

# Plagiat.pl - API v2

---

Last update: 12.07.2022, Author: **Michał Olszewski** API version: 2.8.5, Document version: 2.8.5

---

## Table Of Contents

1. Changes in API
2. Introduction
3. Uploading documents and receiving Similarity Reports
4. Users management
5. Organisation management

## Changes in API

---

### v 2.8.5 (2022-07-12)

- added cross check information to `/api/v2/documents` endpoint
- added `userId` optional parameter to `api/v2/documents/add` endpoint

### v 2.8.0 (2022-01-25)

- added `/api/v2/documents/reject` endpoint
- changed type of `assignmentId` parameter of `api/v2/documents/add` endpoint to String

### v 2.7.0 (2021-12-28)

- added similarity coefficients to `/api/v2/documents` endpoint

### v 2.6.4 (2021-11-24)

- added `assignmentId` optional parameter to `api/v2/documents/add` endpoint

### v 2.6.3 (2021-11-21)

- minor changes

### v 2.6.2 (2021-02-18)

- added `expirationRedirection` parameter to interactive report token request

### v 2.6.1 (2021-02-08)

- added `lang` parameter to `api/v2/documents/report` endpoint

### v 2.6.0 (2020-12-01)

- instructions for accessing interactive report

### v 2.5.0 (2019-05-21)

- Changes in entry point: GET `api/v2/documents/report` : 1. added two request params: `short` Boolean short or full report. Default is full `pdf` Boolean pdf or html report. Default is html 2. Changes in response: Media-type `"text/pdf; charset=UTF-8"`, if parameter `pdf` was true

### v. 2.4.1

1. new method `documents/remove` for removing document from database

### v. 2.4

1. new method `documents/remove-from-database` for removing documents from reference database

### v. 2.3

1. information about document size added in method `documents/`

### v. 2.2

1. new method `documents/add-to-database` that adds the document to reference database.

### v. 2.1

1. `documents/get` returns fingerprint (MD5 sum) of the source document sent or analysis
2. `documents/get` returns `foreign-alphabet-alert` - anti-cheat alert that counts characters that don't match base language character set and additional 3 coefficients (5 in total)

# Introduction

---

This document describes API to the antiplagiarism service. This interface is intended to be used by 3<sup>rd</sup> party systems to:

1. Manage operators' and users' accounts
2. Create and manage organisational structure
3. Upload documents for anti-plagiarism analysis and receive reports.

## Definitions

---

### System (API)

The antiplagiarism service that exposes API according to the specification described here.

### Document

The piece of work which contains text in any of languages recognised by the System, which the customer is intended to check for plagiarism. The document should be delivered in a text (.txt) or binary format. In the latter case, the format should be one of the following:

- Microsoft Word 6.0/95/97/2000/XP (.doc)
- Microsoft Word 2007 XML (.docx)
- OpenDocument format (.odt)
- Portable Document Format (.pdf)

### (Similarity) Report

The result of the anti-plagiarism analysis in a form of HTML, UTF-8 encoded document. The report consists of coverage coefficients and references within the analysed document that point to potential borrowings.

### Client Application (Application)

The information system that users Plagiat.pl API to manage Clients account, send documents to anti-plagiarism analysis and receive reports

### Callback URL

URL provided by Application that should be requested after the report is ready.

### API Key (Key)

Credential information provided by Plagiat.pl Customer Service as a part of the licence agreement. The Application that refers the Key uses API on behalf of the customer (as a part of URL, example of key value: f74c00cad0540d5c910a26591a55f957 ). There is usually one API Key per customer account nevertheless Customer can ask for next Keys or request blocking of the existing Key.

There are 2 levels of the API keys:

- user-level API key
- client-level API key

The client-level API key must be used to manage users and organisational units. The user-level API key can be used to manage documents and reports.

## The analysis workflow

---

The workflow consists of two main parts: the first is started by Application that is intended to check the document for potential plagiarism. The second is started by the System that sends a read report to LMS.

At the beginning, Client Application sends the document along with some metadata and also a callback URL. If the request meets the specification and credentials are valid, the system sends back both the status message and document ID - please notice that if the ID is not provided during the request, new GUID will be created. From that moment the status of the analysis process can be considered as "in progress". In the case the request doesn't meet the specification or the credentials are invalid (or some

communication/internal problems occurred) the appropriate message is send back to the Application.

