

GoLismo REST API specification

Created by: Daniel García García & Mario Vilas

November 27, 2013



GoLismero REST API specification	1
Document versions.....	5
General information	7
Objective	7
Prerequisites.....	7
Working mode.....	7
Connections types.....	7
Service access	7
Security notes	7
Default user	7
Service definition	8
Service schematics	8
1. <i>Response error checking</i>	9
2. <i>Session management</i>	9
Base URI: /api/auth/	9
Session timeout.....	9
About the session	9
Sending the token.....	9
Header omission in method documentation.....	10
URI: /api/auth/login/	11
SESSION :: LOGOUT method.....	12
URI: /api/auth/logout/	12
3. <i>Audit management</i>	15
Base URI: /api/audits/	15
Audit states	16
AUDIT :: LIST method.....	17
URI: /api/audits/list/{STATE}	17
AUDIT :: CREATE method	19
URI: /api/audits/create/	19
AUDIT :: DELETE method	23
URI: /api/audits/delete/{ID}	23
AUDIT :: START method	25
URI: /api/audits/start/{ID}	25
AUDIT :: STOP method	27
URI: /api/audit/stop/{ID}	27
AUDIT :: STATE method	29
URI: /api/audits/state/{ID}	29
AUDIT :: PROGRESS method.....	31
URI: /api/audits/progress/{ID}	31
AUDIT :: RESULTS method.....	34
URI: /api/audits/results/{ID}/{FORMAT}	34
URI: /api/audits/results/summary/{ID}	37
AUDIT :: DETAILS method.....	39
URI: /api/audits/details/{ID}	39
AUDIT :: PAUSE method	41



URI: /api/audits/pause/{ID}	41
AUDIT :: RESUME method	43
URI: /api/audits/resume/{ID}	43
AUDIT :: LOG method	45
URI: /api/audits/log/{ID}	45
4. <i>User management</i>	49
Base URI: /api/users/	49
User restrictions and roles	49
USERS :: LIST method	50
URI: /api/users/list/	50
USERS :: CREATE method	51
URI: /api/users/create/	51
USERS :: EDIT method	52
URI: /api/users/edit/{ID}	52
USERS :: DELETE method	53
URI: /api/users/delete/{ID}	53
USERS :: DETAILS method	54
URI: /api/users/details/{ID}	54
5. <i>Plugin management</i>	55
Base URI: /api/plugins/	55
PLUGINS :: LIST method	56
URI: /api/plugins/list/	56
PLUGINS :: DETAILS method	57
URI: /api/plugins/details/{NAME}	57
PLUGINS :: SEARCH method	58
URI: /api/plugins/search/{TEXT}	58
6. <i>Profile management</i>	59
Base URI: /api/profiles/	59
About profiles	59
PROFILES :: LIST method	60
URI: /api/profiles/list/	60
PROFILES :: CREATE method	61
URI: /api/profiles/create/	61
PROFILES :: EDIT method	62
URI: /api/profiles/edit/{ID}	62
PROFILES :: DELETE method	63
URI: /api/profiles/delete/{ID}	63
PROFILES :: DETAILS method	64
URI: /api/profiles/details/{ID}	64
7. <i>Node administration</i>	65
Base URI: /api/nodes/	65
About nodes	65
Important notes	65
NODE :: LIST method	66
URI: /api/nodes/list/	66
NODE :: START method	67
URI: /api/nodes/start/{ID}	67
NODE :: STOP method	68
URI: /api/nodes/stop/{ID}	68
NODE :: RESTART method	69
URI: /api/nodes/restart/{ID}	69



GoLismero Project

NODE :: STATE method.....	70
URI: /api/nodes/state/{ID}/	70
NODE :: CONFIG DETAILS method.....	71
URI: /api/nodes/config/{ID}	71
NODE :: CONFIG EDIT method.....	72
URI: /api/nodes/config/edit/{ID}	72



Document versions

Comments	Date	Version
First version of the document. Format of URLs defined. Pending to be defined: POST parameters, responses and examples.	August 8, 2013	0.1
Change log: <ul style="list-style-type: none">Fixed bugs.Added new options for scans: stop/resume.Added the “web socket” option.Modified the table of new scan: add new returned parameter.Specified the timeout for the session.Specified new parameters returned for the scan status.Updated mind map.	August 28, 2013	0.2
Change log: <ul style="list-style-type: none">Translated the document to English.Removed the list of actions from the “service definition” section. It’s redundant.Changed logotype for the new version of it.Added new method for plugins: extended list	October 11, 2013	0.3
Change log: <ul style="list-style-type: none">Renamed ‘scan’ to ‘audit’.Added information about login and session management.Added information about audit managementExtended the documentation of each method.New look and feel.Redesigned URIsModified mind mapChanged structure: removed admin section.	October 15, 2013	0.4



Comments	Date	Version
<p>Change log:</p> <ul style="list-style-type: none">Added error codes in responses.Added new method for audits: results, results summary and filtered list.Added "list" word in all URIs, instead of use base URI.Modified details URI for plugins.	October 17, 2013	0.5
<p>Change log:</p> <ul style="list-style-type: none">Finished audit/create method definition.Added new method for audit: progressAdded information about the default user/password.Updated mindmap.	November 4, 2013	0.6
<p>Change log:</p> <ul style="list-style-type: none">Defined new method: audit/resultsDefined new generic error return code.Simplified target parameter specification when create an audit.Added new state to audit: "finished"	November 8, 2013	0.7
<p>Change log:</p> <ul style="list-style-type: none">Removed "canceled" state. It means the same that "stopped".Added new state for audit: "error".Added state diagram.Added explanation for each audit state.	November 9, 2013	0.8
<p>Change log:</p> <ul style="list-style-type: none">Added import action in audit.Fixed report URI name, in results end point.	November 26, 2013	0.9



GoLismero project

General information

Objective

The objective of this document is to define the API communication with the GoLismero vulnerability scanner, when it's working as service.

Prerequisites

GoLismero must be working as a service.

