Driver Authentication

Abstract

MongoDB supports various authentication strategies across various versions. When authentication is turned on in the database, a driver must authenticate before it is allowed to communicate with the server. This spec defines when and how a driver performs authentication with a MongoDB server.

This report is following a specification from the Mongodb website and is shown below.

Mongo credential

Drivers should contain a type called MongoCredential. In this driver it is called MongoDB::Authenticate::Credential.

```
use lib 't';
use Test-support;
use MongoDB::Authenticate::Credential;
use MongoDB::Client;
use BSON::Document;
my MongoDB::Test-support $ts .= new;
my MongoDB::Client $client;
my MongoDB::Authenticate::Credential $cred .= new(
  :username<user>, :password<pencil>,
  :auth-mechanism<SCRAM-SHA-1>
);
ok $cred.defined, 'T0';
is $cred.password, "pencil", 'T1';
```

✔ **T0:** Credential is defined. Not all named arguments are used. There are also :source and :mechanism-properties. Btw. the field mechanism is named auth-mechanism because of the name authMechanism used in the url as an option.

✔ **T1:** Data can also be retrieved from it again.

Some combinations of credential parameters should fail

```
my $x = {
  $cred .= new(
    :username<user>, :password<pencil>, :auth-mechanism<MONGODB-X509>
  );
};
dies-ok $x, 'T2';
my $y = {
  $cred .= new( :password<pencil>, :auth-mechanism<MONGODB-CR>);
};
dies-ok $y, 'T3';
```

✔ **T2:** Mechanism MONGODB-X509 does not expect a password

✔ **T3:** Mechanism MONGODB-CR must have a username

Credential parameters are taken from the information in the uri. The authentication mechanism can also be guessed from the max wire version from the info returned by the isMaster command. This, however can only be checked after a socket has been opened caused by a user action. B.t.w Cleanup the client object when done because its concurrent processes will keep connected to the server even after undefining.

```
my Int $port1 = $ts.server-control.get-port-number('s1');
$client .= new(:uri("mongodb://localhost:$port1"));
ok $client.defined, 'T4';
is $client.credential.auth-mechanism, "", 'T5';
my $database = $client.database('tdb'); my BSON::Document $req .= new: ( insert => 'tcol', documents => [ BSON::Document.new(( name => 'Larry', surname => 'Walll', )), ] ); my BSO
is $doc<ok>, 1, 'T6';
is $doc<n>, 1, 'T7';
is $client.credential.auth-mechanism, "SCRAM-SHA-1", 'T8';
$client.cleanup;
```

✔ **T4:** Returned client object is defined, even when the uri is not pointing to an existing mongod server.

✔ **T5:** Mechanism is not yet defined.

✔ **T6:** insert request ok

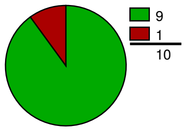
✔ **T7:** inserted 1 document

✘ **T8:** Mechanism is a SCRAM-SHA-1 version

```
$client .= new(:uri("mongodb://localhost:$port1"));
ok $client.defined, 'T9';
$client.cleanup;
```

✔ **T9:** Returned client object is defined, even when the uri is not pointing to an existing mongod server.

Normal tests



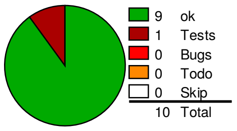
Bug issues

🟩 0
🟥 0
0

Todo tests

🟩 0
🟧 0
0

Summary



Generated using SemiXML, SxmlLib::Testing::*, XML, ©Google prettify