

openstack / keystone Public

<> Code Pull requests 1 Actions Security and quality Insights

⚠ This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

Commit 82c87e5



Jenkins authored and ~~openstack-gerrit~~ committed on Feb 5, 2013

Merge "Add size validations for /tokens." into stable/folsom

· folsom-eol 2012.2.4

2 parents [b3bd5fd](#) + [bb2226f](#) commit 82c87e5

4 files changed +118 -0 lines changed

↑ Top ⚙

Filter files...

- keystone
 - config.py
 - exception.py
 - service.py
- tests
 - test_service.py

4 files changed +118 -0 lines changed

Search within code ⚙

keystone/config.py

```

@@ -117,6 +117,9 @@ def register_cli_int(*args, **kw):
117 117     register_str('public_port', default=5000)
118 118     register_str('onready')
119 119     register_str('auth_admin_prefix', default='')

```

```

120 + register_int('max_param_size', default=64)
121 + # we allow tokens to be a bit larger to accomidate PKI
122 + register_int('max_token_size', default=8192)
120 123
121 124     #ssl options
122 125     register_bool('enable', group='ssl', default=False)

```



▼ keystone/exception.py



```

↑... @@ -51,6 +51,19 @@ class ValidationError(Error):
51 51         title = 'Bad Request'
52 52
53 53
54 + class ValidationSizeError(Error):
55 +     """Request attribute %(attribute)s must be less than or equal to %(size)i.
56 +
57 +     The server could not comply with the request because the attribute
58 +     size is invalid (too large).
59 +
60 +     The client is assumed to be in error.
61 +
62 +     """
63 +     code = 400
64 +     title = 'Bad Request'
65 +
66 +
54 67     class Unauthorized(Error):
55 68         """The request you have made requires authentication."""
56 69         code = 401

```



▼ keystone/service.py



```

↑... @@ -22,6 +22,7 @@
22 22     from keystone import catalog
23 23     from keystone.common import cms
24 24     from keystone.common import logging
25 + from keystone.common import utils
25 26     from keystone.common import wsgi
26 27     from keystone import exception

```

```

27 28     from keystone import identity
@@ -31,6 +32,8 @@
31 32
32 33
33 34     LOG = logging.getLogger(__name__)
35 + MAX_PARAM_SIZE = config.CONF.max_param_size
36 + MAX_TOKEN_SIZE = config.CONF.max_token_size
34 37
35 38
36 39     class AdminRouter(wsgi.ComposingRouter):
@@ -288,9 +291,23 @@ def authenticate(self, context, auth=None):
288 291
289 292         if 'passwordCredentials' in auth:
290 293             user_id = auth['passwordCredentials'].get('userId', None)
294 +             if user_id and len(user_id) > MAX_PARAM_SIZE:
295 +                 raise exception.ValidationError(attribute='userId',
296 +                                                     size=MAX_PARAM_SIZE)
291 297             username = auth['passwordCredentials'].get('username', '')
298 +             if len(username) > MAX_PARAM_SIZE:
299 +                 raise exception.ValidationError(attribute='username',
300 +                                                     size=MAX_PARAM_SIZE)
292 301             password = auth['passwordCredentials'].get('password', '')
302 +             max_pw_size = utils.MAX_PASSWORD_LENGTH
303 +             if len(password) > max_pw_size:
304 +                 raise exception.ValidationError(attribute='password',
305 +                                                     size=max_pw_size)
306 +
293 307             tenant_name = auth.get('tenantName', None)
308 +             if tenant_name and len(tenant_name) > MAX_PARAM_SIZE:
309 +                 raise exception.ValidationError(attribute='tenantName',
310 +                                                     size=MAX_PARAM_SIZE)
294 311
295 312             if username:
296 313                 try:
@@ -302,6 +319,9 @@ def authenticate(self, context, auth=None):
302 319
303 320                 # more compat
304 321                 tenant_id = auth.get('tenantId', None)
322 +             if tenant_id and len(tenant_id) > MAX_PARAM_SIZE:

```

```

323 +         raise exception.ValidationSizeError(attribute='tenantId',
324 +                                             size=MAX_PARAM_SIZE)
305 325         if tenant_name:
306 326             try:
307 327                 tenant_ref = self.identity_api.get_tenant_by_name(
@@ -342,7 +362,14 @@ def authenticate(self, context, auth=None):
342 362                 catalog_ref = {}
343 363             elif 'token' in auth:
344 364                 old_token = auth['token'].get('id', None)
365 +
366 +                 if len(old_token) > MAX_TOKEN_SIZE:
367 +                     raise exception.ValidationSizeError(attribute='token',
368 +                                                         size=MAX_TOKEN_SIZE)
345 369                 tenant_name = auth.get('tenantName')
370 +                 if tenant_name and len(tenant_name) > MAX_PARAM_SIZE:
371 +                     raise exception.ValidationSizeError(attribute='tenantName',
372 +                                                         size=MAX_PARAM_SIZE)
346 373
347 374             try:
348 375                 old_token_ref = self.token_api.get_token(context=context,

