pgcli Documentation

Release stable

May 07, 2015

Contents

This is a postgres client that does auto-completion and syntax highlighting.

```
stayupdated_test> INSERT INTO goose_db_version
                              admins
                              cpes
                              goose_db_version
                              packages
                              users




stayupdated_test> \d
+----------+-------------------------+----------+---------+
| Schema   | Name                    | Type     | Owner   |
|----------+-------------------------+----------+---------|
| public   | admins                  | table    | amjith  |
| public   | cpes                    | table    | amjith  |
| public   | goose_db_version        | table    | amjith  |
| public   | goose_db_version_id_seq | sequence | amjith  |
| public   | packages                | table    | amjith  |
| public   | packages_id_seq         | sequence | amjith  |
| public   | users                   | table    | amjith  |
| public   | users_id_seq            | sequence | amjith  |
| public   | vulnerabilities         | table    | amjith  |
| public   | vulnerabilities_cpes    | table    | amjith  |
| public   | vulnerabilities_id_seq  | sequence | amjith  |
+----------+-------------------------+----------+---------+
SELECT 11
stayupdated_test> SELECT * FROM users;
+------+--------------+----------+-------------------------+----------------------------+
|   id | display_name | password | email                   | created_on                 |
|------+--------------+----------+-------------------------+----------------------------|
|  177 | DisplayName1 | 1024cms  | user@ex.com             | 2014-11-15 15:02:50.094560 |
|  180 | testname2    | pas5w0rd | email@ex.com            | 2014-11-28 10:25:46.170660 |
|  181 | amjith       | password | amjith@amjith.amjith    | 2014-11-28 18:39:48.195067 |
+------+--------------+----------+-------------------------+----------------------------+
SELECT 3
stayupdated_test> SELECT * FROM
                              admins
                              cpes
                              goose_db_version
                              packages
                              users
```

Quick Start

If you already know how to install python packages, then you can simply do:

```
$ pip install pgcli

or

$ brew install pgcli  # Only on OS X
```

If you don't know how to install python packages, please check the detailed instructions.

Usage

```
$ pgcli [database_name]

or

$ pgcli postgresql://[user[:password]@][netloc][:port][/dbname]
```

Examples:

```
$ pgcli local_database

$ pgcli postgres://amjith:pa$$w0rd@example.com:5432/app_db
```

Features

The *pgcli* is written using prompt_toolkit.

- Auto-completion as you type for SQL keywords as well as tables and columns in the database.

- Syntax highlighting using Pygments.

- Smart-completion (enabled by default) will suggest context-sensitive completion.

    - `SELECT * FROM <tab>` will only show table names.

    - `SELECT * FROM users WHERE <tab>` will only show column names.

- Config file is automatically created at `~/.pgclirc` at first launch.

- Primitive support for `psql` back-slash commands.

- Pretty prints tabular data.

Contributions:

If you're interested in contributing to this project, first of all I would like to extend my heartfelt gratitude. I've written a small doc to describe how to get this running in a development setup.

https://github.com/amjith/pgcli/blob/master/DEVELOP.rst

Please feel free to reach out to me if you need help. My email: amjith.r@gmail.com, Twitter: @amjithr

Detailed Installation Instructions:

5.1 OS X:

Easiest way to install pgcli is using brew. Please be aware that this will install postgres via brew if it wasn't installed via brew.

```
$ brew install pgcli
```

Done!

If you have postgres installed via a different means (such as PostgresApp), you can `brew install --build-from-source pgcli` which will skip installing postgres via brew if postgres is available in the path.

Alternatively, you can install `pgcli` as a python package using a package manager called called `pip`. You will need postgres installed on your system for this to work.

In depth getting started guide for `pip` - https://pip.pypa.io/en/latest/installing.html.

```
$ which pip
```

If it is installed then you can do:

```
$ pip install pgcli
```

If that fails due to permission issues, you might need to run the command with sudo permissions.

```
$ sudo pip install pgcli
```

If pip is not installed check if easy_install is available on the system.

```
$ which easy_install

$ sudo easy_install pgcli
```

5.2 Linux:

In depth getting started guide for `pip` - https://pip.pypa.io/en/latest/installing.html.

Check if pip is already available in your system.

```
$ which pip
```

If it doesn't exist, use your linux package manager to install *pip*. This might look something like:

```
$ sudo apt-get install python-pip    # Debian, Ubuntu, Mint etc

or

$ sudo yum install python-pip  # RHEL, Centos, Fedora etc
```

`pgcli` requires python-dev, libpq-dev and libevent-dev packages. You can install these via your operating system package manager.

```
$ sudo apt-get install python-dev libpq-dev libevent-dev

or

$ sudo yum install python-devel postgresql-devel
```

Then you can install pgcli:

```
$ sudo pip install pgcli
```

Thanks:

A special thanks to Jonathan Slenders for creating Python Prompt Toolkit, which is quite literally the backbone library, that made this app possible. Jonathan has also provided valuable feedback and support during the development of this app.

This app includes the awesome tabulate library for pretty printing the output of tables. The reason for vendoring this library rather than listing it as a dependency in setup.py, is because I had to make a change to the table format which is merged back into the original repo, but not yet released in PyPI.

Click is used for command line option parsing and printing error messages.

Thanks to psycopg for providing a rock solid interface to Postgres database.

Thanks to all the beta testers and contributors for your time and patience. :)