

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>pymongo.errors.OperationFailure: Authentication failed., full error: {'ok':
0.0, 'errmsg': 'Authentication failed.', 'code': 18, 'codeName': 'AuthenticationFailed'} //
Werkzeug Debugger</title>
    <link rel="stylesheet" href="?
__debugger__=yes&cmd=resource&f=style.css"
    type="text/css">
    <!-- We need to make sure this has a favicon so that the debugger does
    not by accident trigger a request to /favicon.ico which might
    change the application state. -->
    <link rel="shortcut icon"
    href="?__debugger__=yes&cmd=resource&f=console.png">
    <script src="?__debugger__=yes&cmd=resource&f=jquery.js"></
script>
    <script src="?__debugger__=yes&cmd=resource&f=debugger.js"></
script>
    <script type="text/javascript">
    var TRACEBACK = 139973754252560,
    CONSOLE_MODE = false,
    EVALEX = true,
    EVALEX_TRUSTED = false,
    SECRET = "dB4zB4kbmRGcar8xLlxe";
  </script>
</head>
<body style="background-color: #fff">
  <div class="debugger">
    <h1>pymongo.errors.OperationFailure</h1>
    <div class="detail">
      <p class="errmsg">pymongo.errors.OperationFailure: Authentication
failed., full error: {'ok': 0.0, 'errmsg': 'Authentication failed.', 'code': 18, 'codeName':
'AuthenticationFailed'}</p>
    </div>
    <h2 class="traceback">Traceback
      <em>(most recent call last)</em>
    </h2>
    <div class="traceback">
      <ul>
        <li>
          <div class="frame" id="frame-139973754347920">
            <h4>File
              <cite class="filename">"/usr/local/lib/python3.7/site-packages/
flask/app.py"</cite>,
              line
                <em class="line">2464</em>,
              in
                <code class="function">__call__</code>
            </h4>

```

```

<div class="source library">
  <pre class="line before">
    <span class="ws"></span>
  </pre>
  <pre class="line before">
    <span class="ws">    </span>def __call__(self, environ,
start_response):
  </pre>
  <pre class="line before">
    <span class="ws">    </span>&quot;&quot;&quot;The
WSGI server calls the Flask application object as the
  </pre>
  <pre class="line before">
    <span class="ws">    </span>WSGI application. This
calls :meth:`wsgi_app` which can be
  </pre>
  <pre class="line before">
    <span class="ws">    </span>wrapped to applying
middleware.&quot;&quot;&quot;
  </pre>
  <pre class="line current">
    <span class="ws">    </span>return self.wsgi_app(environ,
start_response)
  </pre>
  <pre class="line after">
    <span class="ws"></span>
  </pre>
  <pre class="line after">
    <span class="ws">    </span>def __repr__(self):
  </pre>
  <pre class="line after">
    <span class="ws">    </span>return &quot;&lt;%s
%r&gt;&quot; % (self.__class__.__name__, self.name)
  </pre>
</div>
</div>
<li>
  <div class="frame" id="frame-139973754348240">
    <h4>File
      <cite class="filename">"/usr/local/lib/python3.7/site-packages/
flask/app.py"</cite>,
      line
        <em class="line">2450</em>,
      in
        <code class="function">wsgi_app</code>
    </h4>
    <div class="source library">
      <pre class="line before">
        <span class="ws">    </span>try:
      </pre>

```

```

        <pre class="line before">
            <span class="ws">                </span>ctx.push()
        </pre>
        <pre class="line before">
            <span class="ws">                </span>response =
self.full_dispatch_request()
        </pre>
        <pre class="line before">
            <span class="ws">                </span>except Exception as e:
        </pre>
        <pre class="line before">
            <span class="ws">                </span>error = e
        </pre>
        <pre class="line current">
            <span class="ws">                </span>response =
self.handle_exception(e)
        </pre>
        <pre class="line after">
            <span class="ws">                </span>except: # noqa: B001
        </pre>
        <pre class="line after">
            <span class="ws">                </span>error = sys.exc_info()
[1]
        </pre>
        <pre class="line after">
            <span class="ws">                </span>raise
        </pre>
        <pre class="line after">
            <span class="ws">                </span>return response(envIRON,
start_response)
        </pre>
        <pre class="line after">
            <span class="ws">                </span>finally:
        </pre>
    </div>
</div>
<li>
    <div class="frame" id="frame-139973754347600">
        <h4>File
            <cite class="filename">"/usr/local/lib/python3.7/site-
packages/flask/app.py"</cite>,
line
            <em class="line">1867</em>,
in
            <code class="function">handle_exception</code>
        </h4>
        <div class="source library">
            <pre class="line before">
                <span class="ws">                </span># if we want to
repropagate the exception, we can attempt to

```

