

REST API

Table of Contents

- Table of Contents
- General Information
- Authentication
- Resources
 - Language
 - Client
 - Scenario Template
 - Attachment Template
 - Awareness Template
 - Campaign
 - Scenario
 - Recipient Group
 - Recipient
 - Victim
 - Incident
- Endpoint List
 - /api/auth
 - /api/languages
 - /api/clients
 - /api/clients/:id
 - /api/recipient-groups
 - /api/recipient-groups/:id
 - /api/recipient-groups/:id/recipients
 - /api/recipients/:id
 - /api/scenario-templates
 - /api/scenario-templates/:id
 - /api/awareness-templates
 - /api/awareness-templates/:id
 - /api/attachment-templates
 - /api/attachment-templates/:id
 - /api/campaigns
 - /api/campaigns/:id
 - /api/campaigns/:id/recipient-groups
 - /api/campaigns/:id/status
 - /api/campaigns/:id/copy
 - /api/campaigns/:id/victims
 - /api/campaigns/:id/scenarios
 - /api/scenarios/:id
 - /api/scenarios/:id/recipient-groups
 - /api/scenarios/:id/victims
 - /api/incidents
 - /api/incidents/:id
 - /api/incidents/:id

General Information

Lucy API is a RESTful web service. The API does not keep the client's state and all requests can be performed in any order.

Some general requirements are:

- The service uses JSON as the only data interchange format, all API responses are in JSON, most request parameters are expected to be in JSON. All requests to API should have a "Content-Type" header set to "application/json".
- All requests should be performed strictly over HTTPS protocol. Plain HTTP requests to API will be rejected by server.

Authentication

Lucy API uses JWT (JSON web tokens) for authentication. The token should be obtained by sending an [Authentication](#) request before issuing any other requests (see [Authentication](#) request description below for details).

The obtained authentication token should be specified in the "Authorization" HTTP header with "Bearer" scheme, for all subsequent requests. The

only request that doesn't require authentication is the authentication request itself, all other requests require a token. Example:

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ
```

The client should store the token locally and keep it until the token expires. The expiration date and time is specified in the token header. Please check out <https://jwt.io/> for JWT token formation description and libraries for various programming languages.

Resources

Resources are used when getting some data from Lucy API or creating/changing some objects in Lucy. Resource structure is always the same and does not depend on the type of action you do - whether you are getting information from Lucy or creating an object, the object representation doesn't change. For example, campaigns - when you get a list of campaigns or get a single campaign, the structure doesn't differ from what the server expects when you are creating a new campaign.

The only difference is links - when you are posting/putting something to API, the server does not expect any links (and ignores them if they are present in request). The links are only used for exposing relations in GET requests. For specifying relations with other existing objects in system, please use integer IDs.

Language

```
{
  "id": int,
  "name": str,
  "default": boolean
}
```

Client

```
{
  "id": int,
  "name": str,
  "country": str,
  "state": str,
  "city": string,
  "address": str,
  "postcode": str,
  "website": str,
  "email": str,
  "phone": str,
  "fax": str,
  "contact": str
}
```

Scenario Template

```
{
  "id": int,
  "created_at": datetime,
  "updated_at": datetime,
  "type":
  "data|file|mixed|hyperlink|malware_test|usb_attack|awareness",
  "client_id": int,
  "name": str,
  "description": str,
  "no_landing": boolean,
  "category": str,
  "audience": str,
  "message_type": "email|sms",
  "lure_message_type": "email|sms",

  "links": [{
    "rel": "client",
    "href": "/link/to/client"
  }]
}
```

Attachment Template

```
{
  "id": int,
  "created_at": datetime,
  "updated_at": datetime,
  "name": str,
  "description": str,
  "variables": [],
  "configurations": []
}
```

Awareness Template

```
{  
  "id": int,  
  "created_at": datetime,  
  "updated_at": datetime,  
  "name": str,  
  "description": str,  
  "email_sender_name": str,  
  "email_sender": str,  
  "quiz": boolean  
}
```

Campaign

```

{
  "id": int,
  "created_at": datetime,
  "started_at": datetime,
  "stopped_at": datetime,
  "client_id": int,
  "benchmark_sector_id": int,
  "name": str,
  "status": "idle|running|waiting_approval|stopped",

  "recipient_count": int,
  "messages_sent": int,
  "clicks": int,
  "successful_attacks": int,
  "vulnerable_victims": int,
  "errors": int,

