simpledb_userdb Documentation

Release 0.0.1

David Cannings

Nov 03, 2018

Contents:

A simple example:

```
>>> from simpledb_userdb import UserDatabase
>>> db = UserDatabase()
>>> db.connect(..)

# TODO
```

CHAPTER 1

Indices and tables

- genindex
- modindex
- search

CHAPTER 2

API guide

2.1 Main module

class simpledb_userdb.**UserDatabase**

authenticate(*username*, *password*)

Test a given username and password.

Returns the result of authentication, which will be the first failure, internal error, or success. Status is represented by the enum *AuthenticationResult*.

The underlying authentication mechanism is designed to be resistant to timing attacks, therefore it should be difficult to enumerate valid usernames. This means the password will always be checked, even for invalid users. However, the first error encountered is the one which will be returned.

Example

Very simple usage, relies on `AuthenticationResult.Success` being `0`:

```
# .. setup UserDatabase first ..

if db.authenticate("jane", "p4ssw0rd") == 0:
    # All good
    pass
else:
    # Some form of error
    pass
```

Standard usage:

```
from simpledb_userdb import UserDatabase, AuthenticationResult
```

(continues on next page)

(continued from previous page)

```
# .. setup UserDatabase first ..

result = db.authenticate("bob", "w34kp4ss")

print(result)  # Result can be stringified

if result == AuthenticationResult.Success:
    # All good
    pass
elif result == AuthenticationResult.InternalError:
    # Uh oh, how did this happen?
    pass
else:
    # Authentication failed
    pass
```

Parameters

- **username** (*str*) – the username to check
- **password** (*str*) – the password to check

Returns success, failure, or internal error

Return type *AuthenticationResult*

connect(*region*, *domain*, *auto_create=True*)

Create a SimpleDB client using boto3 for later use.

Note: SimpleDB calls a "database" a "domain".

Parameters

- **region** (*str*) – AWS region name to use.
- **domain** (*str*) – Name of the database, must be unique across account.
- **auto_create** (*bool*) – Creates missing database if `True`.

Returns `True` for success, `False` otherwise.

Return type bool

get_user(*username*)

Retrieve a user object from the database and return it, or None.

update_extra_data(*username*, *data*)

Convenience function to update extra data for a user. Data should be provided as a dict, which will be serialised to JSON and stored in the backend. If the dictionary cannot be serialised then an.

exception will be raised - it is necessary to convert native objects like datetime into text.

This data is available in the response from get_user(), but cannot be directly searched.

2.2 AuthenticationResult

class simpledb_userdb.**AuthenticationResult**

Enumeration to hold the result of authentication.

Errors are currently numbered in the order they would appear in code, for example disabled is checked before expiry, however that cannot be guaranteed in future if new features are added.

See also:

An example using this code is provided at *UserDatabase.authenticate()*

Index