```

    </pre>
    <pre class="line before">
      <span class="ws">      </span># raise it with the
whole traceback in case we can do that
    </pre>
    <pre class="line before">
      <span class="ws">      </span># (the function was
actually called from the except part)
    </pre>
    <pre class="line before">
      <span class="ws">      </span># otherwise, we just
raise the error again
    </pre>
    <pre class="line before">
      <span class="ws">      </span>if exc_value is e:
    </pre>
    <pre class="line current">
      <span class="ws">      </span>reraise(exc_type,
exc_value, tb)
    </pre>
    <pre class="line after">
      <span class="ws">      </span>else:
    </pre>
    <pre class="line after">
      <span class="ws">      </span>raise e
    </pre>
    <pre class="line after">
      <span class="ws"></span></pre>
    <pre class="line after">
      <span class="ws">      </span>self.log_exception((exc_type, exc_value, tb))
    </pre>
    <pre class="line after">
      <span class="ws">      </span>server_error =
InternalServerError()
    </pre>
  </div>
</div>
</li>
<div class="frame" id="frame-139973754348368">
  <h4>File
    <cite class="filename">"/usr/local/lib/python3.7/site-
packages/flask/_compat.py"</cite>,
    line
      <em class="line">39</em>,
    in
      <code class="function">reraise</code>
  </h4>
<div class="source library">

```

```

collections_abc
    <pre class="line before">
        <span class="ws">    </span>import collections.abc as
    </pre>
    <pre class="line before">
        <span class="ws"></span>
    </pre>
    <pre class="line before">
        <span class="ws">    </span>def reraise(tp, value,
tb=None):
    </pre>
    <pre class="line before">
        <span class="ws">    </span>if
value.__traceback__ is not tb:
    </pre>
    <pre class="line before">
        <span class="ws">    </span>raise
value.with_traceback(tb)
    </pre>
    <pre class="line current">
        <span class="ws">    </span>raise value
    </pre>
    <pre class="line after">
        <span class="ws"></span>
    </pre>
    <pre class="line after">
        <span class="ws">    </span>implements_to_string =
_identity
    </pre>
    <pre class="line after">
        <span class="ws"></span>
    </pre>
    <pre class="line after">
        <span class="ws"></span>else:
    </pre>
    <pre class="line after">
        <span class="ws">    </span>iterkeys = lambda d:
d.iterkeys()
    </pre>
</div>
</div>
<li>
    <div class="frame" id="frame-139973754348432">
        <h4>File
            <cite class="filename">"/usr/local/lib/python3.7/site-
packages/flask/app.py"</cite>,
                line
                    <em class="line">2447</em>,
                in
                    <code class="function">wsgi_app</code>

```

```

</h4>
<div class="source library">
  <pre class="line before">
    <span class="ws">    </span>ctx =
self.request_context(envIRON)
  </pre>
  <pre class="line before">
    <span class="ws">    </span>error = None
  </pre>
  <pre class="line before">
    <span class="ws">    </span>try:
  </pre>
  <pre class="line before">
    <span class="ws">    </span>try:
  </pre>
  <pre class="line before">
    <span class="ws">    </span>ctx.push()
  </pre>
  <pre class="line current">
    <span class="ws">    </span>response =
self.full_dispatch_request()
  </pre>
  <pre class="line after">
    <span class="ws">    </span>except Exception
as e:
  </pre>
  <pre class="line after">
    <span class="ws">    </span>error = e
  </pre>
  <pre class="line after">
    <span class="ws">    </span>response =
self.handle_exception(e)
  </pre>
  <pre class="line after">
    <span class="ws">    </span>except: # noqa:
B001
  </pre>
  <pre class="line after">
    <span class="ws">    </span>error =
sys.exc_info()[1]
  </pre>
</div>
</div>
<li>
  <div class="frame" id="frame-139973754348624">
    <h4>File
      <cite class="filename">"/usr/local/lib/python3.7/site-
packages/flask/app.py" </cite>,
      line
        <em class="line">1952</em>,

```

```

in
    <code class="function">full_dispatch_request</code>
code>
    </h4>
    <div class="source library">
        <pre class="line before">
            <span class="ws">        </span>
span>request_started.send(self)
        </pre>
        <pre class="line before">
            <span class="ws">        </span>rv =
self.preprocess_request()
        </pre>
        <pre class="line before">
            <span class="ws">        </span>if rv is None:
        </pre>
        <pre class="line before">
            <span class="ws">        </span>rv =
self.dispatch_request()
        </pre>
        <pre class="line before">
            <span class="ws">        </span>except Exception
as e:
        </pre>
        <pre class="line current">
            <span class="ws">        </span>rv =
self.handle_user_exception(e)
        </pre>
        <pre class="line after">
            <span class="ws">        </span>return
self.finalize_request(rv)
        </pre>
        <pre class="line after">
            <span class="ws"></span></pre>
        <pre class="line after">
            <span class="ws">    </span>def
finalize_request(self, rv, from_error_handler=False):
        </pre>
        <pre class="line after">
            <span class="ws">        </span>
span>&quot;&quot;&quot;Given the return value from a view function this finalizes
        </pre>
        <pre class="line after">
            <span class="ws">        </span>the request by
converting it into a response and invoking the
        </pre>
    </div>
</div>
<li>

```

