALID : institution_email is not valid"	Invalid institution email id
ERR107	institution_email	"INVALID_USER.EMAIL : User email Your email address is already registered with the union"	Email id is already registered with the union
ERR102	forename	"INVALID: user forename can't be blank"	Forename is mandatory, can't be blank
ERR102	forename	"INVALID: user forename Special characters ? * ! @ # \$ % ^ & * ( ) < > /   { } [ ] ; , \ : " are not allowed."	Invalid forename, Special characters ? * ! @ # \$ % ^ & * ( ) < > /   { } [ ] ; , \ : " are not allowed.
ERR103	surname	"INVALID: user surname can't be blank"	Surname is mandatory, can't be blank

ERR103	surname	"INVALID: user surname Special characters ? * ! @ # \$ % ^ & * ( ) < > /   { } [ ] ; , \ : " are not allowed."	Invalid Surname, Special characters ? * ! @ # \$ % ^ & * ( ) < > /   { } [ ] ; , \ : " are not allowed.
ERR104	dob	"INVALID: user dob must be after 21-12-1915"	Date of birth should be after 21-12-1915 and before present current date time
ERR105	gender	"INVALID: user gender <invalid input> is not a valid gender"	Anything except 'M' or 'F' or 'N' or 'O' is an invalid gender, can be left blank
ERR109	nationality	"INVALID: nationality is invalid."	Nationality is invalid
ERR110	domicile_country	"INVALID: domicile country is invalid."	Domicile country is invalid
ERR111	fee_status	"INVALID: fee status is invalid. Accepted values are (UK, EU, IN)"	Invalid fee status, UK, EU, IN are the only accepted values
ERR112	study_type	"INVALID: study_type is invalid"	Invalid study type
ERR113	programme_level	"INVALID: programme_level is invalid"	Invalid programme level, accepts only blank and values 0 and above
ERR114	end_date	"INVALID: Course finishing year can't be null or blank "	Course finishing year cannot be null or blank
ERR114	end_date	"INVALID: course finishing year must be after <i>current date</i> "	Course finishing year should be after current date time
ERR121	record_type	"MANDATORY_FIELDS_REQUIRED: record_type is mandatory"	Record type is a mandatory field
ERR115	alternate_email_address	"INVALID: user alternate email addr is invalid"	Invalid alternate email address
ERR115	alternate_email_address	"INVALID: user alternate email addr already exists in the system"	Alternate email address already exists in the union
ERR116	library_card	"INVALID: Invalid LIBRARY CARD"	Invalid library card if Regex format configured
ERR116	library_card	"INVALID: user library card is already registered with the union"	Library card is already registered with the system
ERR117	erasmus	"INVALID: erasmus status is invalid . Accepted values :Y or N"	Invalid ERASMUS status, Y or N are the only accepted values
ERR118	finalist	"INVALID: finalist status is invalid. Accepted values :Y or N"	Invalid finalist status, Y or N are the only accepted values
ERR119	mode_of_study	"INVALID: mode of study is invalid. Accepted values: Full-Time or Part-Time"	Invalid mode of study, Full-Time or Part-Time are the only accepted values
ERR120	placement	"INVALID: placement status is invalid. Accepted values: Y or N or P or R"	Invalid placement status, Y or N or P or R are the accepted values
ERR106	additional_identities{id}	"Additional identity: id = {id} for provider = {provider name} already exists within the union"	Identity provider already exists within the union

### 3.5.3 Programme Data Errors

Condition	Code
id is not present.	ID_REQUIRED
programme_name is not present	INVALID_NAME
JACS code does not have valid pattern	INVALID_JACS_CODE
study_type is any value other than the restricted enumeration.	INVALID_STUDY_TYPE
Record_type is not present	RECORD_TYPE_REQUIRED
Record_type is some value other than mentioned in the document	INVALID_RECORD_TYPE
Any other system failure	SYS_ERR

### 3.5.4 Group Join Data Errors

Condition	Code
univ_id is not present in data or not found in system	Invalid univ_id
group_name is not present in data or not found in the system	Invalid group_name
next_of_kin_name is not present and next of kin is required	Invalid membership_type_category
next_of_kin_relationship is not present and next of kin is required	next_of_kin_relationship can't be blank
next_of_kin_address is not present and next of kin is required	next_of_kin_address can't be blank
mobile or work or home number is not present and next of kin is required	Either home or work or mobile number is required
create_user_email is not present in the data	Invalid create_user_email
Student have joined the group	Already a member of this group
Student have joined the group and is blocked by admin	Membership is Blocked
Student have joined the group and their request is awaiting admin's approval	Your request for membership is being processed.
Student have joined the group and their payment is pending.	You have requested for group membership but payment not processed. Use shopping cart to process payment.
Group has been configured for only above eighteen people and the student is below eighteen	Need to be over 18 years of age to join this group.
Membership_type is configured for specific user groups and student do not belong to that user group	Sorry! You do not have Credentials to join this group

## Appendix A – Supported ISO Country Codes

The data import supports 2 or 3 character country codes. In addition to the standard ISO codes, a number of non-standard extensions are also supported for England, Wales, Scotland and Northern Ireland.