Working mode

GoLismero will launch an application server that will listen on a given IP address and a port specified in the configuration file.

The protocol used to communicate with the system will be **standard HTTP**.

The architecture used to interact with the system will be **standard REST**.

Connections types

GoLismero defines 2 types of connections:

- Using an HTTP standard connection.
- Using web sockets.

The type of connection will be specified in the moment of login.

Service access

The service will be accessible by the URI:

`http://SERVER_LOCATION/api/`

Security notes

GoLismero itself does not implement encrypted connections.

If you want to encrypt the communications it's recommended to use a front-end web server, between GoLismero and the Internet connection, acting as an SSL proxy.

Default user

The default user provided with API is: username: **admin**, password: **admin**

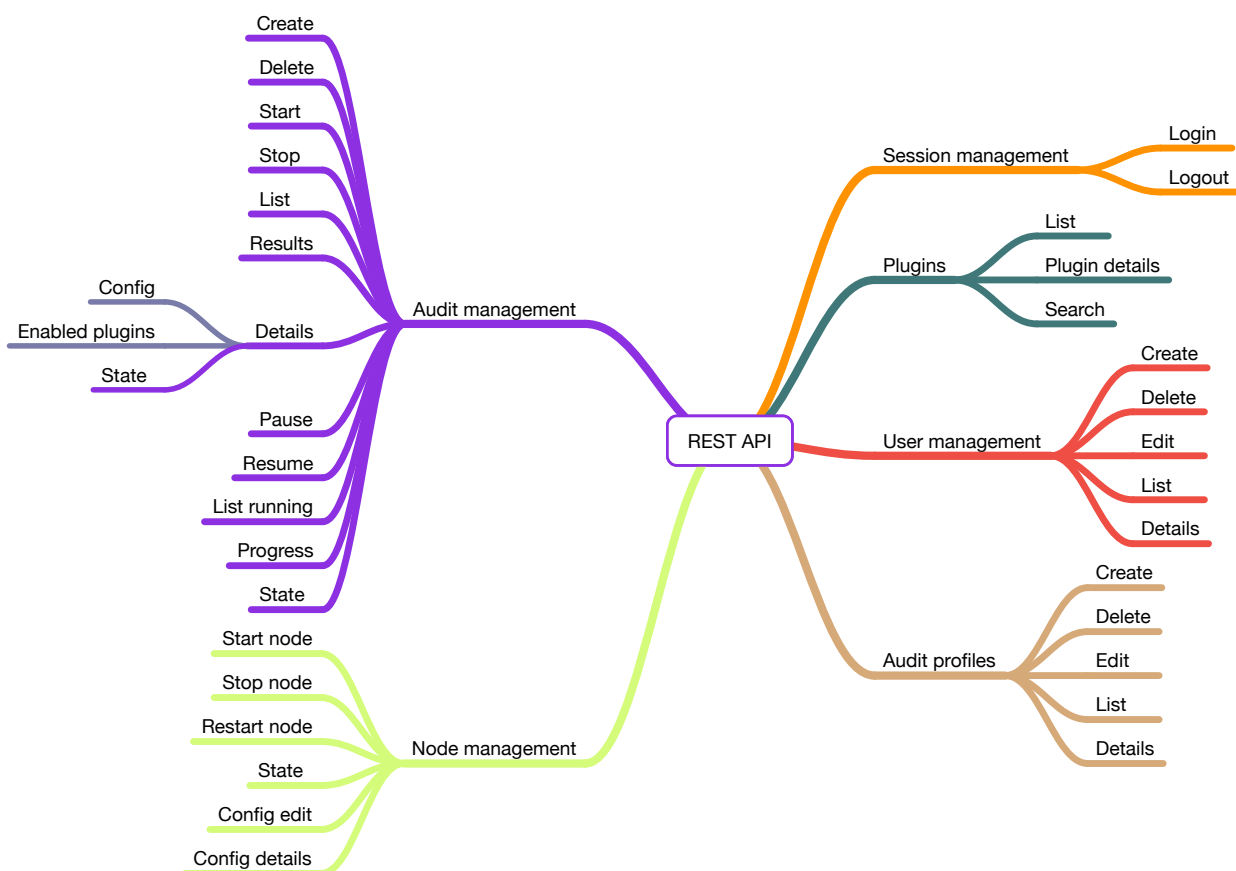


Service definition

The elements that will define the service are detailed below, as well as some useful information to understand it.

Service schematics

The following schematic shows a high-level view of the service definition:





GoLismero project

Method and actions

The GoLismero API has the following actions and methods.

1. Response error checking

To make useful the API usage, all of responses will return a parameter indicating if response was successfully and, in error cases, error descriptions.

Parameter:

- *status* :: String -> Values returned are:
 - *ok*
 - *error*
- *error* :: list(String) -> Description list, in English.
- *error_code* :: Int -> Integer indicating the code of error.

To improve readability, the method definitions in the rest of document will omit the *status* and *error*, but all of the methods will return it.

When unknown and not expected error was detected, *error_code* returned will be "-1".

2. Session management

These are the URIs defined for session management.

Base URI: /api/auth/

These are the methods defined for session management:

- login
- logout

Session timeout

Session timeout is reached if no request is done for the timeout interval.

The timeout of the session by default will be **30 minutes**.

About the session

The session management is **based on a token**. In the first request, the client must request a session token and the next request must have that token.

Sending the token

To continue within the session, the client must send the session token as part of the HTTP headers, as follows:

- The client must add an HTTP field named "*Authorization*" to the headers.
- The field must contain the word "*Token*" and the token value, separated by a blank space.
- The Token value must be 40 chars in length.

Example of a complete HTTP field:

Authorization: Token 641f170216b948d194985a17653c8fc1d1aa02ef

Minimal HTTP header request:



GoLismero project

```
GET /api/plugins/list HTTP/1.1
Host: golismero.mysite.com
Authorization: Token b9f7409391d47ba13c84e3d840b2b3cd652ea85b
```

Header omission in method documentation

To improve readability, the method definitions in the rest of document will omit the *token*, but all of the methods (except for the login method) will need the token to work.