```

<div class="frame" id="frame-139973754348304">
  <h4>File
    <cite class="filename">"/usr/local/lib/python3.7/
site-packages/flask/app.py"</cite>,
  line
    <em class="line">1821</em>,
  in
    <code class="function">handle_user_exception</
code>
  </h4>
  <div class="source library">
    <pre class="line before">
      <span class="ws">      </span>return
self.handle_http_exception(e)
    </pre>
    <pre class="line before">
      <span class="ws"></span>
    </pre>
    <pre class="line before">
      <span class="ws">      </span>handler =
self._find_error_handler(e)
    </pre>
    <pre class="line before">
      <span class="ws"></span>
    </pre>
    <pre class="line before">
      <span class="ws">      </span>if handler is
None:
    </pre>
    <pre class="line current">
      <span class="ws">      </span>
span>reraise(exc_type, exc_value, tb)
    </pre>
    <pre class="line after">
      <span class="ws">      </span>return
handler(e)
    </pre>
    <pre class="line after">
      <span class="ws"></span>
    </pre>
    <pre class="line after">
      <span class="ws">      </span>def
handle_exception(self, e):
    </pre>
    <pre class="line after">
      <span class="ws">      </span>
span>"&quot;&quot;&quot;Handle an exception that did not have an error handler
    </pre>
    <pre class="line after">
      <span class="ws">      </span>associated

```

with it, or that was raised from an error handler.

```
</pre>
</div>
</div>
<li>
  <div class="frame" id="frame-139973754348176">
    <h4>File
      <cite class="filename">"/usr/local/lib/
python3.7/site-packages/flask/_compat.py"</cite>,
    line
      <em class="line">39</em>,
    in
      <code class="function">reraise</code>
    </h4>
    <div class="source library">
      <pre class="line before">
        <span class="ws">  </span>import
collections.abc as collections_abc
      </pre>
      <pre class="line before">
        <span class="ws"></span>
      </pre>
      <pre class="line before">
        <span class="ws">  </span>def reraise(tp,
value, tb=None):
      </pre>
      <pre class="line before">
        <span class="ws">    </span>if
value.__traceback__ is not tb:
      </pre>
      <pre class="line before">
        <span class="ws">    </span>raise
value.with_traceback(tb)
      </pre>
      <pre class="line current">
        <span class="ws">    </span>raise value
      </pre>
      <pre class="line after">
        <span class="ws"></span>
      </pre>
      <pre class="line after">
        <span class="ws">  </span>
span>implements_to_string = _identity
      </pre>
      <pre class="line after">
        <span class="ws"></span>
      </pre>
      <pre class="line after">
        <span class="ws"></span>else:
      </pre>
```

```

lambda d: d.iterkeys()
    <pre class="line after">
        <span class="ws">    </span>iterkeys =
    </pre>
</div>
</div>
<li>
    <div class="frame"
id="frame-139973754348752">
        <h4>File
            <cite class="filename">"/usr/local/lib/
python3.7/site-packages/flask/app.py"</cite>,
            line
                <em class="line">1950</em>,
            in
                <code
class="function">full_dispatch_request</code>
                </h4>
                <div class="source library">
                    <pre class="line before">
                        <span class="ws">    </
span>self.try_trigger_before_first_request_functions()
                    </pre>
                    <pre class="line before">
                        <span class="ws">    </span>try:
                    </pre>
                    <pre class="line before">
                        <span class="ws">    </
span>request_started.send(self)
                    </pre>
                    <pre class="line before">
                        <span class="ws">    </span>rv =
self.preprocess_request()
                    </pre>
                    <pre class="line before">
                        <span class="ws">    </span>if rv is
None:
                    </pre>
                    <pre class="line current">
                        <span class="ws">    </span>rv =
self.dispatch_request()
                    </pre>
                    <pre class="line after">
                        <span class="ws">    </span>except
Exception as e:
                    </pre>
                    <pre class="line after">
                        <span class="ws">    </span>rv =
self.handle_user_exception(e)
                    </pre>

```

```

        <pre class="line after">
            <span class="ws">    </span>return
self.finalize_request(rv)
        </pre>
        <pre class="line after">
            <span class="ws"></span>
        </pre>
        <pre class="line after">
            <span class="ws">    </span>def
finalize_request(self, rv, from_error_handler=False):
        </pre>
    </div>
</div>
<li>
    <div class="frame"
id="frame-139973754348496">
        <h4>File
            <cite class="filename">"/usr/local/lib/
python3.7/site-packages/flask/app.py"</cite>,
            line
                <em class="line">1936</em>,
            in
                <code
class="function">dispatch_request</code>
                </h4>
                <div class="source library">
                    <pre class="line before">
                        <span class="ws">    </span>
span>getattr(rule, &quot;provide_automatic_options&quot;, False)
                    </pre>
                    <pre class="line before">
                        <span class="ws">    </span>and
req.method == &quot;OPTIONS&quot;;
                    </pre>
                    <pre class="line before">
                        <span class="ws">    </span>):
                    </pre>
                    <pre class="line before">
                        <span class="ws">    </span>
span>return self.make_default_options_response()
                    </pre>
                    <pre class="line before">
                        <span class="ws">    </span>#
otherwise dispatch to the handler for that endpoint
                    </pre>
                    <pre class="line current">
                        <span class="ws">    </span>return
self.view_functions[rule.endpoint](**req.view_args)
                    </pre>
                    <pre class="line after">

```

