



GLOBAL WIND
ORGANISATION

WINDA API Documentation

V3.7

Publication date: 18 September 2023

GW  WINDA



CONTENTS

1.	DOCUMENT CONTROLS.....	3
2.	LIST OF ABBREVIATIONS.....	3
3.	CHANGE LOG.....	3
4.	PREFACE.....	6
4.1	Scope	6
4.2	Contact	6
5.	GENERAL INFORMATION.....	7
5.1	Overview	7
5.2	Target Group	7
5.3	Prerequisites	7
6.	WINDA API – GENERAL INFORMATION.....	8
6.1	Authentication	8
6.2	Identification	9
6.3	Endpoints	10
6.4	Timeout value	10
6.5	API Rate Limiting	10
7.	WINDA API - FUNCTIONS.....	11
7.1	Create Upload	11
7.2	Purchase Upload	13
7.3	Download Upload Error File	15
7.4	Discard Upload	16
7.5	List Uploads	18
7.6	Search	19
	ANNEX 1 - CODE EXAMPLES	24
	ANNEX 2 - HTTP JSON STATUS CODES	35



1. DOCUMENT CONTROLS

Owner	Global Wind Organisation – Head of IT
Editor	Global Wind Organisation – WINDA Product Owner
Classification	Publicly available
Revision	Revised at least yearly before 31 November

2. LIST OF ABBREVIATIONS

API	Application Programming Interface
GWO	Global Wind Organisation
WINDA	Application used by Training Providers to manage and document Course Participants Training Records.

3. CHANGE LOG

Amendment date	Version	Author	Description of changes
18-09-2023	3.7	Torben Buch	Added requirement to implement new usage of API in WINDA test instance before implementing to WINDA production instance. Reorganised listing order of functions. Added flowchart to visualise flow between functions. Updated Annex 2 with additional HTTP codes.
26-07-2023	3.6	Torben Buch	Moved to GWO's Standard document template. Minor structural revision. No new or changed functionality introduced with this version.
10-09-2021	3.5	Phi Tran	Updated chapter 2.5.5 – sample CSV to support the WINDA 3.2 update.
09-06-2021	3.4	Jakup Güven	Updated chapter 2.6 to reflect the transition from Delegates to Course Participants.



23-10-2020	3.3	Yun-Fah Chow	Added Status row in API response field for 2.5.6 Organisational Search. Updated API input and response example
25-09-2020	3.2	Yun-Fah Chow	Added PreviousCourseValidUntil row in API response field for 2.5.6 Organisational Search. Updated API input and response example.
03-08-2020	3.1	Frederik Wieth Pedersen	Added C#, Python and Javascript code samples.
19-12-2019	3.0	Jakup Güven	Revise the URL's from "stag-winda" to "test-winda". Updated images. Added a link to the test site. Added a reference to official "Basic HTTP Authentication".
02-05-2019	2.0	WSP	Revise the URL's from "test-winda" to "stag-winda" Update wording to better represent the current behaviour of the application Update images to reflect changes in url
09-04-2018	1.4	WSP	Updated spelling of "organization" in API call
24-07-2017	1.3	WSP	Added note that user account must be API enabled. Added recommended timeout value of 60 seconds for API requests. Added note on API rate limiting.
17-03-2017	1.2	WSP	Removed references to legacy test site URL in samples.
30-09-2016	1.1	WSP	Changed the URL of the test website; Improved section on obtaining identifiers
15-09-2016	1.0a	WSP	Minor clarification to obtaining identifiers



06-09-2016	1.0	WSP	Initial Version
------------	-----	-----	-----------------



4. PREFACE

4.1 Scope

Global Wind Organisation (GWO) is a non-profit body founded by the wind turbine manufacturers and owners.

Our members strive for an injury free work environment in the wind turbine industry, setting common international standards for safety training and emergency procedures.

The scope of this document is to provide information on how to utilise the WINDA API functions.

4.2 Contact

In case any issues with the WINDA API functions or any other questions related to this document arise, contact GWO by sending an email to: info@globalwindsafety.org



5. GENERAL INFORMATION

5.1 Overview

The WINDA API functions are intended to be used by Training Providers or Organisations. There are 6 WINDA API functions, all of which are described in detail in this document.

Training Providers can use the WINDA API endpoints to programmatically interface with and partially automate their interaction with WINDA.

Organisations can use the WINDA API to search for Training Records for specific Course Participants.

5.2 Target Group

The audience of this document are the IT professionals working on behalf of Training Providers and/or Organisations, who plan to or are using the functions provided by the WINDA API to automate part of their interaction with WINDA.

5.3 Prerequisites

To enable usage of the WINDA API functions, GWO needs to authorize API access for each user account.

Please contact info@globalwindsafety.org to request API enablement.

The request should contain list of the users used for authentication when calling the API functions.

Initial API enablement request can only be made for the WINDA test instance.

An API enablement request for WINDA production instance will not be approved until the solution has been thoroughly tested in WINDA test instance.

GWO may request evidence (such as test plans, test cases and test reports) that a thorough test has been conducted successfully in WINDA test instance prior to approval of an API enablement request to WINDA production instance.

The request for API enablement should also contain an estimate of the number of calls to the WINDA API per day per user. GWO will use this to determine the API rate limit of the user. The API rate limit caps the number of calls a user can invoke per day.



6. WINDA API – GENERAL INFORMATION

WINDA offers six functions through a HTTPS JSON API.

This allows Training Providers to automate part of their interaction WINDA. It also allows Organisations to retrieve the training records for specific Course Participants.

The following functions are available through the WINDA API:

Function	Used by
List uploads	Training Providers
Purchase upload	Training Providers
Discard upload	Training Providers
Create upload	Training Providers
Download upload error file	Training Providers
Search	Training Providers
	Organisations

These are described in detail in chapter 7 WINDA API - Functions.

Code examples in C# (Using RestSharp), Python, JavaScript (Using jQuery) are listed in Annex 1 - Code examples.

6.1 Authentication

All requests to the API functions are authenticated over HTTPS using [Basic Authentication](#). The username and password will be checked against WINDA, using the same authentication as if the user was logging into the WINDA website. Any Training Provider or Organisation user can be used to authenticate to the API using their credentials, provided that the user is API enabled.

For unsuccessful authentication the response will be the following:

```
< HTTP/1.1 401 UNAUTHORIZED
< Content-Type: application/json

{
  "message": "Invalid username/password.",
  "status": 401
}
```

For successful authentication the response will be as follows:



```
< HTTP/1.1 200 OK
< Content-Type: application/json
```

NOTE, cf. 5.3 Prerequisites, the user must be API enabled before it can be used to access the API.