GoLismero project

SESSION :: LOGIN method

This method allows a client to authenticate against the GoLismero application.

URI: /api/auth/login/

GET parameters

- No parameters

POST parameters

- username :: String -> A Valid username in GoLismero.
- password :: String -> String with user password.

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- Parameters:
 - *token* :: String -> Token string. The token is 40 chars long.

Error responses

The server may return the following errors:

1. Incorrect credentials
 - HTTP code: 400
 - HTTP text: "BAD REQUEST"
 - Error code: 0
2. Fields required
 - HTTP code: 400
 - HTTP text: "BAD REQUEST"
 - Error code: 1

Request example

```
POST /api/login/token/ HTTP/1.1
Host: golismero.mysite.com
Accept: */*
Content-Length: 29
Content-Type: application/x-www-form-urlencoded

username=myuser&password=mypassword
```

Response example

Normal response:

```
HTTP/1.0 200 OK
Date: Tue, 15 Oct 2013 15:34:09 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: POST, OPTIONS

{"token": "b5fbbbea58c509da5421d9bc5c6709566eadbdbc", status: "ok"}
```

Incorrect credentials

```
HTTP/1.0 400 BAD REQUEST
Date: Tue, 15 Oct 2013 15:36:05 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: POST, OPTIONS

{"non_field_errors": ["Unable to login with provided credentials."], status: "error", "error_code": 1}
```

Field required

```
HTTP/1.0 400 BAD REQUEST
Date: Tue, 15 Oct 2013 15:37:42 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: POST, OPTIONS

{"password": ["This field is required."], status: "error", "error_code": 0}
```

SESSION :: LOGOUT method

This method finalizes the session with GoLismero. It can be called using GET or POST HTTP methods.

URI: /api/auth/logout/

GET parameters

- No parameters



POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- Parameters:
 - *logout* :: String -> String with the text **"ok"**.

Error responses

The server may return the following errors:

1. Session does not exist

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 0
- Parameters:
 - *logout* :: String -> String with the text **"error"**
 - *error* :: list(String) -> ["Session does not exist"]
 - *error_code* :: Int -> Number specifying error type.

2. Invalid session token

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 1
- Parameters:
 - *logout* :: String -> String with the text **"error"**
 - *error* :: list(String) -> ["Invalid session token"]
 - *error_code* :: Int -> Number specifying error type.

Request example

```
POST /api/auth/logout/ HTTP/1.1
Host: golismero.mysite.com
Accept: */*
Authorization: Token f0d09476ce26847ba3ea7c7deee035dd868d222daa
```

Response example

Normal response:



GoLismero project

```
HTTP/1.0 200 OK
Date: Tue, 15 Oct 2013 20:08:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: POST, OPTIONS

{"status": "ok"}
```

Session does not exist

```
HTTP/1.0 400 BAD REQUEST
Date: Tue, 15 Oct 2013 20:13:22 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: POST, OPTIONS

{"status": "error", "error": ["Session does not exist"]}
```

Invalid session token

```
HTTP/1.0 400 BAD REQUEST
Date: Tue, 15 Oct 2013 20:13:22 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: POST, OPTIONS

{"status": "error", "error": ["Invalid session token"]}
```



GoLismero project

3. Audit management

These are the URIs defined for audit management:

Base URI: /api/audits/

These are the available audit management methods:

- list
- create
- delete
- start
- stop
- state
- results
- details
- pause
- resume
- log

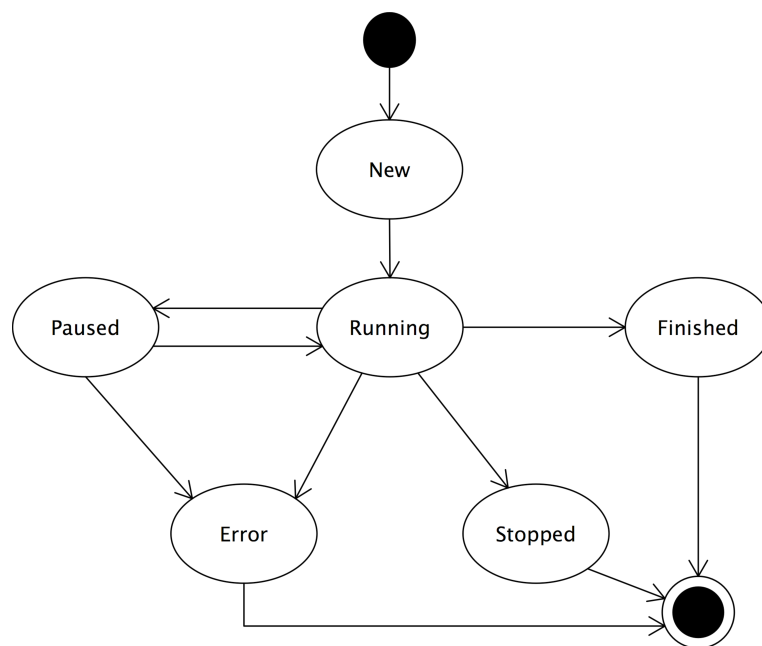


Audit states

An audit can be in some states, depending of user actions or internal state. Available audit states are:

- new: Created but not started.
- running: Audit is running.
- stopped: Audit is canceled by the user.
- paused: Audit is paused by the user.
- error: Audit switch to state when some errors occurs while running or while try to restart a paused audit.
- finished: Audit is finished well.

Next image shows the diagram state:





AUDIT :: LIST method

This method shows all audits. If the caller has the administrator role, all audits will be returned. Otherwise, if the caller is a normal user, only their own audits will be returned.

URI: `/api/audits/list/{STATE}`

GET parameters

- STATE :: String -> Optional value that indicates if method will return all audits or running audits. If no value specified, default option will be "all". Available states are defined at beginning of this section.

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- *results* :: list(dictionary) ->
 - *id* :: Number -> Audit ID
 - *name* :: String -> Audit name.
 - *user* :: String -> Audit owner.
 - *state* :: String -> Audit state. States was defined at de beginning of this section.