When the report is ready, System sends a message to the callback URL, the same that has been received with the document. The message consists of document ID. As the response, Application confirms (HTTP OK) that it received the notification.

The final step is initiated by Client Application that (being assured the report is ready) sends the request for the report and receives it in return.

```
Client->API: Document + metadata
API->Client: HTTP:OK + ID
API-->AntiPlagiarism Service:Document + metadata
Note right of AntiPlagiarism Service: Document is analysing
Note over AntiPlagiarism Service: .....
Note right of AntiPlagiarism Service: Report is ready
AntiPlagiarism Service-->API: Report is ready
API->Client: Notification + ID
Client->API: HTTP:OK
Client->API: Report request + ID
API->Client: The report
```

## Credentials

For all the communication, a valid API Key is needed. The Key is delivered by the Customer Service.

The API Key can have two levels of authorization:

- Account level - the key can be used to upload the documents for all users connected to the account (user Email needs to be explicitly provided during document upload)
- User level - the key is constrained to specific User. In that case user Email is optional but if provided, it has to match the user connected to the key.

# Uploading documents and receiving Similarity Reports

## Upload document

### Request

Entry point: POST <http://lmsapi.plagiat.pl/api/v2/documents/add> Consumes: "multipart/form-data"

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
languageCode	ISO <sup>1</sup>	N	N	leading language of the document
faculty	String	Y	N	the name of the faculty
action	String	Y	N	check ( <i>default</i> ) for antiplagiarism analysis index for adding to reference database only
callback	URL	Y	N	callback URL
userEmail	String	Y <sup>2</sup>	N	email address of the user.

Name	Type	Opt?	Multi	Description
title	String	N	N	title of the document
author	String	N	Y	name of author or co-author
coordinator	String	N	Y	name of the promoter
reviewer	String	Y	Y	name of the reviewer
documentKind	String <sup>3</sup>	Y	N	kind of the document (number)
assignmentId	String	Y	N	assignment id from the external system <sup>4</sup>
userId	String	Y	N	user id from the external system
file	Binary file <sup>5</sup>	N	N	binary content of the document

<sup>1</sup> ISO 639-1 2-letter language codes followed by ISO 3166-1 A-2 - two-letter country code when relevant (e.g. en-gb)

<sup>2</sup> email is optional only if APIKEY is constrained to user.

<sup>3</sup> supports following numbers: 0 - Habilitation Thesis; 1 - Doctoral thesis; 2 - Master's thesis; 3 - Licentiate; 4 - Year Paper; 6 - Essay; 7 - Article; 8 - Case Study; 9 - Engineer; 10 - Diploma Paper; 16 - Project; 25 - Paper; 29 - Other; 40 - Postgraduate Thesis; 101 - Monograph; 103 - Treatise; 300 - Book;

<sup>4</sup> When passed and the cross-check is enabled on the account - the cross-check section will be visible in the interactive report

<sup>5</sup> "application/octet-stream" Media type.

## Response

Response: Format: "application/json; charset=UTF-8"

Status code	Message	Description
200	OK	document has been delivered, the analysis is scheduled
403	Forbidden	API key is incorrect or has been expired
400	Bad request	The values of the parameter does not follow specification
500	Internal Server Error	Some internal error occurred

Message is a JSON formatted list of parameters:

Parameter	Description
status	Literal: new or error
message	The comment to the response

Parameter id	Description
	Identifier either provided explicitly in request or generated automatically (GUID)

Example of successful response:

```
{
  "status": "new",
  "message": "New document has been added",
  "id": "b207015a-4909-4c7c-9f88-18fce7a3bbe6"
}
```

Example of unsuccessful response:

```
{
  "status": "error",
  "message": "Missing file"
}
```

## Remove document from database

### Request

Entry point: POST <http://lmsapi.plagiat.pl/api/v2/documents/remove> Consumes: "application/x-www-form-urlencoded"

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
id	String	N	N	document ID

### Response

Response: Format: "application/json; charset=UTF-8"

Status code	Message	Description
200	OK	document has been removed from database
403	Forbidden	API key is incorrect or has been expired
400	Bad request	The values of the parameter does not follow specification
500	Internal Server Error	Some internal error occurred

### Report notification - request

After the antiplagiarism report is ready, the system calls `callback` URL provided during document upload. In the request, the ID of the document is provided.

