python-tds Documentation

Release 1.6

Mikhail Denisenko

Aug 23, 2023

Contents

Pytds is the top to bottom pure Python TDS implementation, that means cross-platform, and no dependency on ADO or FreeTDS. It supports large parameters (>4000 characters), MARS, timezones, new date types (datetime2, date, time, datetimeoffset). Even though it is implemented in Python performance is comparable to ADO and FreeTDS bindings.

It also supports Python 3.

Contents

CHAPTER 1

pytds – main module

CHAPTER 2

pytds.login – login with NTLM and SSPI

CHAPTER 3

pytds.tz – timezones

CHAPTER 4

pytds.extensions – Extensions to the DB API

4.1 Isolation level constants

pytds.extensions.**ISOLATION_LEVEL_READ_UNCOMMITTED**
: Transaction can read uncommitted data

pytds.extensions.**ISOLATION_LEVEL_READ_COMMITTED**
: Transaction can read only committed data, will block on attempt to read modified uncommitted data

pytds.extensions.**ISOLATION_LEVEL_REPEATABLE_READ**
: Transaction will place lock on read records, other transactions will block trying to modify such records

pytds.extensions.**ISOLATION_LEVEL_SERIALIZABLE**
: Transaction will lock tables to prevent other transactions from inserting new data that would match selected recordsets

pytds.extensions.**ISOLATION_LEVEL_SNAPSHOT**
: Allows non-blocking consistent reads on a snapshot for transaction without blocking other transactions changes

CHAPTER 5

Connection to Mirrored Servers

When MSSQL server is setup with mirroring you should connect to it using two parameters of `pytds.connect()`, one parameter is `server` this should be a main server and parameter `failover_partner` should be a mirror server. See also MSDN article.

CHAPTER 6

Table Valued Parameters

Here is example of using TVP:

```
with conn.cursor() as cur:
    cur.execute('CREATE TYPE dbo.CategoryTableType AS TABLE ( CategoryID int,␣
↪CategoryName nvarchar(50) )')
    conn.commit()

    tvp = pytds.TableValuedParam(type_name='dbo.CategoryTableType', rows=rows_gen())
    cur.execute('SELECT * FROM %s', (tvp,))
```

CHAPTER 7

Testing

To run tests you need to have tox installed. Also you would want to have different versions of Python, you can use pyenv to install those.

At a minimun you should set HOST environment variable to point to your SQL server, e.g.:

```
export HOST=mysqlserver
```

it could also specify SQL server named instance, e.g.:

```
export HOST=mysqlserver\\myinstance
```

By default tests will use SQL server integrated authentication using user sa with password sa and database test. You can specify different user name, password, database with SQLUSER, SQLPASSWORD, DATABASE environment variables.

To enable testing NTLM authentication you should specify NTLM_USER and NTLM_PASSWORD environment variables.

Once environment variables are setup you can run tests by running command:

```
tox
```

Test configuration stored in tox.ini file at the root of the repository.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index

I

P