Error responses

The server may return the following errors:

1. Unknown state
 - HTTP code: 400
 - HTTP text: "BAD REQUEST"
 - Error code: 0

Request example

```
GET /api/audits/list/stopped HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:



GoLismero project

```
HTTP/1.0 200 OK
```

```
Date: Thu, 17 Oct 2013 17:55:44 GMT
```

```
Server: WSGIServer/0.1 Python/2.7.2
```

```
Vary: Accept, Cookie
```

```
X-Frame-Options: SAMEORIGIN
```

```
Content-Type: application/json
```

```
Allow: GET, DELETE, HEAD, OPTIONS
```

```
{"status": "ok", "results": [{"id": "5", "state": "stopped", "name": "audit6", "user": "user4"}, {"id": "5", "state": "stopped", "name": "audit8", "user": "user10"}]}
```

Unknown state

```
HTTP/1.0 200 OK
```

```
Date: Thu, 17 Oct 2013 17:57:34 GMT
```

```
Server: WSGIServer/0.1 Python/2.7.2
```

```
Vary: Accept, Cookie
```

```
X-Frame-Options: SAMEORIGIN
```

```
Content-Type: application/json
```

```
Allow: GET, DELETE, HEAD, OPTIONS
```

```
{"status": "error", "error_code": 0, "error": ["Unknown state"]}
```



AUDIT :: CREATE method

This method creates a new audit, linked to the user who created it.

URI: `/api/audits/create/`

GET parameters

- No parameters.

POST parameters

Post parameters will be sent as JSON format:

- *targets* :: list(String) -> Target list.
- *only_vulns* :: Bool (Optional. Default: False) -> Indicates if report only contains vulns, without resources or any extra info.
- *audit_name* :: String (Optional. Default: Random name will be assigned) -> Audit name.
- *include_subdomains* :: Bool (Optional. Default: True) -> Allow to include any subdomains discovered in the analysis.
- *subdomain_regex* :: String (Optional. Default: *) -> Include to the analysis subdomains matching the regex.
- *depth* :: Number (Optional. Default: 0) -> Spider depth. 0 means infinite.
- *max_links* :: Number (Optional. Default: 0) -> Maximum number of links to audit. 0 means infinite.
- *follow_redirects* :: Bool (Optional. Default: True) -> Enable following redirects.
- *follow_first_redirect* :: Bool (Optional. Default: True) -> Enable following only first redirect.
- *proxy_addr* :: String (Optional. Default: Empty) -> String with proxy address as format: IP:PORT.
- *proxy_user* :: String (Optional. Default: Empty) -> Proxy user.
- *proxy_pass* :: String (Optional. Default: Empty) -> Proxy password.
- *cookie* :: String (Optional. Default: Empty) -> Raw text to add to Cookie HTTP header in each request.
- *user_agent* :: String (Optional. Default: random)-> Custom HTTP user agent string or "random" text.
- *disable_plugins* :: list(String) (Optional. Default: Empty list -all plugins are enabled-) -> Disabled plugins list, by plugin name.
- *enable_plugins* :: list(dict) (Optional. Default: all plugin will be enabled by default) -> Enabled plugins list, by plugin name. Also include plugins params, if are needed. Dictionary content:
 - *plugin_name* :: String -> Plugin Name to enable.
 - *params* :: list(dict) (Optional. Default: Each plugin take their default values) -> a list of tuples in format [{ 'PARAM_NAME' : 'PARAM_VALUE' }]

Response

Normal response

If the request was successfully done, the server will return the following:



GoLismero project

- HTTP code: 200
- HTTP text: OK
- *audit_id* :: String -> Audit ID.

Error responses

The server may return the following errors:

1. Plugin not found

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 0

2. Parameters error

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 1

3. Parameters missing

Some parameters are missed.

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 2

4. Not allowed target

Error raised when got not allowed target.

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 3

5. Not plugins selected

When no plugins selected, of the alias "all", raises this exception.

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 4

Request example

Post parameters are line separated for improve the readability.



GoLismero project

```
POST /api/audits/create/ HTTP/1.1
Host: golismero.mysite.com
Content-Type: application/json
Accept: */*

{
  "audit_name": "audit1",
  "targets": [ "127.0.0.1", "http://mysite.com"],
  "enabled_plugins": [
    {
      "plugin_name": "spider",
      "params": [
        {
          "param_name": "spider_param1",
          "param_value": "value 1"
        }
      ]
    }
  ]
}
```

Response example

Normal response:

```
HTTP/1.0 200 OK
Date: Mon, 04 Nov 2013 18:16:49 GMT
Server: WSGIServer/0.1 Python/2.7.5
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: POST, DELETE, OPTIONS

{"status": "ok", "audit_id": "918230191"}
```

Plugin not found



GoLismero project

HTTP/1.0 400 BAD REQUEST

Date: Mon, 04 Nov 2013 18:25:23 GMT

Server: WSGIServer/0.1 Python/2.7.5

Vary: Accept, Cookie

X-Frame-Options: SAMEORIGIN

Content-Type: application/json

Allow: POST, DELETE, OPTIONS

```
{"status": "error", "error_code": 0, "error": ["Plugin INVENT not found."]}
```

Parameter error

HTTP/1.0 400 BAD REQUEST

Date: Mon, 04 Nov 2013 18:25:23 GMT

Server: WSGIServer/0.1 Python/2.7.5

Vary: Accept, Cookie

X-Frame-Options: SAMEORIGIN

Content-Type: application/json

Allow: POST, DELETE, OPTIONS

```
{"status": "error", "error_code": 1, "error": ["Parameter 'INVENT' for plugin 'spider' is incorrect."]}
```

Parameters are missing

HTTP/1.0 400 BAD REQUEST

Date: Mon, 04 Nov 2013 18:25:23 GMT

Server: WSGIServer/0.1 Python/2.7.5

Vary: Accept, Cookie

X-Frame-Options: SAMEORIGIN

Content-Type: application/json

Allow: POST, DELETE, OPTIONS

