ACME Mangement Server

Release 0.3.1

Oct 20, 2018

Contents

1 Develop on ACMEMS **3**
- 1.1 Manager . . . 3
- 1.2 Server . . . 3
- 1.3 Authentication & Processing . . . 3
- 1.4 Configuration . . . 4
- 1.5 Exceptions . . . 5

2 ChangeLog **7**
- 2.1 Version 0 . . . 7

3 Indices and tables **9**

Python Module Index **11**

Contents:

CHAPTER 1

Develop on ACMEMS

1.1 Manager

1.2 Server

class acmems.server.**ACMEAbstractHandler**(*request*, *client_address*, *server*)
Bases: http.server.BaseHTTPRequestHandler

send_data(*data*, *content_type='text/plain'*, *response_code=200*)
Helper method to send data as HTTP response. The data are transfered as *text/plain*.

Parameters

- **data** (*str*) – The text to send as Python String.
- **response_code** (*int*) – HTTP response code

class acmems.server.**ACMEHTTPHandler**(*validator*, **args*, ***kwargs*)
Bases: *acmems.server.ACMEAbstractHandler*

do_GET()
Handles POST request (upload files).

class acmems.server.**ACMEMgmtHandler**(*request*, *client_address*, *server*)
Bases: *acmems.server.ACMEAbstractHandler*

do_POST()
Handles POST request (upload files).

1.3 Authentication & Processing

class acmems.auth.**SubjectAltName**(**args*, ***kwargs*)
Bases: ndg.httpsclient.subj_alt_name.SubjectAltName

ASN.1 implementation for subjectAltNames support

class acmems.auth.**IPAuthMethod**(*ips=None*)
Bases: object

Autentication by source IP

class acmems.auth.**HmacAuthMethod**
Bases: object

Authentication by HMAC / secret key

class acmems.auth.**AllAuthMethod**
Bases: object

Allow all authentication

class acmems.auth.**Block**(*name*, *options*, *config*)
Bases: object

One authentication block - combination of authentications and list of allowed domains

class acmems.auth.**Processor**(*auth*, *client_address*, *headers*, *rfile*)
Bases: object

Helper object to process a request, check authentication, reads and parse CSR

acceptable()
process the given request parameter for a CSR signing request and decide whether this request is allowed or not.

Parameters

- **str** (*client_ip*) – The source IP of the client (TCP level)
- **headers** (*dict*) – The request header
- **get_body** (*callable*) – function to read in body (CSR)

Return bool whether request should be accepted

1.4 Configuration

exception acmems.config.**ConfigurationError**
Bases: Exception

exception acmems.config.**MissingSectionError**
Bases: *acmems.config.ConfigurationError*

exception acmems.config.**UnknownVerificationError**
Bases: *acmems.config.ConfigurationError*

exception acmems.config.**UnknownStorageError**
Bases: *acmems.config.ConfigurationError*

exception acmems.config.**SingletonOptionRedifined**(*section*, *option*, *old*, *new*)
Bases: *acmems.config.ConfigurationError*

exception acmems.config.**ConfigurationWarning**
Bases: UserWarning

exception acmems.config.**UnusedOptionWarning**
Bases: *acmems.config.ConfigurationWarning*

exception acmems.config.**OptionRedifinitionWarning**
Bases: *acmems.config.ConfigurationWarning*

exception acmems.config.**UnusedSectionWarning**
Bases: *acmems.config.ConfigurationWarning*

1.5 Exceptions

exception acmems.exceptions.**AcmeException**
Bases: Exception

Base exception call to be able to catch all ACMEMS specific errors

exception acmems.exceptions.**NoChallengeMethodsSupported**
Bases: *acmems.exceptions.AcmeException*

The domain can not be validated HTTP01

exception acmems.exceptions.**ChallengeFailed**(*domain*, *message*, *challenge_uri*)
Bases: *acmems.exceptions.AcmeException*

The challenge to validate the requested domain failed.

Variables

- **domain** (*str*) – the domain which the challenge should validate
- **message** (*str*) – message description from ACME server
- **challenge_uri** (*str*) – the URI of the failed challenge

exception acmems.exceptions.**ChallengesUnknownStatus**
Bases: *acmems.exceptions.AcmeException*

We do not known the status of the challenge. No clue what to do

exception acmems.exceptions.**AuthorizationNotYetProcessed**(*wait_until*)
Bases: *acmems.exceptions.AcmeException*

The authorization is be processed; until the next refresh it should at least be wait until `wait_until`

Variables **wait_until** (*datetime.datetime*) – first allowed retry time

exception acmems.exceptions.**AuthorizationNotYetRequested**(*event*)
Bases: *acmems.exceptions.AcmeException*

The newly created authorization challenge, was installed, but has not yet been requested by any client and is therefore currently pending or invalid.