If the user is not API enabled then you will see a response as the following:

```
< HTTP/1.1 403 FORBIDDEN
< Content-Type: application/json

{
  "message": "Account is not API enabled",
  "status": 403
}
```

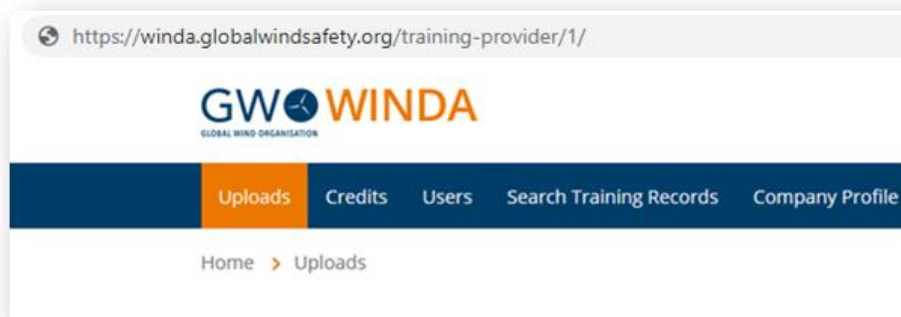
6.2 Identification

In addition to authenticating as a user, the API URLs include a unique identifier for each training provider / organisation.

Training Provider Identifier

To find your Training Provider identifier, login to the website as a training provider user and visit the Uploads list page, then look at the URL in your browser address bar.

For example the identifier for this training provider is 1:



Organisation Identifier

To find your Organisation identifier, login to the website as an organisation user and visit the My Profile page, then look at the URL in your browser address bar.

For example the identifier for this organisation is 16:



Note that these identifiers may differ between the testing and production websites. However they are the same for all users within a training provider.

6.3 Endpoints

There are two separate WINDA websites; test and production. All test transactions and development must be conducted against the test website.

The test website is available at: <https://test-winda.azurewebsites.net/>

The production website is available at: <https://winda.globalwindsafety.org/>

6.4 Timeout value

When calling a function, a timeout value of at least 60 seconds must be set to allow that long-running API calls to return a result. Failing to set the timeout value may result in empty responses or failed requests.

6.5 API Rate Limiting

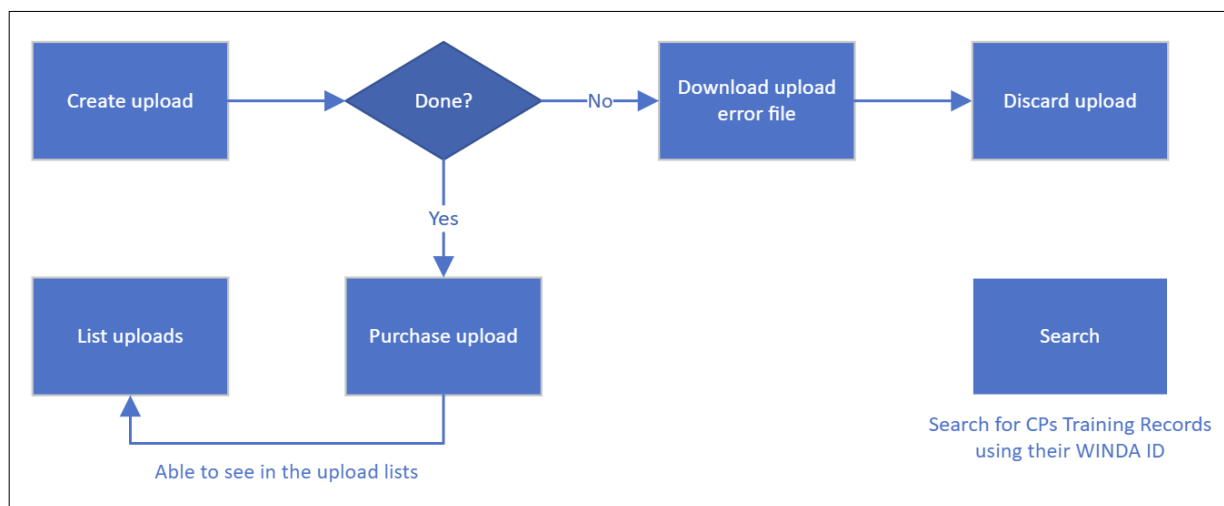
The number of daily calls performed is limited to protect against misuse, rogue internet bots and unintended loop errors from the caller. This is called the API rate limit and is set and controlled by GWO, based on your input on expected number of daily calls. If the daily call limit is exceeded a HTTP status code 429 will be returned.

```
{  
  "message": " API request quota exceeded",  
  "status": 429  
}
```



7. WINDA API - FUNCTIONS

In this chapter the six WINDA API provided functions are described in detail. To understand the flow of all the API calls and how they are connected, a flowchart is presented below:



7.1 Create Upload

This will submit a new bulk upload file to the system, the same as clicking the Upload Record button in the form below:

The screenshot shows the 'ACME Training - New Bulk Upload' form in the WINDA system. The header includes the 'GW WINDA' logo and the user 'John Doe (Training Provider: ABC Training) [admin]'. The navigation bar has links for 'Uploads', 'Credits', 'Users', 'Search Training Records', 'Company Profile', 'Change Password', and 'Logout'. The breadcrumb trail is 'Home > Uploads > New Bulk Upload'. The form title is 'ACME Training - New Bulk Upload'. Below the title, it says 'Upload a new set of training records in CSV format. Refer to the upload instructions and download the template provided.' and 'Maximum of 20000 training records per upload.' The form has two input fields: 'Upload file' with a 'Browse' button, and 'Description' with a text input field. At the bottom, there is a large 'UPLOAD RECORD' button. A note at the very bottom says 'Please wait up to 60 seconds for purchases to complete'.

A sample CSV file is displayed below:



Note: For the courses which were created before 06/09/2021, the third field (facility type) should be empty between the comma:

```
WINDA ID,Course Code,Facility type,Course Completion Date,Previous Course Valid Until Date
1010106,FAW,FIXED Training Facility,2021-05-20,
202020,FAW,MOBILE Training Facility,2021-05-21,
202020,FA,ONSITE Training Facility,2021-06-21,
202020,FAWR,Digital Learning,2021-05-23,2018-06-29
1010106,WAH,,2021-05-20,
202020,FAW,,2021-05-21,
202020,FAWR,,2021-05-23,2018-06-29
```

Description This API will submit a new upload in CSV format.

Endpoint <https://winda.globalwindsafety.org/api/training-provider/{id}/uploads>

Method HTTPS POST

Security Authenticated Training Provider User and Training Provider Admin only

Parameters {id} is the training provider ID

{description} is the description to apply to the upload

{upload_file} is the CSV file to upload

Request The upload file in CSV format

This function can be called from the command line as follows:

Curl (Test):

```
curl --form "description=<description>" --form "upload_file=@<upload_file>" -u
'<username>:<password>' https://test-winda.azurewebsites.net/api/training-provider/<id>/uploads
```

Curl (Production):

```
curl --form "description=<description>" --form "upload_file=@<upload_file>" -u
'<username>:<password>' https://winda.globalwindsafety.org/api/training-provider/<id>/uploads
```