```
{"status": "error", "error_code": 2, "error": ["Targets are missing."]}
```



AUDIT :: DELETE method

This method deletes an audit. If the caller has the administrator role, any audit can be deleted. Otherwise, if caller is a normal user, only their own audits can be deleted.

URI: `/api/audits/delete/{ID}`

GET parameters

- *ID* :: Number -> Audit ID to delete

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- *status* :: String -> "ok"

Error responses

The server may return the following errors:

1. Audit not found
 - HTTP code: 400
 - HTTP text: "BAD REQUEST"
 - Error code: 0
2. Audit are running

Audit is running. Stop first.

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 1

Request example

```
GET /api/audits/delete/7 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:



GoLismero project

```
HTTP/1.0 200 OK
Date: Fri, 18 Oct 2013 10:22:14 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "ok"}
```

Audit not found

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 0, "error": ["Provided audit ID not found"]}
```




GoLismero project

AUDIT :: START method

This method starts an audit. If the caller has the administrator role, any audit can be started. Otherwise, if the caller is a normal user, only their own audits can be started.

URI: `/api/audits/start/{ID}`

GET parameters

- *ID* :: Number -> Audit ID to delete

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- *status* :: String -> "ok"

Error responses

The server may return the following errors:

1. Audit not found

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 0

2. Audit state error

This error occurs when try to start an audit in different state that "new".

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 1

Request example

```
GET /api/audits/start/7 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:



GoLismero project

```
HTTP/1.0 200 OK
Date: Fri, 18 Oct 2013 10:22:14 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "ok"}
```

Audit not found

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 0, "error": ["Provided audit ID not found"]}
```

Audit state error

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 1, "error": ["Audit '7' is 'running'. Only new audits can be started."]}
```



GoLismero project

AUDIT :: STOP method

This method stops an audit. If the caller has the administrator role, any audit can be stopped. Otherwise, if the caller is a normal user, only their own audits can be stopped.

URI: /api/audit/stop/{ID}

GET parameters

- *ID* :: Number -> Audit ID to delete

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- *status* :: String -> "ok"

Error responses

The server may return the following errors:

1. Audit not found

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 0

2. Audit state error

This error occurs when try to start an audit in different state that "running".

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 1

Request example

```
GET /api/audits/stop/7 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:



GoLismero project

```
HTTP/1.0 200 OK
Date: Fri, 18 Oct 2013 10:22:14 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "ok"}
```

Audit not found

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 0, "error": ["Provided audit ID not found"]}
```

Audit state error

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 1, "error": ["Audit '7' is 'running'. Only running audits can be stopped."]}
```



GoLismero project

AUDIT :: STATE method

This method checks the audit state. If the caller has the administrator role, the state of any audit can be checked. Otherwise, if the caller is a normal user, only their own audits can be checked.

URI: `/api/audits/state/{ID}`

GET parameters

- `ID` :: String -> Audit ID.

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- `status` :: String -> "ok"
- `state` :: String -> Audit state. One of these available states: { running | stopped | paused | new | finished }

Error responses

The server may return the following errors:

1. Audit not found
 - HTTP code: 400
 - HTTP text: "BAD REQUEST"
 - Error code: 0

Request example

```
GET /api/audits/state/1 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:



GoLismero project

```
HTTP/1.0 200 OK
Date: Mon, 04 Nov 2013 19:31:39 GMT
Server: WSGIServer/0.1 Python/2.7.5
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "ok", "state": "new"}
```

Audit not found

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 0, "error": ["Provided audit ID not found"]}
```



AUDIT :: PROGRESS method

This method checks the audit progress. If the caller has the administrator role, the state of any audit can be checked. Otherwise, if the caller is a normal user, only their own audits can be checked.

URI: `/api/audits/progress/{ID}`

GET parameters

- *ID* :: String -> Audit ID.

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- *status* :: String -> "ok"
- *current_stage* :: String -> Audit stage at the moment of calling.
- *steps* :: Integer -> Total loops count.
- *tests_remain* :: Integer -> Total number of tests remains in the stage.
- *tests_done* :: Integer -> Total number of tests done in the stage.

Error responses

The server may return the following errors:

1. Audit not found

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 0

2. Audit state error

This error occurs when try to start an audit in different state that "running".

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 1

3. Audit finished error

This error occurs when try to get progress of a finished audit.

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 2

Request example



GoLismero project

```
GET /api/audits/progress/1 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:

```
HTTP/1.0 200 OK
Date: Mon, 04 Nov 2013 19:31:39 GMT
Server: WSGIServer/0.1 Python/2.7.5
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "ok", "tests_done": 25, "tests_remain": 4, "steps": 2, "current_stage": "recon"}
```

Audit not found

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 0, "error": ["Provided audit ID not found"]}
```

Audit state error

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 1, "error": ["Audit '7' is 'running'. Can't obtain progress from not running audits."]}
```

Audit state error



GoLismero project

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS
```

```
{"status": "error", "error_code": 2, "error": ["Provided audit is finished."]}
```



AUDIT :: RESULTS method

This method gets the audit results. If the caller has the administrator role, the results from any audit can be obtained. Otherwise, if the caller is a normal user, only their own audits results can be obtained.

By default, the report will be returned as HTML.

Getting results

URI: /api/audits/results/{ID}/{FORMAT}

GET parameters

- *ID* :: String -> Audit ID.

POST parameters

- No parameters

Response

Normal response

Normal response will return a file as format specified (see examples)

Error responses

The server may return the following errors:

1. Audit not found

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 0

2. Unknown format

Format requested is not valid.

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 1

3. Report not available

Requested report is not available.

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 2

Request example

Report by default:



GoLismero project

```
GET /api/audits/results/1 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Report specifying format:

```
GET /api/audits/results/1/xml HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response (text format):

```
HTTP/1.0 200 OK
Date: Mon, 04 Nov 2013 19:31:39 GMT
Server: WSGIServer/0.1 Python/2.7.5
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: text/plain
Allow: GET, DELETE, HEAD, OPTIONS
```