Sends: HTTP: POST `"multipart/form-data"`

Name	Type	Opt?	Multi	Description
<code>id</code>	String	N	N	ID - the same that has been responded when the document has been sent

## Report notification - response

Response the System is expecting from Client Application: Format: `"application/json; charset=UTF-8"`

Status code	Message	Description
200	OK	report notification has ben successfully delivered
400	Bad request	The values are missing or do not follow specification
500	Internal Server Error	Some internal error occurred

Message is a JSON formatted list of parameters:

Parameter	Description
<code>status</code>	Literal: <code>new</code> or <code>error</code>
<code>message</code>	The comment to the response

Example of successful response:

```
{
  "status": "new",
  "message": "Notification received"
}
```

Example of unsuccessful response:

```
{
  "status": "error",
  "message": "Access denied"
}
```

## Add document to reference database

### Request

Entry point: POST `http://lmsapi.plagiat.pl/api/v2/documents/add-to-database` Consumes: `"application/x-www-`

form-urlencoded"

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
id	String	Y	N	document ID

## Response

Response: Format: "application/json; charset=UTF-8"

Status code	Message	Description
200	OK	document has been added to database
403	Forbidden	API key is incorrect or has been expired
400	Bad request	The values of the parameter does not follow specification
500	Internal Server Error	Some internal error occurred

## Remove document from reference database

### Request

Entry point: POST <http://lmsapi.plagiart.pl/api/v2/documents/remove-from-database> Consumes: "application/x-www-form-urlencoded"

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
id	String	N	N	document ID

### Response

Response: Format: "application/json; charset=UTF-8"

Status code	Message	Description
200	OK	document has been removed from index
403	Forbidden	API key is incorrect or has been expired
400	Bad request	The values of the parameter does not follow specification

Status code	Message	Description
500	Internal Server Error	Some internal error occurred

## Reject report

---

### Request

Entry point: POST `http://lmsapi.plagiat.pl/api/v2/documents/reject` Consumes: `"application/x-www-form-urlencoded"`

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
id	String	N	N	document ID

### Response

Response: Format: `"application/json; charset=UTF-8"`

Status code	Message	Description
200	OK	report has been rejected
403	Forbidden	API key is incorrect or has been expired
400	Bad request	The values of the parameter does not follow specification
500	Internal Server Error	Some internal error occurred

## Get document

---

### Request

Entry point: GET `http://lmsapi.plagiat.pl/api/v2/documents`

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
id	String	N	N	document ID
md5sum	String	N	N	MD5 sum of the binary document sent for analysis

Document can be identified either by `id` or `md5sum`. At least one of those parameters should be provided.



## Response

Media-type: "application/json; charset=UTF-8"

Status code	Description	Comment
200	OK	response contains document details
403	Forbidden	API key is incorrect or has been expired
400	Bad request	The values of the parameters does not follow specification
500	Internal Server Error	Some internal error occurred

Message is a JSON formatted list of parameters:

Parameter	Type	Description
title	String	Title of the thesis
author	String	Author(s) name
coordinator	String	The coordinator / promoter
report-ready	Boolean	true if the report is ready for download, false otherwise
indexed	Boolean	true if the document has been indexed
factor1	Number	The first factor / coefficient
factor2	Number	The second factor / coefficient
factor3	Number	The third factor / coefficient
factor4	Number	The fourth factor / coefficient
factor5	Number	The fifth factor / coefficient
internetSourcesFactor	Number	The factor / coefficient of the Internet sources
localSourcesFactor	Number	The factor / coefficient of the local database sources
otherSourcesFactor	Number	The factor / coefficient of the Database Exchange Program sources
legalSourcesFactor	Number	The factor / coefficient of the Legal Acts sources
refbooksSourcesFactor	Number	The factor / coefficient of the Refooks sources
reversedFactor1	Number	Reversed factor / coefficient (if enabled)
quotationsFactor	Number	Quotations factor / coefficient (if enabled)

Parameter	Type	Description
alphabet-alert	Number	Number of characters that don't match base language character set.
cross-checks	Array	Array of the cross-checked documents information (if enabled), including <code>documentTitle</code> , <code>documentAuthor</code> and <code>similarity</code> fields <sup>1</sup>
document-size	Number	Number of characters of document.
md5sum	String	MD5 sum of the binary file sent for analysis