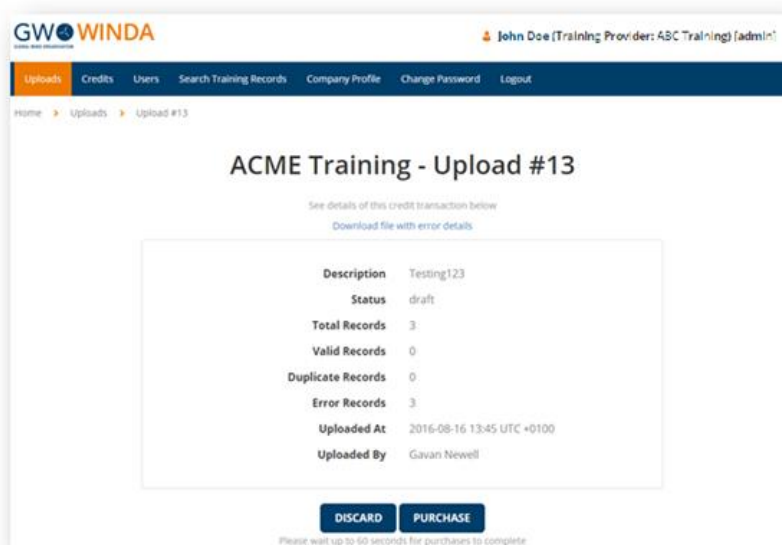
Response	JSON PATH	Description
	status	Value of "200" same as the HTTP response code
	message	Value of "" (empty string)
	records	Array of records containing a single entry
	records[...].id	Upload ID
	records[...].status	Upload status



records[...].description	Upload description
records[...].created_at	Upload created at
records[...].created_by	Upload created by (user first name + last name)
records[...].purchased_at	Upload purchased at
records[...].purchased_by	Upload purchased by (user first name + last name)
records[...].discarded_at	Upload discarded at
records[...].discarded_by	Upload discarded by (user first name + last name)
records[...].total	Upload total records
records[...].valid	Upload valid records
records[...].duplicate	Upload duplicate records
records[...].error	Upload error records

7.2 Purchase Upload

This function will purchase a draft upload using the Training Providers available training credits. This is the same action as clicking the Purchase button on the following screen:





Description This function will purchase a draft upload.

Endpoint <https://winda.globalwindsafety.org/api/training-provider/{id}/uploads/{uid}/purchase>

Method HTTPS POST

Security Authenticated Training Provider User and Training Provider Admin only

Parameters {id} is the training provider ID

{uid} is the upload ID

Request Empty request body / request body is ignored

This function can be called from a Windows command line as follows:

Curl (Test):

```
curl -X POST -u <username>:<password> 'https://test-winda.azurewebsites.net/api/training-provider//<id>/uploads//<uid>/purchase'
```

Curl (Production):

```
curl -X POST -u <username>:<password> 'https://winda.globalwindsafety.org/api/training-provider//<id>/uploads//<uid>/purchase'
```

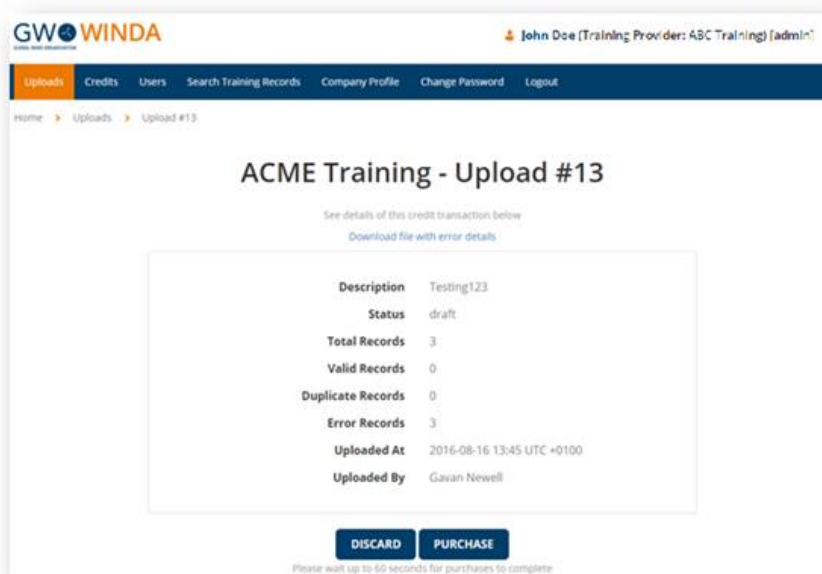
Response	JSON PATH	Description
	status	Value of "200" same as the HTTP response code
	message	Value of "" (empty string)
	records	Array of records containing a single entry
	records[...].id	Upload ID
	records[...].status	Upload status
	records[...].description	Upload description
	records[...].created_at	Upload created at
	records[...].created_by	Upload created by (user first name + last name)
	records[...].purchased_at	Upload purchased at
	records[...].purchased_by	Upload purchased by (user first name + last name)



<code>records[...].discarded_at</code>	Upload discarded at
<code>records[...].discarded_by</code>	Uploaded discarded by (user first name + last name)
<code>records[...].total</code>	Upload total records
<code>records[...].valid</code>	Upload valid records
<code>records[...].duplicate</code>	Upload duplicate records
<code>records[...].error</code>	Upload error records

7.3 Download Upload Error File

This will download the upload error file associated with an upload. This is the same as clicking the link *Download file* with error details on the following screen:



Description This API will download the upload error file in CSV format.

Endpoint <https://winda.globalwindsafety.org/api/training-provider/{id}/uploads/{uid}/download>

Method HTTPS GET

Security Authenticated Training Provider User and Training Provider Admin only

Parameters `{id}` is the training provider ID



{uid} is the upload ID

This API can be called from the command line as follows:

Curl (test):

```
curl -u '<username>:<password>' https://test-winda.azurewebsites.net/api/training-provider/<id>/uploads/<uid>/download
```

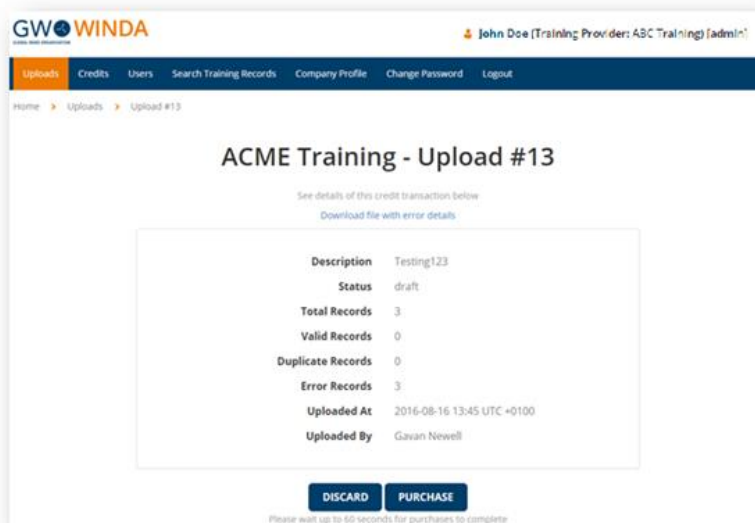
Curl (Production):

```
curl -u '<username>:<password>' https://winda.globalwindsafety.org/api/training-provider/<id>/uploads/<uid>/download
```

Response The upload error details file in CSV format

7.4 Discard Upload

This function will discard a draft upload. This is the same action as clicking the Discard button on the following screen:



Description This function will discard a draft upload.