```

        <span class="ws"></span>
    </pre>
    <pre class="line after">
        <span class="ws">    </span>def
full_dispatch_request(self):
        </pre>
        <pre class="line after">
            <span class="ws">    </
span>&quot;&quot;&quot;Dispatches the request and on top of that performs request
        </pre>
        <pre class="line after">
            <span class="ws">    </span>pre
and postprocessing as well as HTTP exception catching and
        </pre>
        <pre class="line after">
            <span class="ws">    </span>error
handling.
        </pre>
    </div>
</div>
<li>
    <div class="frame"
id="frame-139973754347664">
        <h4>File
        <cite class="filename">"/app/app.py"</
cite>,
        line

        in

        <em class="line">52</em>,

        <code class="function">login</code>
    </h4>
    <div class="source ">
        <pre class="line before">
            <span class="ws"></span>
        </pre>
        <pre class="line before">
            <span class="ws">    </span>email=
request.form['email']
        </pre>
        <pre class="line before">
            <span class="ws">    </
span>password= request.form['password']
        </pre>
        <pre class="line before">
            <span class="ws">    </
span>response= &quot;&quot;;
        </pre>
        <pre class="line before">
            <span class="ws"></span>
    </pre>

```

```

span>query_result = Citizen.objects(email=email)
len(query_result) > 0:
bcrypt.check_password_hash(query_result[0]['password'], password):
query_result[0]['citizen_type'] == 'employee':
id="frame-139973754347728">
python3.7/site-packages/mongoengine/queryset/manager.py
line
in
code>
span># Document object being used rather than a document class
span>return self

```

owner is the document that contains the QuerySetManager

```
</pre>
<pre class="line before">
  <span class="ws">    </
span>queryset_class = owner._meta.get(&quot;queryset_class&quot;;, self.default)
</pre>
<pre class="line current">
  <span class="ws">    </
span>queryset = queryset_class(owner, owner._get_collection())
</pre>
<pre class="line after">
  <span class="ws">    </span>if
```

self.get_queryset:

```
</pre>
<pre class="line after">
  <span class="ws">    </
span>arg_count = self.get_queryset.__code__.co_argcount
</pre>
<pre class="line after">
  <span class="ws">    </
span>if arg_count == 1:
  </pre>
  <pre class="line after">
    <span class="ws">    </
```

span>queryset = self.get_queryset(queryset)

```
</pre>
<pre class="line after">
  <span class="ws">    </
span>elif arg_count == 2:
```

```
</pre>
</div>
</div>
<li>
  <div class="frame"
```

id="frame-139973754349072">

```
<h4>File
  <cite class="filename">"/usr/local/
lib/python3.7/site-packages/mongoengine/document.py"</cite>,
  line
  in
```

```
<code
class="function">_get_collection</code>
```

```
</h4>
<div class="source library">
  <pre class="line before">
    <span class="ws">    </
span>if not hasattr(cls, &quot;_collection&quot;,) or cls._collection is None:
  </pre>
  <pre class="line before">
```

```

># Get the collection, either capped or regular.
</pre>
<pre class="line before">
<span class="ws"> </
>if cls._meta.get(&quot;max_size&quot;) or
cls._meta.get(&quot;max_documents&quot;):
</pre>
<pre class="line before">
<span class="ws"> </
>cls._collection = cls._get_capped_collection()
</pre>
<pre class="line before">
<span class="ws"> </
>else:
</pre>
<pre class="line current">
<span class="ws"> </
>db = cls._get_db()
</pre>
<pre class="line after">
<span class="ws"> </
>collection_name = cls._get_collection_name()
</pre>
<pre class="line after">
<span class="ws"> </
>cls._collection = db[collection_name]
</pre>
<pre class="line after">
<span class="ws"></span>
</pre>
<pre class="line after">
<span class="ws"> </
># Ensure indexes on the collection unless auto_create_index was
</pre>
<pre class="line after">
<span class="ws"> </
># set to False.
</pre>
</div>
</div>
<li>
<div class="frame"
id="frame-139973754348944">
<h4>File
<cite class="filename">"/usr/
local/lib/python3.7/site-packages/mongoengine/document.py" </cite>,
line
<em class="line">185</em>,
in

```

class="function">_get_db</code>

span>return hash(self.pk)

span>@classmethod

span>def _get_db(cls):

span>"""Some Model using other db_alias""";

span>return get_db(cls._meta.get("db_alias";
DEFAULT_CONNECTION_NAME))

span>@classmethod

span>def _disconnect(cls):

span>"""Detach the Document class from the (cached) database
collection""";

span>cls._collection = None

<code

</h4>

<div class="source library">

<pre class="line before">

 </

</pre>

<pre class="line before">

</pre>

<pre class="line before">

 </

</pre>

<pre class="line before">

 </

</pre>

<pre class="line before">

 </

</pre>

<pre class="line current">

 </

</pre>

<pre class="line after">

</pre>

<pre class="line after">

 </

</pre>

<pre class="line after">

 </

</pre>

<pre class="line after">

 </

</pre>

<pre class="line after">

 </

</pre>

</div>

</div>

