
gs.auth.token Documentation

Release 2.1.1

GroupServer.org

April 28, 2016

1	<code>gs_auth_token_create</code>	3
1.1	Synopsis	3
1.2	Description	3
1.3	Positional arguments	3
1.4	Optional arguments	3
1.5	Returns	3
1.6	Authentication Configuration	4
1.7	Examples	4
2	<code>gs_auth.token</code> API	5
2.1	The interface	5
2.2	The form	6
3	Changelog	7
4	2.1.1 (2015-06-18)	9
5	2.1.0 (2014-06-23)	11
6	2.0.2 (2013-08-09)	13
7	2.0.1 (2012-09-02)	15
8	2.0.0 (2012-07-26)	17
9	1.0.0 (2012-06-18)	19
10	Indices and tables	21
11	Resources	23

Author Michael JasonSmith

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-06-18

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

This product provides the token-authentication system used by *web hooks* in [GroupServer](#). It provides both the script that generates new tokens, and the API used to check supplied tokens.

Contents:

gs_auth_token_create

1.1 Synopsis

```
gs_auth_token_create [-h] dsn
```

1.2 Description

gs_auth_token_create creates a new token and adds it to the relational database. Once run it is necessary to change the *authentication configuration*.

1.3 Positional arguments

dsn

The data source name (DSN) in the form `postgres://<user>:<password>@<host>:<port>/<database_name>`. The configuration file for GroupServer (normally `etc/gsconfig.ini`) lists all the DSN entries for the system.

1.4 Optional arguments

-h, --help

Show a help message and exit

1.5 Returns

- **gs_auth_token_create** returns 0 on success and the new token is displayed on the standard output.
- 1 is returned if the system failed to make the initial connection to the database (the *engine* failed to be created).
- 2 is returned if the system failed to connect for any other reason.

1.6 Authentication Configuration

The scripts that use the web-hooks, such as **smtp2gs** ¹, use the configuration file for storing the token. **gs_auth_token_create** leaves these configuration files **unaltered**, because they can become complex. Instead it displays the new token and the administrator must change the entries in the configuration.

1.7 Examples

Generate a new token and place it in the `production` database. In this example PostgreSQL is running on the default port of the local machine — and has as been set up so authentication is unnecessary:

```
$ gs_auth_token_create postgres://localhost/production
```

Generate a new token and place it in the `testing` database on a *remote* machine `groups.example.com` — with the port 5432 explicitly passed. Authentication is used.

```
$ gs_auth_token_create \  
postgres://databaseUser:secretPass@groups.example.com:5432/testing
```

¹ See `gs.group.messages.add.smtp2gs` <<https://github.com/groupserver/gsgroup.messages.add.smtp2gs>>

gs.auth.token API

Using token authentication is normally done in two steps. First the token is added to *the interface*, and then used in *the form*.

2.1 The interface

The interface defines the parameters that the web hook accepts. To use token authentication one of the parameters should be an instance of the *AuthToken* class.

class AuthToken (*title, token, required*)

An authentication token field.

Parameters

- **title** (*str*) – The title (almost always Token)
- **description** (*str*) – The description of the field.
- **required** (*bool*) – Weather the field is required (almost always True)

Raises *AuthenticationTokenMismatch* – There was a miss-match between the supplied token and the stored token.

Web-hooks that want to use token authentication include a *AuthToken* attribute as a parameter.

class AuthenticationTokenMismatch

The supplied token failed to match the token stored in the database.

2.1.1 Example

In the following example the *ISomeHook* interface class is created with the *token* property set to be an instance of the *AuthToken* class.

```
from __future__ import unicode_literals
from zope.interface.interface import Interface
from gs.auth.token import AuthToken

class ISomeHook(Interface):

    token = AuthToken(
        title='Token',
        description='The authentication token',
        required=True)
```

2.2 The form

A form that actually supplies the web-hook uses *the interface*, and handles any errors using the `log_auth_error()` function.

log_auth_error (*context, request, errors*)

Log a token authentication error.

Parameters

- **context** – The context of the current page (hook).
- **request** – The current request.
- **errors** – The errors that have occurred.

This utility will check for the `AuthenticationTokenMismatch` error in the list of `errors`. If present it will add an audit-event to the audit-trail table.

2.2.1 Example

Typically the form that provides a web-hook is a subclass of the JSON `SiteEndpoint` class². If there is an error in the form then the utility `gs.auth.token.log_auth_error()` should be called.

```
from zope.formlib import form
from gs.auth.token import log_auth_error
from gs.content.form.api.json import SiteEndpoint
from .interfaces import ISomeHook

class SomeHook(SiteEndpoint):
    '''The hook'''
    label = 'Some hook'
    form_fields = form.Fields(ISomeHook, render_context=False)

    @form.action(label='Some', name='some', prefix='',
                 failure='handle_some_failure')
    def handle_some(self, action, data):
        '''Do something

:param action: The button that was clicked.
:param dict data: The form data.'''

    def handle_some_failure(self, action, data, errors):
        log_auth_error(self.context, self.request, errors)
        retval = self.build_error_response(action, data, errors)
        return retval
```

² See the `gs.content.form.api.json` product <<https://github.com/groupserver/gs.content.form.api.json>>

Changelog

2.1.1 (2015-06-18)

- Updating the documentation

2.1.0 (2014-06-23)

- Adding unit tests
- Switching to Unicode literals
- Adding support for Python 3 and PyPy

2.0.2 (2013-08-09)

- Updating the product metadata

2.0.1 (2012-09-02)

- Fixing an error with `munge_date`

2.0.0 (2012-07-26)

- Switching from many options to a single DSN parameter
- Update to SQLAlchemy

1.0.0 (2012-06-18)

Initial version. Prior to the creation of this product the authentication of web hooks was provided by the ZMI.

Indices and tables

- `genindex`
- `modindex`
- `search`

Resources

- Documentation: <http://groupserver.readthedocs.io/projects/gsauthtoken>
- Code repository: <https://github.com/groupserver/gs.auth.token/>
- Questions and comments to <http://groupserver.org/groups/development>
- Report bugs at <https://redmine.iopen.net/projects/groupserver>

Symbols

-h, --help

gs_auth_token_create command line option, 3

A

AuthenticationTokenMismatch (built-in class), 5

AuthToken (built-in class), 5

D

dsn

gs_auth_token_create command line option, 3

G

gs_auth_token_create command line option

-h, --help, 3

dsn, 3

L

log_auth_error() (built-in function), 6