Country	ISO 2	ISO 3
Afghanistan	AF	AFG
Åland Islands	AX	ALA
Albania	AL	ALB
Algeria	DZ	DZA
American Samoa	AS	ASM
Andorra	AD	AND
Angola	AO	AGO
Anguilla	AI	AIA
Antarctica	AQ	ATA
Antigua and Barbuda	AG	ATG
Argentina	AR	ARG
Armenia	AM	ARM
Aruba	AW	ABW
Australia	AU	AUS
Austria	AT	AUT
Azerbaijan	AZ	AZE
Bahamas	BS	BHS
Bahrain	BH	BHR
Bangladesh	BD	BGD
Barbados	BB	BRB
Belarus	BY	BLR
Belgium	BE	BEL
Belize	BZ	BLZ
Benin	BJ	BEN
Bermuda	BM	BMU
Bhutan	BT	BTN
Bolivia Plurinational State of	BO	BOL
Bonaire Sint Eustatius and Saba	BQ	BES
Bosnia and Herzegovina	BA	BIH
Botswana	BW	BWA
Bouvet Island	BV	BVT
Brazil	BR	BRA
British Indian Ocean Territory	IO	IOT
Brunei Darussalam	BN	BRN
Bulgaria	BG	BGR
Burkina Faso	BF	BFA
Burundi	BI	BDI
Cambodia	KH	KHM
Cameroon	CM	CMR

---

Canada	CA	CAN
Cape Verde	CV	CPV
Cayman Islands	KY	CYM
Central African Republic	CF	CAF
Chad	TD	TCD
Chile	CL	CHL
China	CN	CHN
Christmas Island	CX	CXR
Cocos (Keeling) Islands	CC	CCK
Colombia	CO	COL
Comoros	KM	COM
Congo	CG	COG
Congo the Democratic Republic of the	CD	COD
Cook Islands	CK	COK
Costa Rica	CR	CRI
Côte d'Ivoire	CI	CIV
Croatia	HR	HRV
Cuba	CU	CUB
Curaçao	CW	CUW
Cyprus	CY	CYP
Cyprus(not otherwise specified)	XC	XCC
Cyprus(not otherwise specified)	XA	XAA
Djibouti	DJ	DJI
Dominica	DM	DMA
Dominican Republic	DO	DOM
Ecuador	EC	ECU
Egypt	EG	EGY
El Salvador	SV	SLV
England	EN	ENG
Equatorial Guinea	GQ	GNQ
Eritrea	ER	ERI
Estonia	EE	EST
Ethiopia	ET	ETH
Falkland Islands (Malvinas)	FK	FLK
Faroe Islands	FO	FRO
Fiji	FJ	FJI
Finland	FI	FIN
France	FR	FRA
French Guiana	GF	GUF
French Polynesia	PF	PYF
French Southern Territories	TF	ATF
Gabon	GA	GAB
Gambia	GM	GMB
Georgia	GE	GEO
Germany	DE	DEU
Ghana	GH	GHA

---

---

Gibraltar	GI	GIB
Greece	GR	GRC
Greenland	GL	GRL
Grenada	GD	GRD
Guadeloupe	GP	GLP
Guam	GU	GUM
Guatemala	GT	GTM
Guernsey	GG	GGY
Guinea	GN	GIN
Guinea-Bissau	GW	GNB
Guyana	GY	GUY
Haiti	HT	HTI
Heard Island and McDonald Islands	HM	HMD
Holy See (Vatican City State)	VA	VAT
Honduras	HN	HND
Hong Kong	HK	HKG
Hungary	HU	HUN
Iceland	IS	ISL
India	IN	IND
Indonesia	ID	IDN
Iran Islamic Republic of	IR	IRN
Iraq	IQ	IRQ
Ireland	IE	IRL
Isle of Man	IM	IMN
Israel	IL	ISR
Italy	IT	ITA
Jamaica	JM	JAM
Japan	JP	JPN
Jersey	JE	JEY
Jordan	JO	JOR
Kazakhstan	KZ	KAZ
Kenya	KE	KEN
Kiribati	KI	KIR
Korea Democratic People's Republic of	KP	PRK
Korea Republic of	KR	KOR
Kosovan	QO	QOO
Kyrgyzstan	KG	KGZ
Lao People's Democratic Republic	LA	LAO
Latvia	LV	LVA
Lebanon	LB	LBN
Lesotho	LS	LSO
Liberia	LR	LBR
Libya	LY	LBY
Liechtenstein	LI	LIE
Lithuania	LT	LTU
Luxembourg	LU	LUX