Endpoint https://winda.globalwindsafety.org/api/training-provider/{id}/uploads/{uid}/discard

Method HTTPS POST

Security Authenticated Training Provider User and Training Provider Admin only

Parameters {id} is the training provider ID



{uid} is the upload ID

Request Empty request body / request body is ignored

This function can be called from the command line as follows:

Curl (Test):

```
curl -X POST -u '<username>:<password>' https://test-winda.azurewebsites.net/api/training-provider/<id>/uploads/<uid>/discard
```

Curl (Production):

```
curl -X POST -u '<username>:<password>' https://winda.globalwindsafety.org/api/training-provider/<id>/uploads/<uid>/discard
```

Response	JSON PATH	Description
	status	Value of "200" same as the HTTP response code
	message	Value of "" (empty string)
	records	Array of records containing a single entry
	records[...].id	Upload ID
	records[...].status	Upload status
	records[...].description	Upload description
	records[...].created_at	Upload created at
	records[...].created_by	Upload created by (user first name + last name)
	records[...].purchased_at	Upload purchased at
	records[...].purchased_by	Upload purchased by (user first name + last name)
	records[...].discarded_at	Upload discarded at
	records[...].discarded_by	Uploaded discarded by (user first name + last name)
	records[...].total	Upload total records
	records[...].valid	Upload valid records
	records[...].duplicate	Upload duplicate records
	records[...].error	Upload error records



7.5 List Uploads

This function will list all the Training Providers uploads.

The function provides the same information as the following screen when using the website interactively. The below is a representation of the data which will be displayed:

Id	Status	Description	Uploaded At	Uploaded By	Purchased At	Purchased By	Total	Valid	Duplicate	Errors
42	Purchased	Gavan BMT	2016-08-24	Gavan Newell	2016-08-24	Gavan Newell	1	1	0	0
29	Draft	Testing Soren	2016-08-18	Gavan Newell			2	1	0	1
15	Invalidation	Revoke	2016-08-16	Gavan Newell			1	1	0	0
14	Purchased	test	2016-08-16	Gavan Newell	2016-08-16	Gavan Newell	1	1	0	0
13	Draft	Testing123	2016-08-16	Gavan Newell			3	0	0	3

Description This function provides a list of uploads for the training provider.

Endpoint <https://winda.globalwindsafety.org/api/training-provider/{id}/uploads>

Method HTTPS GET

Security Authenticated Training Provider User and Training Provider Admin only

Parameters {id} is the training provider ID

This function can be called from the command line as follows:

Curl (Test):

```
curl -u '<username>:<password>' https://test-winda.azurewebsites.net/api/training-provider/<id>/uploads
```

Curl (Production):

```
curl -u '<username>:<password>' https://winda.globalwindsafety.org/api/training-provider/<id>/uploads
```

Response	JSON PATH	Description
	status	Value of "200" same as the HTTP response code
	message	Value of "" (empty string)



`records` Array of records

`records[...].id` Upload ID

`records[...].status` Upload status

`records[...].description` Upload description

`records[...].created_at` Upload created at

`records[...].created_by` Upload created by (user first name + last name)

`records[...].purchased_at` Upload purchased at

`records[...].purchased_by` Upload purchased by (user first name + last name)

`records[...].discarded_at` Upload discarded at

`records[...].discarded_by` Uploaded discarded by (user first name + last name)

`records[...].total` Upload total records

`records[...].valid` Upload valid records

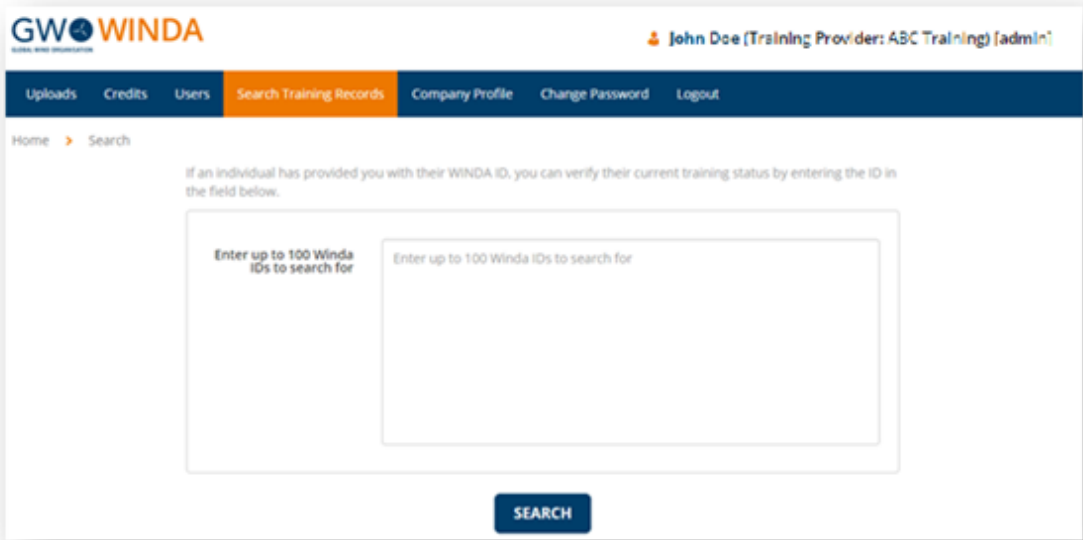
`records[...].duplicate` Upload duplicate records

`records[...].error` Upload error records

7.6 Search

Search is available to both Training Providers and Organisations. The search provides a list of It is the same as performing a bulk training record search as follows:

NOTE: As of May 2021, Delegates are referred to as Course Participants by GWO in all contexts excluding the API. For backward compatibility purposes, the API functions will still return properties referring to delegates. This means that consumers of the API functions need to make no changes to their software.



Description This API provides search results when searching for course providers. Up to 100 WINDA IDs can be included in the search criteria.

Endpoint Available at two endpoints, one for training providers and one for organisations:

`https://winda.globalwindsafety.org/api/training-provider/{id}/search`

`https://winda.globalwindsafety.org/api/organization/{oid}/search`

Method HTTPS POST

Security Authenticated Training Provider User and Training Provider Admin (first endpoint)

Authentication Organisation User (second endpoint)

Parameters {id} is the training provider ID (first endpoint)

{oid} is the organisation user ID (second endpoint)

Request	JSON PATH	Description
	<code>delegates[...]</code>	Array of WINDA ID records to search for

The list of records to search for must be provided in JSON input format. For the example this would search for WINDA IDs 'JH038986DK and TD070198DK.

```
{
  "delegates": [
    "JH038986DK", "TD070198DK"
  ]
}
```



```
]
}
```

In this example we have saved this search criteria to local file called 'search.json':

```
$ cat search.json
delegates": ["JH038986DK", "TD070198DK"]}
```

This API can be called from the command line as follows:

Curl (Test) Training Provider:

```
curl -X POST -d @search.json --header "Content-Type:application/json" -u '<username>:<password>'
https://test-winda.azurewebsites.net/api/training-provider/<id>/search
```

Curl (Test) Organisation:

```
curl -X POST -d @search.json --header "Content-Type:application/json" -u '<username>:<password>'
https://test-winda.azurewebsites.net/api/organization/<oid>/search
```

Curl (Production) Training Provider:

```
curl -X POST -d @search.json --header "Content-Type:application/json" -u '<username>:<password>'
https://winda.globalwindsafety.org/api/training-provider/<id>/search
```