```
"\n--= Report =--\n\n-# Summary #- \n\nAudit started: 2013-11-08 13:56:11.018939\nAudit
ended: 2013-11-08 13:56:11.186552\nExecution time: 0 days, 0 hours, 0 minutes and 0 se-
conds\n\nScanned hosts: 1\nVulnerabilities: 0\n\n-# Audit Scope #- \n\n+-----+-----
-----+
\n| IP addresses | 127.0.0.1 | \n+-----+-----+-----+ \n| Web pages
| http://127.0.0.1/ | \n+-----+-----+-----+ \n\n-# Web Servers #- \n\n+-----
-+-----+-----+ \n| Base URL | http://127.0.0.1/
| \n+=====+=====+ \n| Visited URLs | http://127.0.0.1/
| \n+-----+-----+ \n\n-# Vulnerabilities #- \n\nNo vulnerabilities found.\n\n"
```

Unknown format

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS
```

```
{"status": "error", "error_code": 1, "error": ["Unknown report format 'jpg'."]}
```

Report not available



GoLismero project

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS
```

```
{"status": "error", "error_code": 2, "error": ["Not finished audit. Report is not available."]}
```



GoLismero project

Getting summarized results

URI: /api/audits/results/summary/{ID}

GET parameters

- *vulns_number* :: Number -> Total vulnerability number.
- *discovered_hosts* :: Number -> Discovered hosts during audit process. In this number are not included hosts provided as audit parameters.
- *total_hosts* :: Number -> The sum of discovered hosts and audited hosts.
- *vulns_by_level* :: Dictionary -> List of dictionaries detailing each vulnerabilities in each level.
 - *info* :: Number -> Informational vulnerabilities number.
 - *low* :: Number -> Lower vulnerabilities number.
 - *medium* :: Number -> Informational vulnerabilities number.
 - *high* :: Number -> Informational vulnerabilities number.
 - *critical* :: Number -> Informational vulnerabilities number.

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- *status* :: String -> "ok"

Error responses

The server may return the following errors:

4. Audit not found
 - HTTP code: 400
 - HTTP text: "BAD REQUEST"
 - Error code: 0

Request example

```
GET /api/audits/results/summary/7 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:



GoLismero project

```
HTTP/1.0 200 OK
```

```
Date: Fri, 18 Oct 2013 13:40:25 GMT
```

```
Server: WSGIServer/0.1 Python/2.7.2
```

```
Vary: Accept, Cookie
```

```
X-Frame-Options: SAMEORIGIN
```

```
Content-Type: application/json
```

```
Allow: GET, DELETE, HEAD, OPTIONS
```

```
{"status": "ok", "vulns_number": "12", "vulns_by_level": {"info": "5", "high": "2", "medium": "2",  
"critical": "1", "low": "2"}, "discovered_hosts": "3", "total_hosts": "4"}
```

Audit not found

```
HTTP/1.0 400 BAD REQUEST
```

```
Date: Fri, 18 Oct 2013 10:29:57 GMT
```

```
Server: WSGIServer/0.1 Python/2.7.2
```

```
Vary: Accept, Cookie
```

```
X-Frame-Options: SAMEORIGIN
```

```
Content-Type: application/json
```

```
Allow: GET, DELETE, HEAD, OPTIONS
```

```
{"status": "error", "error_code": 0, "error": ["Provided audit ID not found"]}
```



GoLismero project

AUDIT :: DETAILS method

This method gets the audit details. If the caller has the administrator role, any audit details can be retrieved. Otherwise, if the caller is a normal user, only their own audits details can be retrieved.

URI: `/api/audits/details/{ID}`

GET parameters

- *ID* :: Number -> Audit ID to delete

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP CODE: 200
- HTTP TEXT: OK
- Params:
 - *start_date* :: Date -> Audit start date and time.
 - *end_date* :: Date -> Audit start date and time.
 - *user* :: String -> Audit owner.
 - *config* :: Dictionary ->
 - *targets* :: list(String) -> Target list.
 - *only_vulns* :: Bool -> Indicates if report only contains vulns, without resources or any extra info.
 - *audit_name* :: String -> Audit name.
 - *imports* :: list(String) -> imported files list.
 - *enable_plugins* :: list(String) -> Enabled plugins list, by plugin name.
 - *disable_plugins* :: list(String) -> Disabled plugins list, by plugin name.
 - *include_subdomains* :: Bool -> Allow to include any subdomains discovered in the analysis.
 - *subdomain_regex* :: String -> Include to the analysis subdomains matching the regex.
 - *depth* :: Number -> Spider depth. 0 means infinite.
 - *max_links* :: Number -> Maximum number of links to audit. 0 means infinite.
 - *follow_redirects* :: Bool -> Enable following redirects.
 - *follow_first_redirect* :: Bool -> Enable following only first redirect.
 - *proxy_addr* :: String -> String with proxy address as format: IP:PORT.
 - *proxy_user* :: String -> Proxy user.
 - *proxy_pass* :: String -> Proxy password.
 - *cookie* :: String -> Raw text to add to Cookie HTTP header in each request.
 - *user_agent* :: String -> Custom HTTP user agent string or "random" text.



Error responses

The server may return the following errors:

5. Audit not found

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 0

Request example

```
GET /api/audits/details/7 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:

```
HTTP/1.0 200 OK
Date: Fri, 18 Oct 2013 14:10:56 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "ok", "config": {"subdomain_regex": "", "proxy_user": "", "proxy_pass": "", "imports":
[""], "follow_first_redirect": "True", "depth": "0", "enable_plugins": ["spider", "sqlmap", "openvas"],
"max_links": "0", "disable_plugins": [""], "proxy_addr": "", "only_vulns": "False", "follow_redirects":
"True", "cookie": "", "audit_name": "audit1", "include_subdomains": "True", "targets":
["www.target1.com", "target2.com"], "user_agent": "random"}, "start_date": "2013-10-18
16:03:32", "end_date": "2013-10-14 10:23:32", "user": "user1"}
```

Incorrect credentials

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 0, "error": ["Provided audit ID not found"]}
```




GoLismero project

AUDIT :: PAUSE method

This method pauses an audit. If the caller has the administrator role, any audit can be paused. Otherwise, if the caller is a normal user, only their own audits can be paused.