```

</li>
<div class="frame"
id="frame-139973754349200">
    <h4>File
    <cite class="filename">"/usr/
local/lib/python3.7/site-packages/mongoengine/connection.py"</cite>,
    line
    <em class="line">368</em>,
    in
    <code
class="function">get_db</code>
    </h4>
    <div class="source library">
    <pre class="line before">
    <span class="ws">    </
span>if conn_settings[&quot;username&quot;] and (
    </pre>
    <pre class="line before">
    <span class="ws">
    </span>conn_settings[&quot;password&quot;]
    </pre>
    <pre class="line before">
    <span class="ws">
    </span>or conn_settings[&quot;authentication_mechanism&quot;] ==
    &quot;MONGODB-X509&quot;
    </pre>
    <pre class="line before">
    <span class="ws">    </
span>):
    </pre>
    <pre class="line before">
    <span class="ws">
    </span>db.authenticate(
    </pre>
    <pre class="line current">
    <span class="ws">
    </span>conn_settings[&quot;username&quot;],
    conn_settings[&quot;password&quot;], **auth_kwargs
    </pre>
    <pre class="line after">
    <span class="ws">
    </span>
    </pre>
    <pre class="line after">
    <span class="ws">    </
span>_dbs[alias] = db
    </pre>
    <pre class="line after">
    <span class="ws">    </
span>return _dbs[alias]

```

span>

span>

id="frame-139973754349136">

usr/local/lib/python3.7/site-packages/pymongo/database.py"</cite>,
line

em>,
in

class="function">authenticate</code>

self.name)

span>

self.client._cache_credentials(

self.name,

credentials,

connect=True)

```
</pre>
<pre class="line after">
  <span class="ws"></
```

```
</pre>
<pre class="line after">
  <span class="ws"></
```

```
</pre>
</div>
</div>
<li>
  <div class="frame"
```

```
<h4>File
  <cite class="filename">"/
  <em class="line">1495</
```

```
<code
```

```
</h4>
<div class="source library">
  <pre class="line before">
    <span class="ws">
```

```
</pre>
<pre class="line before">
  <span class="ws"></
```

```
</pre>
<pre class="line before">
  <span class="ws">
```

```
</pre>
<pre class="line before">
  <span class="ws">
```

```
</pre>
<pre class="line before">
  <span class="ws">
```

```
</pre>
<pre class="line current">
  <span class="ws">
```

```
</pre>
<pre class="line after">
```

span>	</
	</pre>
	<pre class="line after">
	
return True	
	</pre>
	<pre class="line after">
	</
span>	
	</pre>
	<pre class="line after">
	 </
span>def logout(self):	
	</pre>
	<pre class="line after">
	
""";**DEPRECATED**: Deauthorize use of this database.	
	</pre>
	</div>
	</div>
	<div class="frame"
id="frame-139973754349264">	
	<h4>File
	<cite
class="filename">"/usr/local/lib/python3.7/site-packages/pymongo/	
mongo_client.py"</cite>,	
line	
	<em class="line">781</
em>,	
in	
	<code
class="function">_cache_credentials</code>	
	</h4>
	<div class="source
library">	
before">	<pre class="line
	
writable_preferred_server_selector)	
	</pre>
	<pre class="line
before">	
	</
span>	
	</pre>
	<pre class="line
before">	
	

```

</span># get_socket() logs out of the database if logged in with old
before">
</span># credentials, and logs in with new ones.
before">
</span>with server.get_socket(all_credentials) as sock_info:
current">
</span>sock_info.authenticate(credentials)
span>
</span># If several threads run _cache_credentials at once, last one wins.
</span>self.__all_credentials[source] = credentials
span>
</span>def _purge_credentials(self, source):
id="frame-139973754349328">
class="filename">"/usr/local/lib/python3.7/site-packages/pymongo/pool.py"</cite>,
line
class="line">810</em>,
in

```

class="function">authenticate</code>

library">

before">

Can raise ConnectionFailure or OperationFailure.

before">

class="ws">

before">

:Parameters:

before">

- `credentials`: A MongoCredential.

before">

"""

current">

auth.authenticate(credentials, self)

after">

self.authset.add(credentials)

after">

negotiated_mechanisms are no longer needed.

after">

self.negotiated_mechanisms.pop(credentials, None)

</h4>

<div class="source

<pre class="line

</pre>

<pre class="line

<span

</pre>

<pre class="line

</pre>

<pre class="line

</pre>

<pre class="line

</pre>

<pre class="line

</pre>

<pre class="line

</pre>

<pre class="line

</pre>

<pre class="line

</pre>

<pre class="line

```

after">
                                <span class="ws">
</span>self.auth_ctx.pop(credentials, None)
                                </pre>
                                <pre class="line
after">
                                <span
class="ws"></span>
                                </pre>
                                </div>
                                </div>
                                <li>
                                <div class="frame"
id="frame-139973754349392">
                                <h4>File
                                <cite
class="filename">"/usr/local/lib/python3.7/site-packages/pymongo/auth.py"</cite>,
                                line
                                <em
class="line">673</em>,
                                in
                                <code
class="function">authenticate</code>
                                </h4>
                                <div class="source
library">
                                <pre class="line
before">
                                <span
class="ws"></span>
                                </pre>
                                <pre class="line
before">
                                <span
class="ws"></span>def authenticate(credentials, sock_info):
                                </pre>
                                <pre class="line
before">
                                <span
class="ws"> </span>&quot;&quot;&quot;Authenticate
                                sock_info.&quot;&quot;&quot;;
                                </pre>
                                <pre class="line
before">
                                <span
class="ws"> </span>mechanism = credentials.mechanism
                                </pre>
                                <pre class="line
before">
                                <span

```