  "awareness_submitted": int,
  "awareness_opened": int,
  "completed_quiz": int,
  "average_quiz": int,

  "links": [{
    "rel": "client",
    "href": "/link/to/client"
  }, {
    "rel": "benchmark_sector",
    "href": "/link/to/benchmark_sector"
  }, {
    "rel": "awareness_template",
    "href": "/link/to/awareness_template"
  }, {
    "rel": "language",
    "href": "/link/to/language"
  }]
}

```

Scenario

```

{
  "id" : int,
  "created_at" : datetime,
  "started_at" : datetime,
  "stopped_at" : datetime,
  "campaign_id" : int,
  "scenario_template_id" : int,

```

```
"language_id" : int,
"name" : str,

"domain" : str,
"subdomain" : str,
"full_domain" : str,
"sender_name" : str,
"sender_email" : str,
"random_email" : boolean,
"redirect_url" : str,
"collect_data" : "no|partial|full",
"login_regexp" : str,
"password_regexp" : str,
"forward_email" : str,
"message_type" : "email|sms",
"double_barrel" : boolean,
"lure_sender_name" : str,
"lure_sender_email" : str,
"lure_random_email" : boolean,
"lure_delay" : int,
"lure_message_type" : "email|sms",
"status" : "idle|running",

"send_awareness_link" : boolean,
"awareness_delay" : int,
"no_landing" : boolean,
"track_opened_emails" : boolean,
"use_reply_to" : boolean,
"anonymous_mode" : boolean,
"success_action" : "click|post|download|file_data",

"recipient_count": int,
"messages_sent": int,
"clicks": int,
"successful_attacks": int,
"vulnerable_victims": int,
"errors": int,

"links": [{
    "rel": "campaign",
    "href": "/link/to/campaign"
}, {
    "rel": "scenario_template",
    "href": "/link/to/scenario_template"
}, {
    "rel": "language",
    "href": "/link/to/language"
}]
}
```



Recipient Group

```
{
  "id": int,
  "name": str,
  "usb_attack": boolean
}
```

Recipient

```
{
  "id": int,
  "last_tested": datetime,
  "recipient_group_id": int,
  "language_id": int,
  "email": str,
  "name": str,
  "phone": str,
  "staff_type": str,
  "location": str,
  "division": str,
  "comment": str,
  "link": str,

  "links": [{
    "rel": "recipient_group",
    "href": "/link/to/recipient_group"
  }, {
    "rel": "language",
    "href": "/link/to/language"
  }]
}
```

Victim

```
{
  "id": int,
  "recipient_id": int,

  "lure_submitted_at" : datetime,
  "submitted_at" : datetime,
  "awareness_submitted_at" : datetime,
  "opened_at" : datetime,
  "clicked_at" : datetime,
  "succeeded_at" : datetime,
  "trained_at" : datetime,
  "reported_at" : datetime,

  "error": boolean,
  "os": str,
  "browser": str,
  "ip": str,
  "proxy_ip": str,
  "country": str,
  "plugins": str,

  "quiz_score": int,
  "clicks": int,

  "email": str,
  "name": str,
  "phone": str,
  "staff_type": str,
  "location": str,
  "division": str,
  "comment": str,
  "link": str,

  "links": [{
    "rel": "recipient",
    "href": "/link/to/recipient"
  }]
}
```

Incident

```
{
  "id": int,
  "email": str,
  "status":
  "open|in_progress|dismissed|simulation|real_phishing|closed",
```



```
"time": datetime,
"score": float,
"more_analysis": boolean,
"client_id": int,
"campaign_id": int,
"links": [
  {
    "rel": "self",
    "href": "/api/incidents/:id"
  },
  {
    "rel": "delete",
    "href": "/api/incidents/:id"
  }
],
"message_url": "/admin/incident/:id/message",
"subject": str,
"headers": [
  {
    "from": str,
    "ip": str,
    "by": str,
    "with": str,
    "time": str,
    "blacklisted": boolean,
    "score": float
  },
  ...
],
"domains": [
  {
    "source": str,
    "domain": str,
    "score": float,
    "checks": [
      {
        "service_name": str,
        "flagged": boolean
      },
      ...
    ]
  },
  ...
],
"body_analysis": [
  {
    "rule": str,
    "score": float
  },
  ...
]
```



