

Contents

1	Introduction	1
2	Migration from .NET / Mono to Python	2
3	Installation	3
4	Creating Users and Connections	3
4.1	Logging in as user <i>admin</i>	3
4.2	Creating another user	5
4.3	Signing in as the new user	9
4.4	Creating connections	9

1 Introduction

DBMS stands for *database management system*. It may be a simple function library or even a larger system composed by several programs and processes running separately and in parallel, whose main function is to manage one or several databases hosted in a server. It has the responsibility to manipulate and to keep the consistency of data, allowing the software developers to focus on functionalities. Thus, practically any modern system that manages data utilizes some kind of DBMS, regardless of the amount of stored information.

OmniDB's first version was created as an undergrad final project in the Computer Science Course from the Federal University of Paraná. The objective was to trace a common line between popular DBMS, and to study deeply their *metadata*. The result was a tool capable of connecting and identifying the main structures (tables, keys, indexes and constraints), in a generic way, from several DBMS:

- Firebird
- MariaDB / MySQL
- Oracle
- PostgreSQL
- SQLite
- Microsoft SQL Server

Since early development, OmniDB was designed as an web app. Consequently, it runs in any browser, from any operational system. It can be accessed by several computers and multiple users, each one of them with his/her own group of connections. It also may host any operational system, without the need of install any dependencies. We will see further details on installation in the next chapters.

OmniDB's main objective is to offer an unified workspace with all functionalities needed to manipulate different DMBS. DBMS specific tools aren't required: in OmniDB, the context switch between different DBMS is done with a simple connection switch, without leaving the same page. The end-user's sensation is that there is no difference when he/she manipulates different DBMS, it just feels like different connections.

2 Migration from .NET / Mono to Python

OmniDB was rewritten to Python using the Django framework. Starting from version 2.0, OmniDB Python version will receive new features and will be actively maintained.

The source code for the ASP.NET/C# version is in the branch `csharp`. The next release of OmniDB C# version is 1.7, and it will only receive bug fixes.

OmniDB source code is hosted on GitHub and there are 3 main branches:

- **master**: Contains the current beta release of OmniDB Python version
- **dev**: Contains the current development release of OmniDB Python version
- **csharp**: Contains the .NET / Mono version of OmniDB

Besides being written in Python, initial version of OmniDB 2.0 contains the following main differences from the C# version:

- Support to HTTPS;
- It allows query execution in background and cancellation through the use of websockets;
- Initially, only an improved support of PostgreSQL is implemented. More RDBMS support coming soon;
- There is a new Snippet feature;

- Log capabilities and a test suite are being developed;
- You don't need to install dependencies and web servers any more. Everything OmniDB needs is now bundled in a single executable.

3 Installation

In order to run OmniDB, you don't need to install any additional piece of software. Just head to omnidb.org and download the latest zip package for your specific operating system and architecture:

- Linux 32 bits / 64 bits
- Windows 32 bits / 64 bits
- Mac OSX

Extract the zip file somewhere in your computer. It will create a folder called `omnidb`. Get inside this folder and run the `omnidb` executable.

```
user@machine:~$ cd omnidb
user@machine:~/omnidb$ ./omnidb
Starting OmniDB at http://localhost:12000
Open OmniDB in your favorite browser
```

Now that the web server is running, you may access OmniDB web app on your favorite browser. Type in address bar: `localhost:12000` and hit **Enter**. If you are running in any port other than 12000, utilize the appropriate port number. If everything went fine, you shall see a page like this:

Now you know that OmniDB is running correctly. In the next chapters, we will see how to login for the first time, how to create an user and to utilize OmniDB.

4 Creating Users and Connections

4.1 Logging in as user *admin*

OmniDB comes only with the user *admin*. The first thing to do is sign in as admin, the default password is admin.