```

class="ws">  </span>auth_func = _AUTH_MAP.get(mechanism)
current">
class="ws">  </span>auth_func(credentials, sock_info)
after">
class="ws"></span>
after">
class="ws"></span>
after">
class="ws"></span>def logout(source, sock_info):
after">
class="ws">  </span>"Log out from a
database.""""
after">
class="ws">  </span>sock_info.command(source, {'logout': 1})
id="frame-139973754349456">
class="filename">"/usr/local/lib/python3.7/site-packages/pymongo/auth.py"</cite>,
line
class="line">591</em>,
in
class="function">_authenticate_default</code>
library">

```

| | | |
|-------------|--|-------------------|
| before"> | | </pre class="line |
| class="ws"> | source, cmd, publish_events=False).get(| </pre> |
| before"> | | </pre class="line |
| class="ws"> | 'saslSupportedMechs', []) | </pre> |
| before"> | | </pre class="line |
| class="ws"> | if 'SCRAM-SHA-256' in mechs: | </pre> |
| before"> | | </pre class="line |
| class="ws"> | return _authenticate_scam(credentials, sock_info, | </pre> |
| | 'SCRAM-SHA-256') | </pre class="line |
| before"> | | </pre> |
| class="ws"> | else: | </pre> |
| current"> | | </pre class="line |
| class="ws"> | return _authenticate_scam(credentials, sock_info, | </pre> |
| | 'SCRAM-SHA-1') | </pre class="line |
| after"> | | </pre> |
| class="ws"> | elif sock_info.max_wire_version >= 3: | </pre> |
| after"> | | </pre class="line |
| class="ws"> | return _authenticate_scam(credentials, sock_info, | </pre> |
| | 'SCRAM-SHA-1') | </pre class="line |
| after"> | | </pre> |
| class="ws"> | else: | </pre> |
| after"> | | </pre class="line |

```
<span
class="ws">    </span>return _authenticate_mongo_cr(credentials, sock_info)
</pre>
<pre class="line
after">
<span
class="ws"></span>
</pre>
</div>
</div>
<li>
<div class="frame"
id="frame-139973754349520">
<h4>File
<cite
class="filename">"/usr/local/lib/python3.7/site-packages/pymongo/auth.py"</cite>,
line
class="line">295</em>,
in
class="function">_authenticate_scram</code>
</h4>
<div>
<pre>
class="line before">
<span
class="ws">    </span>nonce, first_bare = ctx.scram_data
</pre>
<pre>
class="line before">
<span
class="ws">    </span>res = ctx.speculative_authenticate
</pre>
<pre>
class="line before">
<span
class="ws">    </span>else:
</pre>
<pre>
class="line before">
<span
class="ws">    </span>nonce, first_bare, cmd = _authenticate_scram_start(
</pre>
<pre>
class="line before">
<span
class="ws">    </span>credentials, mechanism)
</pre>
```

```
class="line current">
class="ws">    </span>res = sock_info.command(source, cmd)
class="line after">
class="ws"></span>
class="line after">
class="ws">    </span>server_first = res['payload']
class="line after">
class="ws">    </span>parsed = _parse_scam_response(server_first)
class="line after">
class="ws">    </span>iterations = int(parsed[b'i'])
class="line after">
class="ws">    </span>if iterations &lt; 4096:
class="frame" id="frame-139973754349584">
class="filename">"/usr/local/lib/python3.7/site-packages/pymongo/pool.py"</cite>,
line
class="line">694</em>,
in
class="function">command</code>
class="source library">
class="line before">
```

<pre>

</pre>
<pre>

</pre>
<pre>

</pre>
<pre>

</pre>
<pre>

</pre>
<pre>

</pre>
<pre>

</pre>
<div>
</div>
<div>

<div>
<h4>File
<cite>
</cite>,&br/>
,&br/><code>
</h4>
<div>
<pre>


```

class="ws">
        </span>collation=collation,
    </pre>
</pre>
class="line before">
    <span
class="ws">
        </span>compression_ctx=self.compression_context,
    </pre>
</pre>
class="line before">
    <span
class="ws">
        </span>use_op_msg=self.op_msg_enabled,
    </pre>
</pre>
class="line before">
    <span
class="ws">
        </span>unacknowledged=unacknowledged,
    </pre>
</pre>
class="line before">
    <span
class="ws">
        </span>user_fields=user_fields,
    </pre>
</pre>
class="line current">
    <span
class="ws">
        </span>exhaust_allowed=exhaust_allowed)
    </pre>
</pre>
class="line after">
    <span
class="ws">
        </span>except OperationFailure:
    </pre>
</pre>
class="line after">
    <span
class="ws">
        </span>raise
    </pre>
</pre>
class="line after">
    <span
class="ws">
        </span># Catch socket.error, KeyboardInterrupt, etc. and close
        ourselves.
    </pre>
</pre>
class="line after">
    <span
class="ws">
        </span>except BaseException as error:
    </pre>
</pre>
class="line after">

```