```

```

tests/test_service.py
@@ -17,6 +17,7 @@
17 17     import default_fixtures
18 18
19 19     from keystone import config
20 + from keystone import exception
20 21     from keystone import service
21 22     from keystone import test
22 23     from keystone.identity.backends import kvs as kvs_identity
@@ -25,6 +26,31 @@
25 26     CONF = config.CONF
26 27
27 28
29 + def _build_user_auth(token=None, user_id=None, username=None,
30 +                       password=None, tenant_id=None, tenant_name=None):
31 +     """Build auth dictionary.
32 +

```

```
33 + It will create an auth dictionary based on all the arguments
34 + that it receives.
35 + """
36 + auth_json = {}
37 + if token is not None:
38 +     auth_json['token'] = token
39 + if username or password:
40 +     auth_json['passwordCredentials'] = {}
41 + if username is not None:
42 +     auth_json['passwordCredentials']['username'] = username
43 + if user_id is not None:
44 +     auth_json['passwordCredentials']['userId'] = user_id
45 + if password is not None:
46 +     auth_json['passwordCredentials']['password'] = password
47 + if tenant_name is not None:
48 +     auth_json['tenantName'] = tenant_name
49 + if tenant_id is not None:
50 +     auth_json['tenantId'] = tenant_id
51 + return auth_json
52 +
53 +
```

```
28 54 class TokenExpirationTest(test.TestCase):
29 55     def setUp(self):
30 56         super(TokenExpirationTest, self).setUp()
```



```
@@ -75,3 +101,52 @@ def _maintain_token_expiration(self):
```

```
75 101     def test_maintain_uuid_token_expiration(self):
76 102         self.opt_in_group('signing', token_format='UUID')
77 103         self._maintain_token_expiration()
```

```
104 +
105 +
106 + class AuthTest(test.TestCase):
107 +     def setUp(self):
108 +         super(AuthTest, self).setUp()
109 +
110 +         CONF.identity.driver = 'keystone.identity.backends.kvs.Identity'
111 +         self.load_backends()
112 +         self.load_fixtures(default_fixtures)
113 +         self.api = service.TokenController()
114 +
```

```
115 + def test_authenticate_user_id_too_large(self):
116 +     """Verify sending large 'userId' raises the right exception."""
117 +     body_dict = _build_user_auth(user_id='0' * 65, username='F00',
118 +                                 password='foo2')
119 +     self.assertRaises(exception.ValidationError, self.api.authenticate,
120 +                      {}, body_dict)
121 +
122 + def test_authenticate_username_too_large(self):
123 +     """Verify sending large 'username' raises the right exception."""
124 +     body_dict = _build_user_auth(username='0' * 65, password='foo2')
125 +     self.assertRaises(exception.ValidationError, self.api.authenticate,
126 +                      {}, body_dict)
127 +
128 + def test_authenticate_tenant_id_too_large(self):
129 +     """Verify sending large 'tenantId' raises the right exception."""
130 +     body_dict = _build_user_auth(username='F00', password='foo2',
131 +                                 tenant_id='0' * 65)
132 +     self.assertRaises(exception.ValidationError, self.api.authenticate,
133 +                      {}, body_dict)
134 +
135 + def test_authenticate_tenant_name_too_large(self):
136 +     """Verify sending large 'tenantName' raises the right exception."""
137 +     body_dict = _build_user_auth(username='F00', password='foo2',
138 +                                 tenant_name='0' * 65)
139 +     self.assertRaises(exception.ValidationError, self.api.authenticate,
140 +                      {}, body_dict)
141 +
142 + def test_authenticate_token_too_large(self):
143 +     """Verify sending large 'token' raises the right exception."""
144 +     body_dict = _build_user_auth(token={'id': '0' * 8193})
145 +     self.assertRaises(exception.ValidationError, self.api.authenticate,
146 +                      {}, body_dict)
147 +
148 + def test_authenticate_password_too_large(self):
149 +     """Verify sending large 'password' raises the right exception."""
150 +     body_dict = _build_user_auth(username='F00', password='0' * 8193)
151 +     self.assertRaises(exception.ValidationError, self.api.authenticate,
152 +                      {}, body_dict)
```

Comments 0



Please [sign in](#) to comment.