URI: `/api/audits/pause/{ID}`

GET parameters

- *ID* :: Number -> Audit ID to delete

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- *status* :: String -> "ok"

Error responses

The server may return the following errors:

1. Audit not found

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 0

2. Audit state error

This error occurs when try to pause an audit in different state that "running".

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 1

Request example

```
GET /api/audits/start/7 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:



GoLismero project

```
HTTP/1.0 200 OK
Date: Fri, 18 Oct 2013 10:22:14 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "ok"}
```

Audit not found

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 0, "error": ["Provided audit ID not found"]}
```

Audit state error

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 1, "error": ["Audit '7' is 'running'. Only running audits can be stopped."]}
```



GoLismero project

AUDIT :: RESUME method

This method resumes a paused audit. If the caller has the administrator role, any audit can be resumed. Otherwise, if the caller is a normal user, only their own audits can be resumed.

URI: /api/audits/resume/{ID}

GET parameters

- *ID* :: Number -> Audit ID to delete

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- *status* :: String -> "ok"

Error responses

The server may return the following errors:

1. Audit not found
 - HTTP code: 400
 - HTTP text: "BAD REQUEST"
 - Error code: 0

Request example

```
GET /api/audits/resume/7 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:



GoLismero project

```
HTTP/1.0 200 OK
Date: Fri, 18 Oct 2013 10:22:14 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "ok"}
```

Audit not found

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 0, "error": ["Provided audit ID not found"]}
```

Audit state error

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 1, "error": ["Audit '7' is 'running'. Only paused audits can be re-summed."]}
```



GoLismero project

AUDIT :: LOG method

This method return audit generated log. If the caller has the administrator role, all audits logs will be returned. Otherwise, if the caller is a normal user, only their own audits logs will be returned.

It return all related log generated by the audit. This method can be called in any moment. If audit is not finished, it returns available audit log for the moment of call.

URI: `/api/audits/log/{ID}`

GET parameters

- `ID` :: Number -> Audit ID to delete

POST parameters

- No parameters

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- `status` :: String -> "ok"
- `log` :: String -> String with log information. The String could has a huge size.

Error responses

The server may return the following errors:

1. Audit not found
 - HTTP code: 400
 - HTTP text: "BAD REQUEST"
 - Error code: 0
2. Audit state error

This error occurs when try to get log from an audit in state "new".

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 1

Request example

```
GET /api/audits/log/7 HTTP/1.1
Host: golismero.mysite.com
Accept: */*
```

Response example

Normal response:



GoLismero project

```
GET /api/audits/start/7 HTTP/1.1
```

```
Host: golismero.mysite.com
```

```
Accept: */*
```

```
{"status": "ok", "log": "[2013-11-02 12:06:32:1383390392] Added 4 new targets to the data-
base.[2013-11-02 12:06:32:1383390392] Audit scope:\n\n\t\t IP addresses:\n\t\t
208.84.244.10\n\n\t\t Domains:\n\t\t\t *.terra.es\n\t\t\t terra.es\n\t\t\t
www.terra.es\n\n\t\t Web pages:\n\t\t\t http://www.terra.es/\n\t\t\t [2013-11-02
12:06:33:1383390393] Spidering URL: \"http://www.terra.es/\""}}
```

Audit not found

```
HTTP/1.0 400 BAD REQUEST
```

```
Date: Fri, 18 Oct 2013 10:29:57 GMT
```

```
Server: WSGIServer/0.1 Python/2.7.2
```

```
Vary: Accept, Cookie
```

```
X-Frame-Options: SAMEORIGIN
```

```
Content-Type: application/json
```

```
Allow: GET, DELETE, HEAD, OPTIONS
```

```
{"status": "error", "error_code": 0, "error": ["Provided audit ID not found"]}
```

Audit state error

```
HTTP/1.0 400 BAD REQUEST
```

```
Date: Fri, 18 Oct 2013 10:29:57 GMT
```

```
Server: WSGIServer/0.1 Python/2.7.2
```

```
Vary: Accept, Cookie
```

```
X-Frame-Options: SAMEORIGIN
```

```
Content-Type: application/json
```

```
Allow: GET, DELETE, HEAD, OPTIONS
```

```
{"status": "error", "error_code": 1, "error": ["Audit '7' is not 'running'. Can't obtain log for non start-
ed audits."]}
```



AUDIT :: IMPORT method

This method allow to create a special type of audit called "import". This type of audit not does any test, only import results from different tools.

It creates a new audit, with all of the related elements, and results files.

IMPORTANT: Currently this method only can import files from local file system.

URI: /api/audits/import/

GET parameters

- No parameters

POST parameters

- *audit_name* :: String (Optional. Default: Random name will be assigned) -> Audit name.
- *enable_plugins* :: list(dict) (Optional. Default: all plugin will be enabled by default) -> Enabled plugins list, by plugin name. Also include plugins params, if are needed. Dictionary content:
 - *plugin_name* :: String -> Plugin Name to enable.
 - *params* :: list(dict) (Optional. Default: Each plugin take their default values) -> a list of tuples in format [{ 'PARAM_NAME' : 'PARAM_VALUE' }]
- *imports* :: list(String) -> List with files path to import.

Response

Normal response

If the request was successfully done, the server will return the following:

- HTTP code: 200
- HTTP text: OK
- *status* :: String -> "ok"

Error responses

The server may return the following errors:

1. Parameter error
 - HTTP code: 400
 - HTTP text: "BAD REQUEST"
 - Error code: 1
2. Error in import files

Some specified files not exists or is not a file.