Curl (Production) Organisation:

```
curl -X POST -d @search.json --header "Content-Type:application/json" -u '<username>:<password>'
https://winda.globalwindsafety.org/api/organization/<oid>/search
```

Response

JSON PATH	Description
status	Value of "200" same as the HTTP response code
message	Value of "" (empty string)
delegates	Array of matching Course Participant records, one per distinct WINDA ID that was searched for
delegates[...].delegate_id	WINDA ID that was searched for
delegates[...].first_name	First name of the Course Participant (if found, null if not found)
delegates[...].last_name	Last name of the Course Participant (if found, null if not found)
delegates[...].found	Boolean flag 'true' if found, 'false' if not found
delegates[...].records	Array of training record results for this course participant, empty if not found



`delegates[...].records[...].course_title` Title of the course

`delegates[...].records[...].course_code` Code of the course

`delegates[...].records[...].training_provider` Training provider name

`delegates[...].records[...].country` Country name

`delegates[...].records[...].completion_date` Course completion date

`delegates[...].records[...].valid_from` Record valid from date

`delegates[...].records[...].valid_until` Record valid until date

`delegates[...].records[...].previous_course_valid_until` Previous course valid until date (if found, empty string if equals to null)

`delegates[...].records[...].status` Status of course record



Annexes



ANNEX 1 - CODE EXAMPLES

This section contains the code examples of each API function.

1. List Uploads

Following is presented the code example for **List Uploads** endpoint in programming languages: **C# (Using RestSharp), Python, JavaScript (Using jQuery)**.

C# examples uses the RestSharp client Nuget Package: <https://www.nuget.org/packages/RestSharp/>

C# (Using RestSharp):

```
var client = new RestClient("https://winda.globalwindsafety.org/api/training-provider/" + id +
"/uploads");
client.Authenticator = new HttpBasicAuthenticator("your_username", "your_password");
client.Timeout = -1;
var request = new RestRequest(Method.GET);
IRestResponse response = client.Execute(request);
Console.WriteLine(response.Content);
```

Python:

```
import http.client
import mimetypes

conn = http.client.HTTPSConnection("https://winda.globalwindsafety.org")
payload = ''
usernameAndPassword = b64encode(b"username:password").decode("ascii")
headers = { 'Authorization' : 'Basic %s' % usernameAndPassword }
conn.request("GET", "/api/training-provider/" + uid + "/uploads", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

JavaScript (jQuery):

```
var settings = {
  "url": "https://winda.globalwindsafety.org/api/training-provider/" + id + "/uploads",
  "method": "GET",
  "timeout": 0,
  "headers": {
    "Authorization": "Authorization: Basic " + btoa(your_username + ":" + your_password)
  },
};

$.ajax(settings).done(function (response) {
  console.log(response);
});
```

Response:

A successful response will be similar to the following:

```
{
  "status": 200,
  "message": "",
  "count": 5,
  "next": null,
  "previous": null,
```




```
"records": [
  {
    "id": 13,
    "status": "draft",
    "description": "Testing",
    "created_at": "2016-08-16T12:45:57.948899Z",
    "created_by": "John Doe",
    "purchased_at": null,
    "purchased_by": "",
    "discarded_at": null,
    "discarded_by": "",
    "total": 3,
    "valid": 0,
    "duplicate": 0,
    "error": 3
  },
  {
    "id": 14,
    "status": "purchased",
    "description": "test",
    "created_at": "2016-08-16T13:51:27.306585Z",
    "created_by": "John Doe",
    "purchased_at": "2016-08-16T13:51:30.841925Z",
    "purchased_by": "John Doe",
    "discarded_at": null,
    "discarded_by": "",
    "total": 1,
    "valid": 1,
    "duplicate": 0,
    "error": 0
  },
  {
    "id": 15,
    "status": "invalidation",
    "description": "Revoke",
    "created_at": "2016-08-16T13:58:52.038099Z",
    "created_by": "John Doe",
    "purchased_at": null,
    "purchased_by": "",
    "discarded_at": null,
    "discarded_by": "",
    "total": 1,
    "valid": 1,
    "duplicate": 0,
    "error": 0
  },
  {
    "id": 29,
    "status": "draft",
    "description": "Testing Soren",
    "created_at": "2016-08-18T13:56:08.572087Z",
    "created_by": "John Doe",
    "purchased_at": null,
    "purchased_by": "",
    "discarded_at": null,
    "discarded_by": "",
    "total": 2,
    "valid": 1,
    "duplicate": 0,
    "error": 1
  },
  {
    "id": 42,
    "status": "purchased",
    "description": "John BMT",
    "created_at": "2016-08-24T09:00:24.838555Z",
    "created_by": "John Doe",
    "purchased_at": "2016-08-24T09:00:28.843537Z",
    "purchased_by": "John Doe",
    "discarded_at": null,
    "discarded_by": ""
  }
]
```



```
        "total": 1,  
        "valid": 1,  
        "duplicate": 0,  
        "error": 0  
    }  
]  
}
```

2. Purchase Upload

Following is presented the code examples for **Purchase Upload** endpoint in programming languages: **C# (Using RestSharp), Python, JavaScript (Using jQuery)**

C# examples uses the RestSharp client Nuget Package: <https://www.nuget.org/packages/RestSharp/>

C# (Using RestSharp):

```
var client = new RestClient("https://winda.globalwindsafety.org/api/training-provider/" + id +  
    "/uploads/" + uid + "/purchase");  
client.Authenticator = new HttpBasicAuthenticator("your_username", "your_password");  
client.Timeout = -1;  
var request = new RestRequest(Method.POST);  
IRestResponse response = client.Execute(request);  
Console.WriteLine(response.Content);
```

Python:

```
import http.client  
import mimetypes  
  
conn = http.client.HTTPSConnection("https://winda.globalwindsafety.org")  
payload = ''  
usernameAndPassword = b64encode(b"username:password").decode("ascii")  
headers = { 'Authorization' : 'Basic %s' % usernameAndPassword }  
conn.request("POST", "/api/training-provider/" + id + "/uploads/" + uid + "/purchase", payload,  
headers)  
res = conn.getresponse()  
data = res.read()  
print(data.decode("utf-8"))
```

JavaScript (jQuery):

```
var settings = {  
    "url": "https://winda.globalwindsafety.org/api/training-provider/" + id + "/uploads/" + uid +  
    "/purchase",  
    "method": "POST",  
    "timeout": 0,  
    "headers": {
```



```
        "Authorization": "Authorization": "Basic " + btoa(your_username + ":" + your_password)
    },
};
$.ajax(settings).done(function (response) {
    console.log(response);
});
```

Response:

A successful response will be similar to the following:

```
{
  "message": "",
  "status": 200,
  "records": [
    {
      "id": 13,
      "status": "purchased",
      "description": "Testing",
      "created_at": "2016-08-16T12:45:57.948899Z",
      "created_by": "John Doe",
      "purchased_at": "2016-09-06T09:34:04.271929Z",
      "purchased_by": "John Doe",
      "discarded_at": null,
      "discarded_by": "",
      "total": 3,
      "valid": 0,
      "duplicate": 0,
      "error": 3
    }
  ]
}
```

An unsuccessful response due to trying to purchase a non-draft upload will be similar to the following:

```
{
  "non_field_errors": [
    "Purchase only permitted for draft uploads"
  ],
  "message": "Purchase only permitted for draft uploads",
  "status": 400
}
```

3. Discard Upload

Following is presented the code examples for **Discard Upload** endpoint in programming languages: **C# (Using RestSharp), Python, JavaScript (Using jQuery)**

C# examples uses the RestSharp client Nuget Package: <https://www.nuget.org/packages/RestSharp/>

C# (Using RestSharp):

```
var client = new RestClient("https://winda.globalwindsafety.org/api/training-provider/" + id +
"/uploads/" + uid "/discard");
client.Authenticator = new HttpBasicAuthenticator("your_username", "your_password"); client.Timeout =
-1;
var request = new RestRequest(Method.POST);
```



```
IRestResponse response = client.Execute(request);  
Console.WriteLine(response.Content);
```

Python:

```
import http.client  
import mimetypes  
  
conn = http.client.HTTPSConnection("https://winda.globalwindsafety.org")  
payload = ''  
usernameAndPassword = b64encode(b"username:password").decode("ascii")  
headers = { 'Authorization' : 'Basic %s' % usernameAndPassword }  
conn.request("GET", "/api/training-provider/" + id + "/uploads/" + uid + "/discard", payload,  
headers)  
res = conn.getresponse()  
data = res.read()  
print(data.decode("utf-8"))
```

JavaScript (jQuery):

```
import http.client  
import mimetypes  
  
conn = http.client.HTTPSConnection("https://winda.globalwindsafety.org")  
payload = ''  
usernameAndPassword = b64encode(b"username:password").decode("ascii")  
headers = { 'Authorization' : 'Basic %s' % usernameAndPassword }  
conn.request("GET", "/api/training-provider/" + id + "/uploads/" + uid + "/discard", payload,  
headers)  
res = conn.getresponse()  
data = res.read()  
print(data.decode("utf-8"))
```

Response:

A successful response will be similar to the following:

```
{  
  "message": "",  
  "status": 200,  
  "records": [  
    {  
      "id": 29,  
      "status": "discarded",  
      "description": "Testing",  
      "created_at": "2016-08-18T13:56:08.572087Z",  
      "created_by": "John Doe",  
      "purchased_at": null,  
      "purchased_by": "",  
      "discarded_at": "2016-09-06T09:47:38.343070Z",  
      "discarded_by": "John Doe",  
      "total": 2,  
      "valid": 1,  
      "duplicate": 0,  
    }  
  ]  
}
```



```
        "error": 1
    }
  ]
}
```

An unsuccessful response due to trying to purchase a non-draft upload will be similar to the following:

```
{
  "non_field_errors": [
    "Discard only permitted for draft uploads"
  ],
  "status": 400,
  "message": "Discard only permitted for draft uploads"
}
```

4. Create Upload

Following is presented the code examples for **Create Upload** endpoint in programming languages: **C# (Using RestSharp), Python, JavaScript (Using jQuery)**

C# examples uses the RestSharp client Nuget Package: <https://www.nuget.org/packages/RestSharp/>

C# (Using RestSharp):

```
var client = new RestClient("https://winda.globalwindsafety.org/api/training-provider/" + id +
"/uploads");
client.Authenticator = new HttpBasicAuthenticator("your_username", "your_password"); client.Timeout =
-1;
var request = new RestRequest(Method.POST);
request.AddParameter("description", "description_text");
request.AddFile("upload_file", FileToBeUploaded.csv);
IRestResponse response = client.Execute(request);
Console.WriteLine(response.Content);
```

Python:

```
import http.client
import mimetypes

conn = http.client.HTTPSConnection("localhost:44358")
dataList = []
boundary = 'wL36Yn8afVp8Ag7AmP8qZ0SA4n1v9T'
dataList.append('--' + boundary)
dataList.append('Content-Disposition: form-data; name=description;')
dataList.append('Content-Type: {}'.format('multipart/form-data'))
dataList.append('')
dataList.append("description_text")
dataList.append('--' + boundary)
dataList.append('Content-Disposition: form-data; name=upload_file;')
filename={0}'.format('UploadTestFile.csv'))
fileType = mimetypes.guess_type('UploadTestFile.csv')[0] or 'application/octet-stream'
dataList.append('Content-Type: {}'.format(fileType))
```



```
dataList.append('')
with open('UploadTestFile.csv') as f:
    dataList.append(f.read())
dataList.append('--'+boundary+'--')
dataList.append('')
body = '\r\n'.join(dataList)
payload = body
usernameAndPassword = b64encode(b"username:password").decode("ascii")
headers = {
    'Authorization': 'Authorization': 'Basic %s' % usernameAndPassword',
    'Content-type': 'multipart/form-data; boundary={}'.format(boundary)
}
conn.request("POST", "/api/training-provider/" + id + "/uploads", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

JavaScript (jQuery):

```
var form = new FormData();
form.append("description", "\"description text\"");
form.append("upload_file", fileInput.files[0], "UploadTestFile.csv");
var settings = {
    "url": "https://winda.globalwindsafety.org/api/training-provider/" + id + "/uploads",
    "method": "POST",
    "timeout": 0,
    "headers": {
        "Authorization": "Basic " + btoa(your_username + ":" + your_password)
    },
    "processData": false,
    "mimeType": "multipart/form-data",
    "contentType": false,
    "data": form
};

$.ajax(settings).done(function (response) {
    console.log(response);
});
```

Response:

A successful response will be similar to the following:

```
{
  "message": "",
  "records": [
    {
      "id": 49,
      "status": "draft",
      "description": "Testing123",
      "created_at": "2016-09-06T09:56:48.912454Z",
    }
  ]
}
```



```
        "created_by": "John Doe",
        "purchased_at": null,
        "purchased_by": "",
        "discarded_at": null,
        "discarded_by": "",
        "total": 3,
        "valid": 0,
        "duplicate": 0,
        "error": 3
    }
],
"status": 200
}
```

5. Download Upload Error File

Following is presented the code example for **Download Upload Error File** endpoint in programming languages: *C# (Using RestSharp), Python, JavaScript (Using jQuery)*

C# examples uses the RestSharp client Nuget Package: <https://www.nuget.org/packages/RestSharp/>

C# (Using RestSharp):

```
var client = new RestClient("https://winda.globalwindsafety.org/api/training-provider/" + id +
"/uploads/" + uid + "/download");
client.Authenticator = new HttpBasicAuthenticator("your_username", "your_password"); client.Timeout =
-1;
var request = new RestRequest(Method.GET);
IRestResponse response = client.Execute(request);
Console.WriteLine(response.Content);
```

Python:

```
import http.client
import mimetypes

conn = http.client.HTTPSConnection("https://winda.globalwindsafety.org")
payload = ''
usernameAndPassword = b64encode(b"username:password").decode("ascii")
headers = { 'Authorization' : 'Basic %s' % usernameAndPassword }
conn.request("GET", "/api/training-provider/" + id + "/uploads/" + uid + "/download", payload,
headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

JavaScript (jQuery):