<sup>1</sup> `cross-checks` is present only when Assignments module is enabled AND `assignmentId` parameter has been passed when uploading the document AND there was another document with the same `assignmentId` parameter value uploaded to the system

Example of successful response:

```
{
  "title": "Test title",
  "author": "Test author",
  "coordinator": "Test person",
  "report-ready": true,
  "indexed": false,
  "factor1": 0.34,
  "factor2": 0.45,
  "document-size": 40235,
  "cross-checks": [
    {
      "documentTitle": "Other document 1",
      "documentAuthor": "John Smith",
      "similarity": 0.3237
    },
    {
      "documentTitle": "Other document 2",
      "documentAuthor": "John Marley",
      "similarity": 0
    }
  ]
}
```

Example of unsuccessful response:

```
{
  "message": "Unknown document ID"
}
```

## Get report

### Request

Entry point: GET `http://lmsapi.plagiat.pl/api/v2/documents/report`

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
id	String	N	N	document ID
md5sum	String	N	N	MD5 sum of the binary document sent for analysis
short	Boolean	Y	N	short or full report. Default is full
pdf	Boolean	Y	N	pdf or html report. Default is html
lang	String	Y	N	report language as an ISO 639-1 code. By default it is the language of user who uploaded the document.

Document can be identified either by `id` or `md5sum`. At least one of those parameters should be provided.

## Response

Media-type: - On success: `"text/html; charset=UTF-8"` or `"text/pdf; charset=UTF-8"` if parameter `pdf` was true - On failure: `"application/json; charset=UTF-8"`

Status code	Description	Comment
200	OK	response contains document details
403	Forbidden	API key is incorrect or has been expired
400	Bad request	The values of the parameters does not follow specification
500	Internal Server Error	Some internal error occurred

Message is a HTML document with the report

Example of unsuccessful response:

```
{
  "message": "Unknown document ID"
}
```

## Interactive report

### Accessing report

The interactive report is accessible as a web application on address: `https://lmsapi.plagiat.pl/report/`. JSON Web Token is required for authorization. JWT should be passed as a query parameter `auth`.

Example:

```
window.location.href = 'https://lmsapi.plagiati.pl/report/?auth=eyJraWQiOiJyZXBvcnQyMDIwIiwiaWwiYWxnIjoiaUIM
```

## Obtaining token

JWT can be obtained through REST service based on API Key. The JWT expires after 24 hours.

### Request

Entry point: POST <https://lmsapi.plagiati.pl/report/api/token>

Consumes: "application/json; charset=UTF-8"

JSON Fields:

Name	Type	Opt?	Default value	Description
apiKey	string	N	-	the valid API key
documentId	string	N	-	document ID
viewOnly	boolean	Y	true	If true, report is not editable.
expirationRedirection	string	Y	-	URL allowing to re-enter the report after JWT expires. It may be the login page address.

Example of a valid request body:

```
{
  "documentId": "7058048",
  "apiKey": "0d26a6d0-360f-11eb-adc1-0242ac120002",
  "viewOnly": true,
  "expirationRedirection": "http://xyz.com/document/4332/report"
}
```

### Response

Media-type:

- On success: "text/html; charset=UTF-8"

- On failure: "application/json; charset=UTF-8"

Status code	Description	Comment
200	OK	response contains token that needs to be passed in auth query parameter
400	Bad request	the request body is incorrect, given values could not be used to generate a valid token
500	Internal Server Error	some internal error occurred

Example of successful response:

```
eyJraWQwI0iJyZXBvcnQyMDIwIiwIYXNjIjo3MDU3MTkyLCJleHAiOjE2MDY5MDI0NDAsInZ
```

Example of unsuccessful response:

```
{
  "timestamp": "2020-12-01T10:28:45.774+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "Invalid documentId.",
  "path": "/api/token"
}
```

# Users Management

## Create user

Creating an user account.

### Request

Entry point: POST `http://lmsapi.plagiart.pl/api/v2/users/create` Consumes: `"application/x-www-form-urlencoded"`

Name	Type	Opt?	Multi	Description
<code>APIKEY</code>	String	N	N	the valid API key
<code>email</code>	String	N	N	email of the user
<code>password</code>	String	N	N	password
<code>orgName</code>	String	N	N	name of the organisation. The organisation should already exist.
<code>name</code>	String	Y	N	first name of the user
<code>surname</code>	String	N	N	surname of the user
<code>testCount</code>	Number	N	N	number of document analyses (tests) granted to the user

### Response

#### On success

When user has been created, HTTP code `200` is returned, content is empty.