```
class="ws">                </span>self._raise_connection_failure(error)                <span
                                </pre>
                                </div>
                                </div>
                                <li>
                                <div>

class="frame" id="frame-139973754349648">                                <h4>File
                                <cite
class="filename">"/usr/local/lib/python3.7/site-packages/pymongo/network.py"</
cite>,
                                line
                                <em>
class="line">162</em>,
                                in
                                <code>
class="function">command</code>
                                </h4>
                                <div>

class="source library">
                                <pre>
class="line before">
                                <span
class="ws">                </span>client._process_response(response_doc, session)
                                </pre>
                                <pre>
class="line before">
                                <span
class="ws">                </span>if check:
                                </pre>
                                <pre>
class="line before">
                                <span
class="ws">                </span>helpers._check_command_response(
                                </pre>
                                <pre>
class="line before">
                                <span
class="ws">                </span>response_doc, sock_info.max_wire_version, None,
                                </pre>
                                <pre>
class="line before">
                                <span
class="ws">                </span>allowable_errors,
                                </pre>
                                <pre>
class="line current">
                                <span
class="ws">                </
```

```

span>parse_write_concern_error=parse_write_concern_error)
</pre>
<pre>
class="line after">
<span>
class="ws"> </span>except Exception as exc:
</pre>
<pre>
class="line after">
<span>
class="ws"> </span>if publish:
</pre>
<pre>
class="line after">
<span>
class="ws"> </span>duration = (datetime.datetime.now() - start) +
encoding_duration
</pre>
<pre>
class="line after">
<span>
class="ws"> </span>if isinstance(exc, (NotMasterError, OperationFailure)):
</pre>
<pre>
class="line after">
<span>
class="ws"> </span>failure = exc.details
</pre>
</div>
</div>
<li>
<div>
class="frame" id="frame-139973754349712">
<h4>File
<cite>
class="filename">"/usr/local/lib/python3.7/site-packages/pymongo/helpers.py"</
cite>,
line
<em>
class="line">168</em>,
in
<code>
class="function">_check_command_response</code>
</h4>
<div>
class="source library">
<pre>
class="line before">
<span class="ws"> </span>raise CursorNotFound(errmsg, code, response,

```

```

class="line before">
<span class="ws">
</span>max_wire_version)
</pre>
<pre>
class="line before">
<span class="ws"></span>
</pre>
<pre>
class="line before">
<span class="ws">
</span>msg = msg or &quot;%s&quot;;
</pre>
<pre>
class="line before">
<span class="ws">
</span>raise OperationFailure(msg % errmsg, code,
response,
</pre>
<pre>
class="line current">
<span class="ws">
</span>max_wire_version)
</pre>
<pre>
class="line after">
<span class="ws"></span>
</pre>
<pre>
class="line after">
<span class="ws"></span>
</pre>
<pre>
class="line after">
<span class="ws"></span>def _check_gle_response(result, max_wire_version):
</pre>
<pre>
class="line after">
<span class="ws">
</span>&quot;&quot;&quot;Return getlasterror response as a
dict, or raise OperationFailure.&quot;&quot;&quot;;
</pre>
<pre>
class="line after">

```

 # Did getlasterror itself fail?

</pre>
</div>
</div>

<blockquote>pymongo.errors.OperationFailure: Authentication failed., full error: {'ok': 0.0, 'errmsg': 'Authentication failed.', 'code': 18, 'codeName': 'AuthenticationFailed'}</blockquote>

</div>
<div>

class="plain">

<form

action="/?__debugger__=yes&cmd=paste" method="post">

<p>
<input

type="hidden" name="language" value="pytb">

This is the Copy/Paste friendly version of the traceback.

<span

class="pastemessage">You can also paste this traceback into
a

gist:

<input

type="submit" value="create paste">

</p>
<textarea

cols="50" rows="10" name="code" readonly>Traceback (most recent call last):

File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2464, in
__call__

return self.wsgi_app(environ, start_response)

File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2450, in
wsgi_app

response = self.handle_exception(e)

File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1867, in
handle_exception

reraise(exc_type, exc_value, tb)

File "/usr/local/lib/python3.7/site-packages/flask/_compat.py", line 39, in
reraise

raise value

File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2447, in
wsgi_app

response = self.full_dispatch_request()

File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1952, in
full_dispatch_request

rv = self.handle_user_exception(e)

File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1821, in
handle_user_exception