```
var settings = {
    "url": "https://winda.globalwindsafety.org/api/training-provider/" + id + "/uploads/" + uid +
"/download",
    "method": "GET",
    "timeout": 0,
```



```
"headers": {  
  "Authorization": "Authorization": "Basic " + btoa(your_username + ":" + your_password)  
},  
};  
$.ajax(settings).done(function (response) {  
  console.log(response);  
});
```

Response:

A successful response will be similar to the following:

```
WINDA ID,Course Code,Course Completion Date,Previous Course Valid Until Date,Calculated Valid From  
Date,Calculated Valid Until Date,Errors  
1010106,FAW,2016-05-20,,2016-05-20,2018-05-20,Not accredited for course; Unknown Winda ID  
202020,FAW,2016-05-21,,2016-05-21,2018-05-21,Not accredited for course; Unknown Winda ID  
202020,FAW,2016-05-23,,2016-05-23,2018-05-23,Not accredited for course; Unknown Winda ID
```

6. Search

Following is presented the code examples for **Search** endpoint in programming languages: **C# (Using RestSharp)**, **Python**, **JavaScript (Using jQuery)**

C# examples uses the RestSharp client Nuget Package: <https://www.nuget.org/packages/RestSharp/>

C# (Using RestSharp):

```
var client = new RestClient("https://winda.globalwindsafety.org/api/training-provider/" + id +  
"/search");  
client.Timeout = -1;  
var request = new RestRequest(Method.POST);  
request.AddHeader("Content-Type", "application/json");  
request.AddParameter("application/json", "{ \"delegates\": [\"delegate_id\"] }",  
ParameterType.RequestBody);  
IRestResponse response = client.Execute(request);  
Console.WriteLine(response.Content);
```

Python:

```
import http.client  
import mimetypes  
conn = http.client.HTTPSConnection("https://winda.globalwindsafety.org")  
payload = "{ \"delegates\": [\"delegate_id\"] }"  
usernameAndPassword = b64encode(b"username:password").decode("ascii")  
headers = {  
  'Authorization': 'Basic %s' % usernameAndPassword,  
  'Content-Type': 'application/json'  
}  
conn.request("POST", "/api/training-provider/" + id + "/search", payload, headers)  
res = conn.getresponse()  
data = res.read()  
print(data.decode("utf-8"))
```

JavaScript (jQuery):

```
var settings = {  
  "url": "https://winda.globalwindsafety.org/api/training-provider/" + id + "/search",  
  "method": "POST",  
  "timeout": 0,  
  "headers": {
```




```
    "Authorization": "Authorization: Basic " + btoa(your_username + ":" + your_password),
    "Content-Type": "application/json"
  },
  "data": JSON.stringify({"delegates":["delegateId"]}),
};

$.ajax(settings).done(function (response) {
  console.log(response);
});
```

Response:

A successful response will be similar to the following:

```
{
  {
    "status": 200,
    "message": "",
    "count": 2,
    "next": null,
    "previous": null,
    "records": [
      {
        "found": true,
        "delegate_id": "JH038986DK",
        "first_name": "Joachim",
        "last_name": "Hansen",
        "records": [
          {
            "course_title": "First Aid",
            "course_code": "FA",
            "country": "Germany",
            "training_provider": "Skola Morska w Gdyni Sp. z.o.o",
            "completed_on": "2018-07-02",
            "valid_from": "2018-07-02",
            "valid_until": "2020-07-02",
            "previous_course_valid_until": "",
            "status": "Record is invalid"
          },
          {
            "course_title": "First Aid",
            "course_code": "FA",
            "country": "Denmark",
            "training_provider": "windhunter academy sp. z o. o.",
            "completed_on": "2020-01-14",
            "valid_from": "2020-01-14",
            "valid_until": "2022-01-14",
            "previous_course_valid_until": "",
            "status": "Record is valid"
          },
          {
            "course_title": "Fire Awareness",
            "course_code": "FAW",
            "country": "Central African Republic",
            "training_provider": "Skola Morska w Gdyni Sp. z.o.o",
            "completed_on": "2019-12-02",
            "valid_from": "2019-12-02",
            "valid_until": "2021-12-03",
            "previous_course_valid_until": "",
            "status": "Record is valid"
          },
          {
            "course_title": "Fire Awareness Refresher",
            "course_code": "FAWR",
            "country": "Central African Republic",
            "training_provider": "Skola Morska w Gdyni Sp. z.o.o",
            "completed_on": "2020-03-23",
            "valid_from": "2018-03-21",
            "valid_until": "2020-03-21",

```