```
}
```

Endpoint List

A list below contains all endpoints exposed by Lucy API. Every endpoint corresponds to some system resource or action. If endpoint returns some system resources or expects some common resource as an input, the specification will refer to that resource using "Resource:Resource Name" mnemonic. For example, if you see "Resource:Campaign" in request/response body, that means you should refer to this document's [Resources](#) section and find the corresponding resource there.

List endpoints support 3 optional GET parameters, that you can use to implement ordering and limits:

- sort - column to sort by. The default sorting works in ascending order, if you wish to sort in descending order, specify a minus sign before the column name (Example: sort=-created_at)
- offset - offset results to the specified number of entries
- limit - max number of entries

Example:

```
GET /api/campaigns?sort=-created_at&offset=10&limit=10
```

/api/auth

POST

- Client authentication
- Request:

```
{
  "email": "user email",
  "password": "user password"
}
```

- Response:

```
{
  "token": "authentication token"
}
```

/api/languages

GET

- Get a list of languages
- Resources:
 - [Language](#)
- Response:

```
{
  "languages": ["Resource:Language", "Resource:Language", ...]
}
```

/api/clients

GET

- Get a list of clients
- Resources:
 - [Client](#)
- Response:

```
{
  "clients": ["Resource:Client", "Resource:Client", ...]
}
```

PUT

- Create a client
- Resources:
 - [Client](#)
- Request:

```
"Resource:Client"
```

Mandatory fields:

- name
- Response:

```
{
  "client": "Resource:Client"
}
```

/api/clients/:id

GET

- Get client
- Resources:
 - [Client](#)
- Response:

```
{
  "client": "Resource:Client"
}
```

POST

- Save client
- Resources:
 - Client
- Request:

```
"Resource:Client"
```

Mandatory fields:

- name
- Response:

```
{
  "client": "Resource:Client"
}
```

/api/recipient-groups

GET

- Get a list of recipient groups
- Resources:
 - Recipient Group
- Response:

```
{
  "recipient_groups": ["Resource:Recipient Group",
    "Resource:Recipient Group", ...]
}
```

PUT

- Create a recipient group
- Resources:
 - Recipient Group
- Request:

```
"Resource:Recipient Group"
```

Mandatory fields:

- name
- usb_attack
- Response:

```
{  
  "recipient_group": "Resource:Recipient Group"  
}
```

/api/recipient-groups/:id

GET

- Get recipient group
- Resources:
 - [Recipient Group](#)
- Response:

```
{  
  "recipient_group": "Resource:Recipient Group"  
}
```

POST

- Save recipient group
- Resources:
 - [Recipient Group](#)
- Request:

```
"Resource:Recipient Group"
```

Mandatory fields:

- name
- usb_attack
- Response:

```
{  
  "recipient_group": "Resource:Recipient Group"  
}
```

/api/recipient-groups/:id/recipients

GET

- Get a list of recipients in group
- Resources:
 - [Recipient](#)
- Response:

```
{
  "recipients": ["Resource:Recipient", "Resource:Recipient", ...]
}
```

PUT

- Create a recipient
- Resources:
 - Recipient
- Request:

```
"Resource:Recipient"

OR

["Resource:Recipient", "Resource:Recipient", ...]
```

Mandatory fields:

- name, email
- Response:

```
{
  "recipient": "Resource:Recipient"
}
```

/api/recipients/:id

GET

- Get recipient
- Resources:
 - Recipient
- Response:

```
{
  "recipient": "Resource:Recipient"
}
```

POST

- Save recipient
- Resources:
 - Recipient
- Request:

```
"Resource:Recipient"
```

Mandatory fields:

- name
- Response:

```
{  
  "recipient": "Resource:Recipient"  
}
```

/api/scenario-templates

GET

- Get a list of scenario templates
- Resources:
 - [Scenario Template](#)
- Response:

```
{  
  "scenario_templates": ["Resource:Scenario Template",  
    "Resource:Scenario Template", ...]  
}
```

/api/scenario-templates/:id

GET

- Get scenario template
- Resources:
 - [Scenario Template](#)
- Response:

```
{  
  "scenario_template": "Resource:Scenario Template"  
}
```

/api/awareness-templates

GET

- Get a list of awareness templates
- Resources:
 - [Awareness Template](#)
- Response:


```
{
  "awareness_templates": ["Resource:Awareness Template",
"Resource:Awareness Template", ...]
}
```

/api/awareness-templates/:id

GET

- Get awareness template
- Resources:
 - Awareness Template
- Response:

```
{
  "awareness_template": "Resource:Awareness Template"
}
```

/api/attachment-templates

GET

- Get a list of attachment templates
- Resources:
 - Attachment Template
- Response:

```
{
  "attachment_templates": ["Resource:Attachment Template",
"Resource:Attachment Template", ...]
}
```

/api/attachment-templates/:id

GET

- Get attachment template
- Resources:
 - Attachment Template
- Response:

```
{
  "attachment_template": "Resource:Attachment Template"
}
```

/api/campaigns

GET

- Get a list of campaigns
- Resources:
 - Campaign
- Response:

```
{
  "campaigns": ["Resource:Campaign", "Resource:Campaign", ...]
}
```

PUT

- Create a campaign
- Resources:
 - Campaign
- Request:

```
"Resource: Campaign"
```

Mandatory fields:

- name
 - client_id
- Response:

```
{
  "campaign": "Resource: Campaign"
}
```

/api/campaigns/:id

GET

- Get campaign
- Resources:
 - Campaign
- Response:

```
{
  "campaign": "Resource: Campaign"
}
```

POST

- Save campaign
- Resources:
 - Campaign
- Request:

```
"Resource: Campaign"
```

Mandatory fields:

- name
- client_id
- Response:

```
{
  "campaign": "Resource: Campaign"
}
```

/api/campaigns/:id/recipient-groups

GET

- Get a list of recipient groups in campaign
- Resources:
 - Recipient Group
- Response:

```
{
  "recipient_groups": ["Resource: Recipient Group",
    "Resource: Recipient Group", ...]
}
```

PUT

- Add a recipient group to a campaign (recipients will be added to all scenarios in campaign)
- Resources:
 - Recipient Group
- Request:

```
{
  "id": int
}
```

- Response:

```
{
  "recipient_group": "Resource:Recipient Group"
}
```

/api/campaigns/:id/status

POST

- Change campaign status. This endpoint intended to run or stop a campaign, therefore it supports only POST request and only two statuses
- Resources:
 - [Campaign](#)
- Request:

```
{
  "status": "running|stopped"
}
```

- Response:

```
{
  "campaign": "Resource:Campaign"
}
```

/api/campaigns/:id/copy

POST

- Copy a campaign with all scenarios.
- Resources:
 - [Campaign](#)
- Request: N/A
- Response:

```
{
  "campaign": "Resource:Campaign"
}
```

/api/campaigns/:id/victims

GET

- Get campaign victims
- Resources:
 - [Victim](#)
- Response:

```
{
  "victims": ["Resource:Victim", "Resource:Victim", ...]
}
```

/api/campaigns/:id/scenarios

GET

- Get scenarios in campaign
- Resources:
 - [Scenario](#)
- Response:

```
{
  "scenarios": ["Resource:Scenario", "Resource:Scenario", ...]
}
```

PUT

- Create scenario in campaign
- Resources:
 - [Scenario](#)
- Request:

```
"Resource:Scenario"
```

Mandatory fields:

- name
 - domain
 - language_id
 - scenario_template_id
- Response:

```
{
  "scenario": "Resource:Scenario"
}
```

/api/scenarios/:id

GET

- Get scenario
- Resources:
 - Scenario
- Response:

```
{
  "scenario": "Resource:Scenario"
}
```

POST

- Save scenario
- Resources:
 - Scenario
- Request:

```
"Resource:Scenario"
```

Mandatory fields:

- name
 - domain
 - language_id
 - scenario_template_id
- Response:

```
{
  "scenario": "Resource:Scenario"
}
```

/api/scenarios/:id/recipient-groups

GET

- Get a list of recipient groups in scenario
- Resources:
 - Recipient Group
- Response:

```
{
  "recipient_groups": ["Resource:Recipient Group",
    "Resource:Recipient Group", ...]
}
```

PUT

- Add a recipient group to a scenario
- Resources:
 - Recipient Group

- Request:

```
{
  "id": int
}
```

- Response:

```
{
  "recipient_group": "Resource:Recipient Group"
}
```

/api/scenarios/:id/victims

GET

- Get scenario victims
- Resources:
 - [Victim](#)
- Response:

```
{
  "victims": ["Resource:Victim", "Resource:Victim", ...]
}
```

/api/incidents

GET

- Get a list of phishing incident reports
- Resources:
 - [Incident](#)
- Response:

```
{
  "incidents": ["Resource:Incident", "Resource:Incident", ...]
}
```

/api/incidents/:id

GET

- Get incident
- Resources:
 - [Incident](#)

- Response:

```
{  
  "incident": "Resource:Incident"  
}
```

/api/incidents/:id

DELETE

- Delete incident
- Response: N/A