## On failure

When an error occurs, non-successful HTTP code is returned and the content contains JSON document with `message` field. The `message` contains description of the failure.

Format: `"application/json; charset=UTF-8"`

Status code	Message	Description
400	Bad request	The values of the parameter does not follow specification
403	Forbidden	API key (included in the URL) is incorrect or has been expired
500	Internal Server Error	Some internal error occurred

Example of unsuccessful response:

```
{
  "message": "Missing email"
}
```

## Update user

---

### Request

Entry point: POST `http://lmsapi.plagiat.pl/api/v2/users/update` Consumes: `"application/x-www-form-urlencoded"`

Name	Type	Opt?	Multi	Description
<code>APIKEY</code>	String	N	N	the valid API key
<code>email</code>	String	N	N	existing email of the user
<code>newEmail</code>	String	N	N	new email of the user
<code>password</code>	String	N	N	new password value
<code>orgName</code>	String	N	N	name of the organisation. The organisation should already exist.
<code>name</code>	String	Y	N	first name of the user
<code>surname</code>	String	N	N	surname of the user
<code>testCount</code>	Number	N	N	number of document analyses (tests) granted to the user

### Response

## On success

When the user has been updated, HTTP code 200 is returned, content is empty.

## On failure

When the error occurs, non-successful HTTP code is returned and the content contains JSON document with `message` field. The `message` contains a description of the failure.

Format: `"application/json; charset=UTF-8"`

Status code	Message	Description
400	Bad request	The values of the parameter does not follow specification
403	Forbidden	API key (included in the URL) is incorrect or has been expired
500	Internal Server Error	Some internal error occurred

Example of unsuccessful response:

```
{
  "message": "Missing email"
}
```

# Delete user

---

## Request

Entry point: DELETE `http://lmsapi.plagiart.pl/api/v2/users` Consumes: `"application/x-www-form-urlencoded"`

Authorisation: only APIKEY with account privileges

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
email	String	N	N	email of the user that should be deleted

## Response

### On success

When user has been deleted, HTTP code 200 is returned, content is empty.

### On failure

When the error occurs, non-successful HTTP code is returned and the content contains JSON document with `message` field. The `message` contains a description of the failure.

Format: `"application/json; charset=UTF-8"`

Status code	Message	Description
400	Bad request	The values of the parameter does not follow specification
403	Forbidden	API key (included in the URL) is incorrect or has been expired
500	Internal Server Error	Some internal error occurred

Example of unsuccessful response:

```
{
  "message": "Missing email"
}
```

## Get user

---

### Request

Entry point: GET `http://lmsapi.plagiat.pl/api/v2/users`

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
email	String	N	N	email of the requested user

### Response

#### On success

On success HTTP code `200` is returned, and the content contains a JSON formatted list of parameters:

Parameter	Type	Description
orgUnitName	String	Name of organisational unit, the user is related to
email	String	User email
name	String	First name of the user
surname	String	Surname of the user
testCount	Number	remaining number of the analyses (tests) allowed by the user

Example of successful response:



```
{
  "orgUnitName": "Department of Economics",
  "email": "john.smith@university.com",
  "name": "John",
  "surname": "Smith",
  "testCount": 35
}
```

## On failure

When the error occurs, non-successful HTTP code is returned and the content contains JSON document with `message` field. The `message` contains a description of the failure.

Format: "application/json; charset=UTF-8"

Status code	Message	Description
400	Bad request	The values of the parameter does not follow specification
403	Forbidden	API key (included in the URL) is incorrect or has been expired
500	Internal Server Error	Some internal error occurred

Example of unsuccessful response:

```
{
  "message": "Missing email"
}
```

## List users

---

### Request

Entry point: GET `http://lmsapi.plagiat.pl/api/v2/users/list`

Query parameters:

Name	Type	Opt?	Multi	Description
<code>APIKEY</code>	String	N	N	the valid API key
<code>orgName</code>	String	Y	N	search by organisation name
<code>email</code>	String	Y	N	search by email
<code>name</code>	String	Y	N	search by name
<code>surname</code>	String	Y	N	search by surname

## Response

### On success

On success HTTP code 200 is returned, and the content contains a JSON with one entry `users` that contains the list of following parameters for each of users found:

Parameter	Type	Description
orgUnitName	String	Name of organisational unit, the user is related to
email	String	User email
name	String	First name of the user
surname	String	Surname of the user
testCount	Number	remaining number of the analyses (tests) allowed by the user

Example of successful response:

```
{ "users" : [  
  {  
    "orgUnitName": "Department of Economics",  
    "email": "john.smith@university.com",  
    "name": "John",  
    "surname": "Smith",  
    "testCount": 35  
  }, {  
    "orgUnitName": "Department of Economics",  
    "email": "tom.bradley@university.com",  
    "name": "Tomas",  
    "surname": "Bradley",  
    "testCount": 12  
  }  
]
```

### On failure

When the error occurs, non-successful HTTP code is returned and the content contains JSON document with `message` field. The `message` contains a description of the failure.

Format: "application/json; charset=UTF-8"

Status code	Message	Description
400	Bad request	The values of the parameter does not follow specification
403	Forbidden	API key (included in the URL) is incorrect or has been expired

Status code	Message	Description
500	Internal Server Error	Some internal error occurred

Example of unsuccessful response:

```
{
  "message": "Missing APIKEY"
}
```

# Organisation management

## Create organisation

### Request

Entry point: POST `http://lmsapi.plagiat.pl/api/v2/orgs/create` Consumes: `"application/x-www-form-urlencoded"`

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
name	String	N	N	name of the new organisation unit

### Response

#### On success

When the organisation unit has been created, HTTP code `200` is returned, content is empty.

#### On failure

When the error occurs, non-successful HTTP code is returned and the content contains JSON document with `message` field. The `message` contains a description of the failure.

Format: `"application/json; charset=UTF-8"`

Status code	Message	Description
400	Bad request	The values of the parameter does not follow specification
403	Forbidden	API key (included in the URL) is incorrect or has been expired
500	Internal Server Error	Some internal error occurred

Example of unsuccessful response:

```
{
  "message": "Missing name"
}
```

## Update organisation

---

### Request

Entry point: POST `http://lmsapi.plagiati.pl/api/v2/orgs/update` Consumes: `"application/x-www-form-urlencoded"`

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
name	String	N	N	existing name of the organisation unit
newName	String	N	N	new name of the organisation unit

### Response

#### On success

When the organisation unit has been updated HTTP code `200` is returned, content is empty.

#### On failure

When the error occurs, non-successful HTTP code is returned and the content contains JSON document with `message` field. The `message` contains a description of the failure.

Format: `"application/json; charset=UTF-8"`

Status code	Message	Description
400	Bad request	The values of the parameter does not follow specification
403	Forbidden	API key (included in the URL) is incorrect or has been expired
500	Internal Server Error	Some internal error occurred

Example of unsuccessful response:

```
{
  "message": "Missing name"
}
```

## Delete organisation

---

## Request

Entry point: DELETE `http://lmsapi.plagiat.pl/api/v2/orgs` Consumes: "application/x-www-form-urlencoded"

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key
name	String	N	N	name of the organisation unit to be removed

## Response

### On success

When the organisation unit has been deleted HTTP code 200 is returned, content is empty.

### On failure

When the error occurs, non-successful HTTP code is returned and the content contains JSON document with `message` field. The `message` contains a description of the failure.

Format: "application/json; charset=UTF-8"

Status code	Message	Description
400	Bad request	The values of the parameter does not follow specification
403	Forbidden	API key (included in the URL) is incorrect or has been expired
500	Internal Server Error	Some internal error occurred

Example of unsuccessful response:

```
{
  "message": "Missing name"
}
```

## Get organisations

---

### Request

Entry point: GET `http://lmsapi.plagiat.pl/api/v2/orgs`

Name	Type	Opt?	Multi	Description
APIKEY	String	N	N	the valid API key

## Response

### On success

On success HTTP code 200 is returned, and the content contains a JSON formatted list of parameters:

Parameter	Type	Description
orgUnitName	String	Name of organisational unit, the user is related to
email	String	User email
name	String	First name of the user
surname	String	Surname of the user
testCount	Number	remaining number of the analyses (tests) allowed by the user

Example of successful response:

```
{
  "orgUnitName": "Department of Economics",
  "email": "john.smith@university.com",
  "name": "John",
  "surname": "Smith",
  "testCount": 35
}
```