```
        "previous_course_valid_until": "2020-04-24",
        "status": "Record is invalid"
    }
]
},
{
    "found": true,
    "delegate_id": "TD070198DK",
    "first_name": "Jakup",
    "last_name": "Güven",
    "records": [
        {
            "course_title": "Fire Awareness",
            "course_code": "FAW",
            "country": "Central African Republic",
            "training_provider": "Skola Morska w Gdyni Sp. z.o.o",
            "completed_on": "2020-02-01",
            "valid_from": "2020-02-01",
            "valid_until": "2022-02-02",
            "previous_course_valid_until": "",
            "status": "Record is valid"
        },
        {
            "course_title": "Fire Awareness",
            "course_code": "FAW",
            "country": "Central African Republic",
            "training_provider": "Skola Morska w Gdyni Sp. z.o.o",
            "completed_on": "2020-02-17",
            "valid_from": "2020-02-17",
            "valid_until": "2022-02-17",
            "previous_course_valid_until": "",
            "status": "Record is valid"
        },
        {
            "course_title": "Sea Survival",
            "course_code": "SS",
            "country": "Central African Republic",
            "training_provider": "Skola Morska w Gdyni Sp. z.o.o",
            "completed_on": "2018-12-02",
            "valid_from": "2018-12-02",
            "valid_until": "2020-12-02",
            "previous_course_valid_until": "",
            "status": "Record will soon be invalid"
        },
        {
            "course_title": "Sea Survival Refresher",
            "course_code": "SSR",
            "country": "Central African Republic",
            "training_provider": "Skola Morska w Gdyni Sp. z.o.o",
            "completed_on": "2020-11-10",
            "valid_from": "2020-11-11",
            "valid_until": "2022-12-04",
            "previous_course_valid_until": "2020-12-02",
            "status": "Record is valid"
        }
    ]
}
]
```



ANNEX 2 - HTTP JSON STATUS CODES

1. Create upload - training-provider/{id}/uploads

HTTP Status Codes	Message	Description
400	Your query does not conform to the Basic HTTP Authentication Scheme. See https://tools.ietf.org/html/rfc7617 for more information.	
401	Invalid username/password.	Ensure that the correct user and password is used to authenticate
403	Account is not API enabled.	The user used has not been setup to allow API usage.
403	You do not have permission to perform this action.	<p>The user used has not been setup to allow API usage.</p> <p>Verify that the correct user is used. If correct user, contact GWO to request to allow user to use the API.</p>
403	Payment pending - You do not have permission to perform this action.	If the user hasn't paid for the credits yet.
429	API request quota exceeded.	<p>The user has exceeded the allowed limit of calls to the API.</p> <p>Investigate if the usage pattern has changed and if the change is valid. If the change is valid and needed then request GWO to increase the API rate limit of the user.</p>
500	Error trying to access data that is currently being used by someone else please try again later.	Try to refresh the page and make sure the purchase upload is not open in more than one page.
500	Something went wrong, please contact an administrator if the problem persists.	
200		OK – Success

2. Purchase upload - training-provider/{id}/uploads/{uid}/purchase



HTTP Status Codes	Message	Description
400	Your query does not conform to the Basic HTTP Authentication Scheme. See https://tools.ietf.org/html/rfc7617 for more information.	
400	Purchase only permitted for draft uploads	Not possible to purchase before creating the record.
400	You don't have the right to purchase this upload.	The user used has not been setup to allow API usage.
400	You have insufficient credit to purchase this upload. Add additional credit to your account then try again.	Need to buy more credits.
400	This upload doesn't have any valid records.	There is an error in the record validity dates.
401	Invalid username/password.	Ensure that the correct user and password is used to authenticate
200		OK – Success
403	You do not have permission to perform this action.	<p>The user used has not been setup to allow API usage.</p> <p>Verify that the correct user is used. If correct user, contact GWO to request to allow user to use the API.</p>
403	Payment pending - You do not have permission to perform this action.	If the user hasn't paid for the credits yet.
404	Not found.	
429	API request quota exceeded	<p>The user has exceeded the allowed limit of calls to the API.</p> <p>Investigate if the usage pattern has changed and if the change is valid. If the change is valid and needed then request GWO to increase the API rate limit of the user.</p>



500	Error trying to access data that is currently being used by someone else please try again later.	Try to refresh the page and make sure the purchase upload is not open in more than one page.
500	Something went wrong, please contact an administrator if the problem persists.	

3. Download upload error file - training-provider/{id}/uploads/{uid}/download

HTTP Status Codes	Message	Description
400	Your query does not conform to the Basic HTTP Authentication Scheme. See https://tools.ietf.org/html/rfc7617 for more information.	
401	Invalid username/password.	Ensure that the correct user and password is used to authenticate
403	Account is not API enabled.	The user used has not been setup to allow API usage.
403	You do not have permission to perform this action.	<p>The user used has not been setup to allow API usage.</p> <p>Verify that the correct user is used. If correct user, contact GWO to request to allow user to use the API.</p>
429	API request quota exceeded.	<p>The user has exceeded the allowed limit of calls to the API.</p> <p>Investigate if the usage pattern has changed and if the change is valid. If the change is valid and needed, then request GWO to increase the API rate limit of the user.</p>
500	Error trying to access data that is currently being used by someone else please try again later.	Try to refresh the page and make sure the purchase upload is not open in more than one page.
500	Something went wrong, please contact an administrator if the problem persists.	
200		OK – Success



4. Discard upload - training-provider/{id}/uploads/{uid}/discard

HTTP Status Codes	Message	Description
400	Your query does not conform to the Basic HTTP Authentication Scheme. See https://tools.ietf.org/html/rfc7617 for more information.	
400	Discard only permitted for draft uploads.	Not possible to discard upload before drafting the record.
401	Invalid username/password.	Ensure that the correct user and password is used to authenticate
403	Account is not API enabled.	The user used has not been setup to allow API usage.
403	You do not have permission to perform this action.	<p>The user used has not been setup to allow API usage.</p> <p>Verify that the correct user is used. If correct user, contact GWO to request to allow user to use the API.</p>
404	Not found.	
429	API request quota exceeded.	<p>The user has exceeded the allowed limit of calls to the API.</p> <p>Investigate if the usage pattern has changed and if the change is valid. If the change is valid and needed then request GWO to increase the API rate limit of the user.</p>
500	Error trying to access data that is currently being used by someone else please try again later.	Try to refresh the page and make sure the purchase upload is not open in more than one page.
500	Something went wrong, please contact an administrator if the problem persists.	
200		OK – Success



5. List uploads - training-provider/{id}/uploads

HTTP Status Codes	Message	Description
400	Your query does not conform to the Basic HTTP Authentication Scheme. See https://tools.ietf.org/html/rfc7617 for more information.	
401	Invalid username/password.	Ensure that the correct user and password is used to authenticate
403	Account is not API enabled.	The user used has not been setup to allow API usage.
403	You do not have permission to perform this action.	The user used has not been setup to allow API usage. Verify that the correct user is used. If correct user, contact GWO to request to allow user to use the API.
404	Not found	
429	API request quota exceeded.	The user has exceeded the allowed limit of calls to the API. Investigate if the usage pattern has changed and if the change is valid. If the change is valid and needed then request GWO to increase the API rate limit of the user.
500	Error trying to access data that is currently being used by someone else please try again later.	Try to refresh the page and make sure the purchase upload is not open in more than one page.
500	Something went wrong, please contact an administrator if the problem persists.	
200		OK – Success

6. Search - training-provider/{id}/search

HTTP Status Codes	Message	Description
-------------------	---------	-------------



400	{Amount} Winda IDs supplied, 100 is the limit.	Search for CPs using WINDA ID is limited to 100 users at a time.
400	Your query does not conform to the Basic HTTP Authentication Scheme. See https://tools.ietf.org/html/rfc7617 for more information.	
401	Invalid username/password.	Ensure that the correct user and password is used to authenticate
403	Account is not API enabled.	The user used has not been setup to allow API usage.
403	You do not have permission to perform this action.	The user used has not been setup to allow API usage. Verify that the correct user is used. If correct user, contact GWO to request to allow user to use the API.
429	API request quota exceeded.	The user has exceeded the allowed limit of calls to the API. Investigate if the usage pattern has changed and if the change is valid. If the change is valid and needed then request GWO to increase the API rate limit of the user.
500	Error trying to access data that is currently being used by someone else please try again later.	Try to refresh the page and make sure the purchase upload is not open in more than one page.
500	Something went wrong, please contact an administrator if the problem persists.	
200		OK – Success

7. Search - organization/{oid}/search

HTTP Status Codes	Message	Description
400	{Amount} Winda IDs supplied, 100 is the limit.	Search for CPs using WINDA ID is limited to 100 users at a time.



400	Your query does not conform to the Basic HTTP Authentication Scheme. See https://tools.ietf.org/html/rfc7617 for more information.	
401	Invalid username/password.	Ensure that the correct user and password is used to authenticate
403	Account is not API enabled.	The user used has not been setup to allow API usage.
403	You do not have permission to perform this action.	<p>The user used has not been setup to allow API usage.</p> <p>Verify that the correct user is used. If correct user, contact GWO to request to allow user to use the API.</p>
429	API request quota exceeded.	<p>The user has exceeded the allowed limit of calls to the API.</p> <p>Investigate if the usage pattern has changed and if the change is valid. If the change is valid and needed then request GWO to increase the API rate limit of the user.</p>
500	The received WINDA IDs contained nulls or whitespaces, please create a new call with the desired fields entered correctly.	WINDA ID is typed incorrectly with nulls or whitespaces between, before or after the WINDA ID.
500	Error trying to access data that is currently being used by someone else please try again later.	Try to refresh the page and make sure the purchase upload is not open in more than one page.
500	Something went wrong, please contact an administrator if the problem persists.	
200		OK – Success