Variables **event** (*threading.Event*) – event that will be signaled if someone requests the challenge.

exception acmems.exceptions.**RateLimited**
Bases: *acmems.exceptions.AcmeException*

To many requests

exception acmems.exceptions.**AccountError**
Bases: *acmems.exceptions.AcmeException*

Generic account error - e.g. - could not read private key - could not refresh the registration

exception acmems.exceptions.**NeedToAgreeToTOS**(*url*)
Bases: *acmems.exceptions.AccountError*

We are registered at the ACME server. But to use it, we need to accept the "Terms of Service"

exception acmems.exceptions.**InvalidDomainName**(*domain*, *detail*)
Bases: *acmems.exceptions.AcmeException*

The domain name is not excepted by the ACME server.

Variables

- **domain** (*str*) – the domain that was rejected
- **detail** (*str*) – the reject reason as string

exception acmems.exceptions.**PayloadToLarge**(*size*, *allowed*)
Bases: *acmems.exceptions.AcmeException*

The payload (CSR) it to large

Variables

- **size** (*int*) – the request size to upload (in bytes)
- **allowed** (*int*) – the maximal size in bytes

exception acmems.exceptions.**PayloadInvalid**
Bases: *acmems.exceptions.AcmeException*

The payload is not a valid CSR

CHAPTER 2

ChangeLog

This page lists all versions with its changes. ACMEMS follows Semantic Versioning.

2.1 Version 0

2.1.1 v0.3.1

Multiple bug fixes:

- Fix auth-block specific storage and verification settings
- IOError when replace certification in file storage
- Fix typos in dns01-dnsUpdate verification

2.1.2 v0.3.0

(Experimental) support for DNS challenges

2.1.3 v0.2.0

Reaching base architecture for 1.0 release. This includes:

- Restucture code and! *config* to support multiple verification mechanism
- WIP: experiment / prepare for dns01 challenge support (via dns updates)
- add storage support to not reissue CSRs the same pem, supporting reissue from multiple machines via a once shared key and CSR
- support newer python-acme releases

2.1.4 v0.1.1

- Fix syntax error in setup.py, preventing to upload to PyPI

2.1.5 v0.1.0

Implement basic feature set:

- submit CSR
- validate domain via HTTP
- sign certificate
- authenticate clients based on IP and HMAC

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

acmems.auth, 3
acmems.config, 4
acmems.exceptions, 5
acmems.manager, 3
acmems.server, 3

Index

A

acceptable() (acmems.auth.Processor method), 4
AccountError, 5
ACMEAbstractHandler (class in acmems.server), 3
AcmeException, 5
ACMEHTTPHandler (class in acmems.server), 3
ACMEMgmtHandler (class in acmems.server), 3
acmems.auth (module), 3
acmems.config (module), 4
acmems.exceptions (module), 5
acmems.manager (module), 3
acmems.server (module), 3
AllAuthMethod (class in acmems.auth), 4
AuthorizationNotYetProcessed, 5
AuthorizationNotYetRequested, 5

B

Block (class in acmems.auth), 4

C

ChallengeFailed, 5
ChallengesUnknownStatus, 5
ConfigurationError, 4
ConfigurationWarning, 4

D

do_GET() (acmems.server.ACMEHTTPHandler method), 3
do_POST() (acmems.server.ACMEMgmtHandler method), 3

H

HmacAuthMethod (class in acmems.auth), 4

I

InvalidDomainName, 6
IPAuthMethod (class in acmems.auth), 4

M

MissingSectionError, 4

N

NeedToAgreeToTOS, 5
NoChallengeMethodsSupported, 5

O

OptionRedifinitionWarning, 4

P

PayloadInvalid, 6
PayloadToLarge, 6
Processor (class in acmems.auth), 4

R

RateLimited, 5

S

send_data() (acmems.server.ACMEAbstractHandler method), 3
SingletonOptionRedifined, 4
SubjectAltName (class in acmems.auth), 3

U

UnknownStorageError, 4
UnknownVerificationError, 4
UnusedOptionWarning, 4
UnusedSectionWarning, 5