```
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python3.7/site-packages/flask/_compat.py", line 39, in
reraise
    raise value
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1950, in
full_dispatch_request
    rv = self.dispatch_request()
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1936, in
dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
File "/app/app.py", line 52, in login
    query_result = Citizen.objects(email=email)
File "/usr/local/lib/python3.7/site-packages/mongoengine/queryset/
manager.py", line 37, in __get__
    queryset = queryset_class(owner, owner._get_collection())
File "/usr/local/lib/python3.7/site-packages/mongoengine/document.py",
line 207, in _get_collection
    db = cls._get_db()
File "/usr/local/lib/python3.7/site-packages/mongoengine/document.py",
line 185, in _get_db
    return get_db(cls._meta.get("db_alias",
DEFAULT_CONNECTION_NAME))
File "/usr/local/lib/python3.7/site-packages/mongoengine/
connection.py", line 368, in get_db
    conn_settings["username"], conn_settings["password"],
**auth_kwargs
File "/usr/local/lib/python3.7/site-packages/pymongo/database.py", line
1495, in authenticate
    connect=True)
File "/usr/local/lib/python3.7/site-packages/pymongo/mongo_client.py",
line 781, in _cache_credentials
    sock_info.authenticate(credentials)
File "/usr/local/lib/python3.7/site-packages/pymongo/pool.py", line 810,
in authenticate
    auth.authenticate(credentials, self)
File "/usr/local/lib/python3.7/site-packages/pymongo/auth.py", line 673,
in authenticate
    auth_func(credentials, sock_info)
File "/usr/local/lib/python3.7/site-packages/pymongo/auth.py", line 591,
in _authenticate_default
    return _authenticate_scram(credentials, sock_info, 'SCRAM-SHA-1')
File "/usr/local/lib/python3.7/site-packages/pymongo/auth.py", line 295,
in _authenticate_scram
    res = sock_info.command(source, cmd)
File "/usr/local/lib/python3.7/site-packages/pymongo/pool.py", line 694,
in command
    exhaust_allowed=exhaust_allowed)
File "/usr/local/lib/python3.7/site-packages/pymongo/network.py", line
162, in command
    parse_write_concern_error=parse_write_concern_error)
```

File "/usr/local/lib/python3.7/site-packages/pymongo/helpers.py";, line
168, in _check_command_response
max_wire_version)
pymongo.errors.OperationFailure: Authentication failed., full error: {'ok': 0.0, 'errmsg':
'Authentication failed.', 'code': 18, 'codeName': 'AuthenticationFailed'}</textarea>

</form>
</div>
<div

class="explanation">

The debugger caught an exception in your WSGI application. You can now
look at the traceback which led to the error.

<span

class="nojavascript">

If you enable JavaScript you can also use additional features such as code
execution (if the evalex feature is enabled), automatic pasting of the
exceptions and much more.

</div>
<div

class="footer">

Brought to you by

<strong

class="arthur">DON'T PANIC, your
friendly Werkzeug powered traceback interpreter.

</div>
</div>
<div class="pin-

prompt">

<div

class="inner">

<h3>Console

Locked</h3>

<p>

The console is locked and needs to be unlocked by entering the PIN.
You can find the PIN printed out on the standard output of your
shell that runs the server.

<form>
<p>PIN:

type=text name=pin size=14>

<input

type=submit name=btn value="Confirm Pin">

<input

</form>
</div>
</div>
</body>
</html>
<!--

Traceback (most recent call last):

```
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2464, in __call__
    return self.wsgi_app(environ, start_response)
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2450, in wsgi_app
    response = self.handle_exception(e)
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1867, in
handle_exception
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python3.7/site-packages/flask/_compat.py", line 39, in reraise
    raise value
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2447, in wsgi_app
    response = self.full_dispatch_request()
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1952, in
full_dispatch_request
    rv = self.handle_user_exception(e)
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1821, in
handle_user_exception
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python3.7/site-packages/flask/_compat.py", line 39, in reraise
    raise value
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1950, in
full_dispatch_request
    rv = self.dispatch_request()
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1936, in
dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
File "/app/app.py", line 52, in login
    query_result = Citizen.objects(email=email)
File "/usr/local/lib/python3.7/site-packages/mongoengine/queryset/manager.py",
line 37, in __get__
    queryset = queryset_class(owner, owner._get_collection())
File "/usr/local/lib/python3.7/site-packages/mongoengine/document.py", line 207, in
_get_collection
    db = cls._get_db()
File "/usr/local/lib/python3.7/site-packages/mongoengine/document.py", line 185, in
_get_db
    return get_db(cls._meta.get("db_alias", DEFAULT_CONNECTION_NAME))
File "/usr/local/lib/python3.7/site-packages/mongoengine/connection.py", line 368,
in get_db
    conn_settings["username"], conn_settings["password"], **auth_kwargs
File "/usr/local/lib/python3.7/site-packages/pymongo/database.py", line 1495, in
authenticate
    connect=True)
File "/usr/local/lib/python3.7/site-packages/pymongo/mongo_client.py", line 781, in
_cache_credentials
    sock_info.authenticate(credentials)
File "/usr/local/lib/python3.7/site-packages/pymongo/pool.py", line 810, in
authenticate
    auth.authenticate(credentials, self)
```

```
File "/usr/local/lib/python3.7/site-packages/pymongo/auth.py", line 673, in
authenticate
    auth_func(credentials, sock_info)
File "/usr/local/lib/python3.7/site-packages/pymongo/auth.py", line 591, in
_authenticate_default
    return _authenticate_scram(credentials, sock_info, 'SCRAM-SHA-1')
File "/usr/local/lib/python3.7/site-packages/pymongo/auth.py", line 295, in
_authenticate_scram
    res = sock_info.command(source, cmd)
File "/usr/local/lib/python3.7/site-packages/pymongo/pool.py", line 694, in command
    exhaust_allowed=exhaust_allowed)
File "/usr/local/lib/python3.7/site-packages/pymongo/network.py", line 162, in
command
    parse_write_concern_error=parse_write_concern_error)
File "/usr/local/lib/python3.7/site-packages/pymongo/helpers.py", line 168, in
_check_command_response
    max_wire_version)
pymongo.errors.OperationFailure: Authentication failed., full error: {'ok': 0.0, 'errmsg':
'Authentication failed.', 'code': 18, 'codeName': 'AuthenticationFailed'}
```