- HTTP code: 400
- HTTP text: "BAD REQUEST"
- Error code: 2



GoLismero project

Request example

```
GET /api/audits/import/ HTTP/1.1
Host: golismero.mysite.com
Accept: */*

{
  "audit_name": "audit1",
  "imports": [ "/home/user/myfile.xml", "/home/user/other_folder/myfile2.xml" ]
}
```

Response example

Normal response:

```
GET /api/audits/start/7 HTTP/1.1
Host: golismero.mysite.com
Content-Type: application/json
Accept: */*

{"status": "ok" }
```

Parameter error

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 1, "error": ["Field 'imports' is required"]}
```

Error in import files

```
HTTP/1.0 400 BAD REQUEST
Date: Fri, 18 Oct 2013 10:29:57 GMT
Server: WSGIServer/0.1 Python/2.7.2
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Allow: GET, DELETE, HEAD, OPTIONS

{"status": "error", "error_code": 2, "error": ["File '/home/user/myfile.xml' not exists."]}
```




GoLismero project

4. User management

These are the URIs defined for user management.

Base URI: /api/users/

These are the methods defined for user management:

- list
- create
- edit
- delete
- details

User restrictions and roles

The following methods can only be called by users with the administrator role:

- list
- create
- delete

Non-privileged users can only manipulate with their own information with the following methods, while users with the administrator role can manipulate information from any user with them:

- edit
- details



USERS :: LIST method

This method shows all available GoLismero users. Only users with the administrator role can call it.

URI: /api/users/list/

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

USERS :: CREATE method

This method creates a new user. Only users with the administrator role can call it.

URI: /api/users/create/

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

USERS :: EDIT method

This method modifies the information for an existing user. Only users with the administrator role can call it on any user ID – non-privileged users may only access their own information.

URI: /api/users/edit/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

USERS :: DELETE method

This method deletes an existing user. Only users with the administrator role can call it.

URI: /api/users/delete/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

USERS :: DETAILS method

This method retrieves the information for an existing user. Only users with the administrator role can call it on any user ID – non-privileged users may only access their own information.

URI: /api/users/details/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

5. Plugin management

These are the URIs defined for plugin management.

Base URI: /api/plugins/

These are the methods defined for plugin management:

- list
- details
- search



GoLismero project

PLUGINS :: LIST method

This method shows all available GoLismero plugins, their parameters and execution stage.

URI: /api/plugins/list/

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Getting all available plugins:

```
GET /api/plugins/ HTTP/1.1  
Host: golismero.myhost.com
```

Response example

Normal response:

Incorrect credentials



GoLismero project

PLUGINS :: DETAILS method

This method shows plugins metadata: author, description, version, parameters, configuration, etc.

URI: /api/plugins/details/{NAME}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Obtaining all detailed information about the “spider” plugin:

```
GET /api/plugins/spider HTTP/1.1
```

```
Host: golismero.myhost.com
```

Response example

Normal response:

Incorrect credentials



GoLismero project

PLUGINS :: SEARCH method

This method provides a search engine for plugins.

URI: /api/plugins/search/{TEXT}

GET parameters

- TEXT :: String -> Text to search within the available plugins. The text may contain wildcards.

POST parameters

- No parameters

Response

Request example

Normal search

```
GET /api/plugins/search/spider HTTP/1.1
Host: golismero.myhost.com
```

Search using wildcards

```
GET /api/plugins/search/spider* HTTP/1.1
Host: golismero.myhost.com
```

Response example

Normal response:

Incorrect credentials



GoLismero project

6. Profile management

These are the URLs defined for profile management.

Base URI: /api/profiles/

These are the methods defined for profile management:

- list
- create
- edit
- delete
- details

About profiles

There are two kinds of profiles:

- Global profiles
- User profiles

Global profiles

Any user can access this type of profiles, but only a user with the administrator role can modify them.

User profiles

All users can add/modify/delete their own profiles. Users with the administrator role can access all user profiles.



GoLismero project

PROFILES :: LIST method

This method lists profiles. If the caller has the administrator role, all profiles will be returned. Otherwise, if the caller is a normal user, only their own profiles will be returned.

URI: /api/profiles/list/

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

PROFILES :: CREATE method

This method creates a new profile. If the caller has the administrator role, it can choose if the profile is global or not. Otherwise, if the caller is a normal user, the profile will only be available to the caller.

URI: /api/profiles/create/

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

PROFILES :: EDIT method

This method allows editing profiles. If the caller has the administrator role, any profile can be edited. Normal users can only edit their own profiles.

URI: /api/profiles/edit/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

PROFILES :: DELETE method

This method allows profile deletion. If the caller has the administrator role, any profile can be deleted. Normal users can only delete their own profiles.

URI: /api/profiles/delete/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

PROFILES :: DETAILS method

This method shows profiles details. If the caller has the administrator role, any profile information can be retrieved. Normal users can only retrieve information for their own profiles.

URI: /api/profiles/details/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

7. Node administration

These are the URIs defined for node administration.

Base URI: /api/nodes/

These are the methods defined for node administration:

- Node administration
 - list
 - start
 - stop
 - restart
 - state
 - Configuration edit
 - Configuration details

About nodes

A node is a running instance of GoLismero and can be managed like a common remote service.

Only users with the administrator role can use this API.

Important notes

Currently, GoLismero not support nodes management. It only runs a single instance of GoLismero.

THIS SECTION OF THE API IS NOT IMPLEMENTED YET



NODE :: LIST method

This method lists all available nodes that GoLismero can manage.

URI: /api/nodes/list/

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

NODE :: START method

This method starts a node.

URI: /api/nodes/start/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



NODE :: STOP method

This method stops a node.

URI: /api/nodes/stop/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

NODE :: RESTART method

This method restarts a node.

URI: /api/nodes/restart/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



NODE :: STATE method

This method shows the state of a node.

URI: /api/nodes/state/{ID}/

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

NODE :: CONFIG DETAILS method

This method shows the configuration of a node.

URI: /api/nodes/config/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials



GoLismero project

NODE :: CONFIG EDIT method

This method allows editing the configuration of a node.

URI: /api/nodes/config/edit/{ID}

GET parameters

- No parameters

POST parameters

- No parameters

Response

Request example

Response example

Normal response:

Incorrect credentials