---



---

Macao	MO	MAC
Macedonia the former Yugoslav Republic of	MK	MKD
Madagascar	MG	MDG
Malawi	MW	MWI
Malaysia	MY	MYS
Maldives	MV	MDV
Mali	ML	MLI
Malta	MT	MLT
Marshall Islands	MH	MHL
Martinique	MQ	MTQ
Mauritania	MR	MRT
Mauritius	MU	MUS
Mayotte	YT	MYT
Mexico	MX	MEX
Micronesia Federated States of	FM	FSM
Moldova Republic of	MD	MDA
Monaco	MC	MCO
Mongolia	MN	MNG
Montenegro	ME	MNE
Montserrat	MS	MSR
Morocco	MA	MAR
Mozambique	MZ	MOZ
Myanmar	MM	MMR
Namibia	NA	NAM
Nauru	NR	NRU
Nepal	NP	NPL
Netherlands	NL	NLD
New Caledonia	NC	NCL
New Zealand	NZ	NZL
Nicaragua	NI	NIC
Niger	NE	NER
Nigeria	NG	NGA
Niue	NU	NIU
Norfolk Island	NF	NFK
Northern Ireland	ND	NIR
Northern Mariana Islands	MP	MNP
Norway	NO	NOR
Oman	OM	OMN
Pakistan	PK	PAK
Palau	PW	PLW
Palestinian Territory Occupied	PS	PSE
Panama	PA	PAN
Papua New Guinea	PG	PNG
Paraguay	PY	PRY
Peru	PE	PER
Philippines	PH	PHL

---

Pitcairn	PN	PCN
Poland	PL	POL
Portugal	PT	PRT
Puerto Rico	PR	PRI
Qatar	QA	QAT
Réunion	RE	REU
Romania	RO	ROU
Russian Federation	RU	RUS
Rwanda	RW	RWA
Saint Barthélemy	BL	BLM
Saint Helena Ascension and Tristan da Cunha	SH	SHN
Saint Kitts and Nevis	KN	KNA
Saint Lucia	LC	LCA
Saint Martin (French part)	MF	MAF
Saint Pierre and Miquelon	PM	SPM
Saint Vincent and the Grenadines	VC	VCT
Samoa	WS	WSM
San Marino	SM	SMR
Sao Tome and Principe	ST	STP
Saudi Arabia	SA	SAU
Scotland	SW	SCT
Senegal	SN	SEN
Serbia	RS	SRB
Seychelles	SC	SYC
Sierra Leone	SL	SLE
Singapore	SG	SGP
Sint Maarten (Dutch part)	SX	SXM
Slovakia	SK	SVK
Slovenia	SI	SVN
Solomon Islands	SB	SLB
Somalia	SO	SOM
South Africa	ZA	ZAF
South Georgia and the South Sandwich Islands	GS	SGS
South Sudan	SS	SSD
Spain	ES	ESP
Sri Lanka	LK	LKA
Sudan	SD	SDN
Suriname	SR	SUR
Svalbard and Jan Mayen	SJ	SJM
Swaziland	SZ	SWZ
Sweden	SE	SWE
Switzerland	CH	CHE
Syrian Arab Republic	SY	SYR
Taiwan Province of China	TW	TWN
Tajikistan	TJ	TJK
Tanzania United Republic of	TZ	TZA

---

Thailand	TH	THA
Timor-Leste	TL	TLS
Togo	TG	TGO
Tokelau	TK	TKL
Tonga	TO	TON
Trinidad and Tobago	TT	TTO
Tunisia	TN	TUN
Turkey	TR	TUR
Turkmenistan	TM	TKM
Turks and Caicos Islands	TC	TCA
Tuvalu	TV	TUV
Uganda	UG	UGA
Ukraine	UA	UKR
United Arab Emirates	AE	ARE
United Kingdom	GB	GBR
United States	US	USA
United States Minor Outlying Islands	UM	UMI
Uruguay	UY	URY
Uzbekistan	UZ	UZB
Vanuatu	VU	VUT
Venezuela Bolivarian Republic of	VE	VEN
Viet Nam	VN	VNM
Virgin Islands British	VG	VGB
Virgin Islands U.S.	VI	VIR
Wales	WL	WLS
Wallis and Futuna	WF	WLF
Western Sahara	EH	ESH
Yemen	YE	YEM
Zambia	ZM	ZMB
Zimbabwe	ZW	ZWE