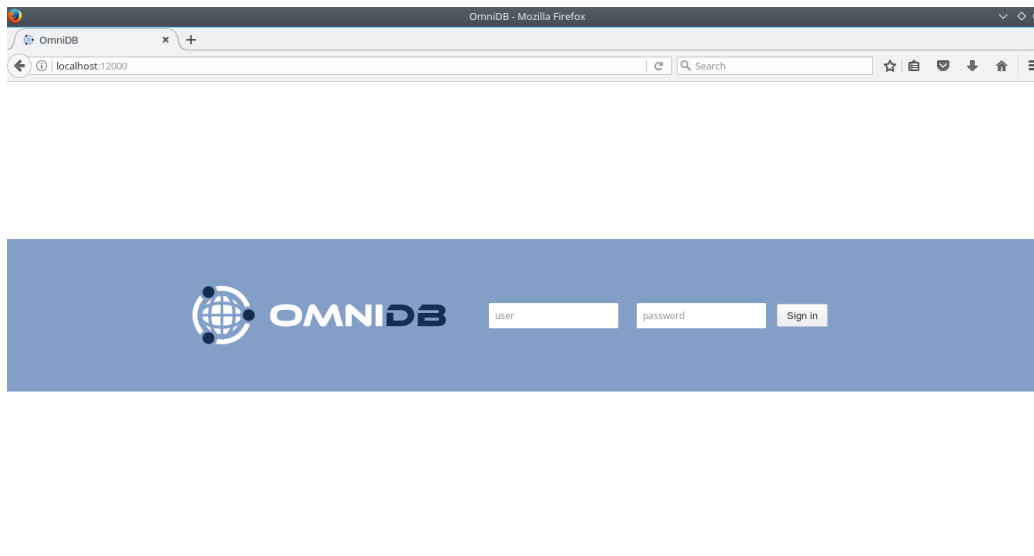


Figure 1: First execution

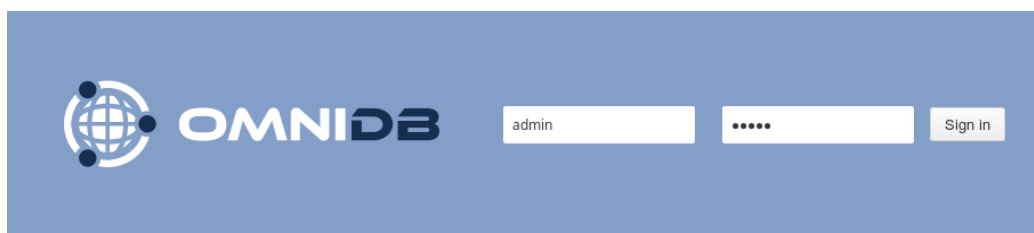


Figure 2: Signing in as admin

The next window is the **Connections** window. We will talk about it later.

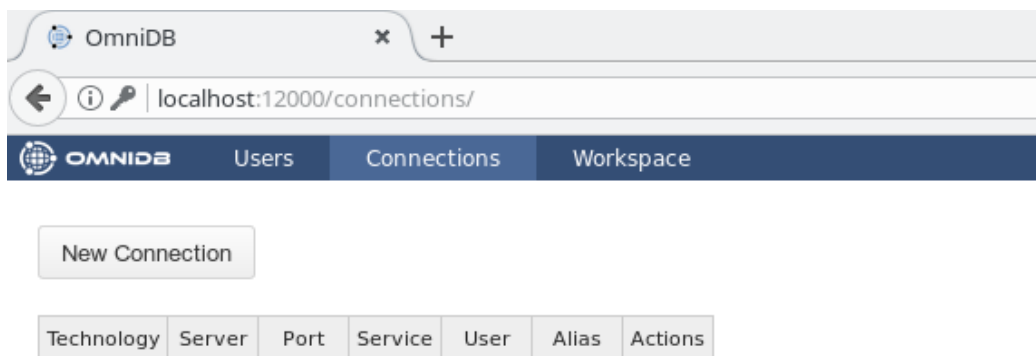


Figure 3: Connections window as admin

4.2 Creating another user

Click on the *Users* tab.

After clicking on the *New User* button the tool inserts a new user called *user2* (if that is the first user after *admin*).

You will have to change the *username* and *password*. Check if you want this new user to be a *super user*. This user management window is only seen by super users. When you are done, click on the *Save Data* button.

You can create as many users as you want, edit existing users and also delete users by clicking on the red cross at the actions column. Now you can logout.

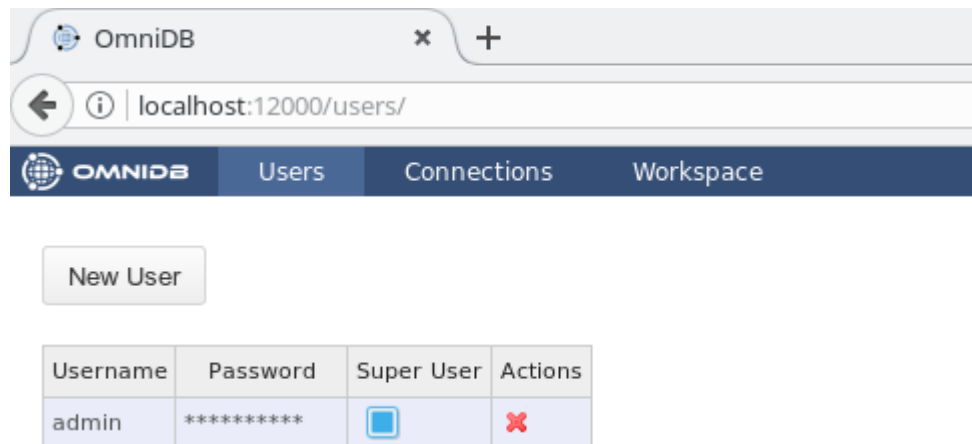


Figure 4: User management window

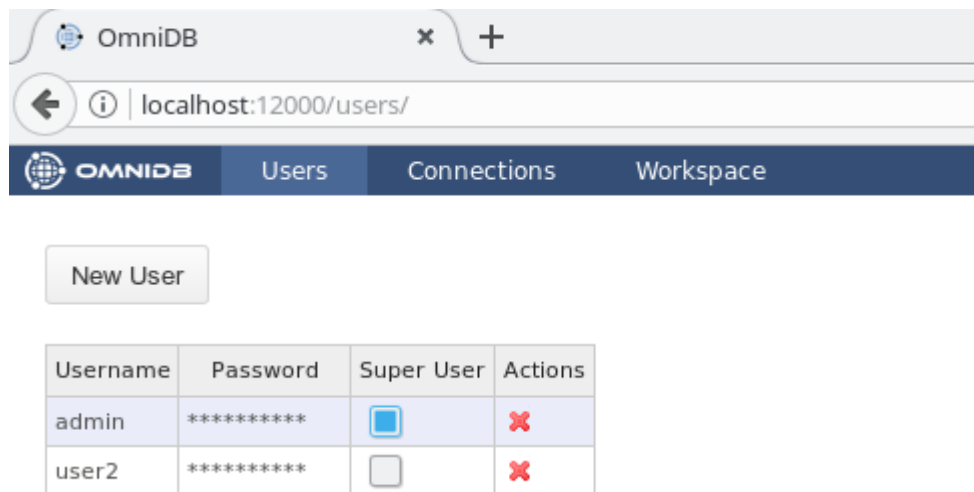


Figure 5: Creating new user

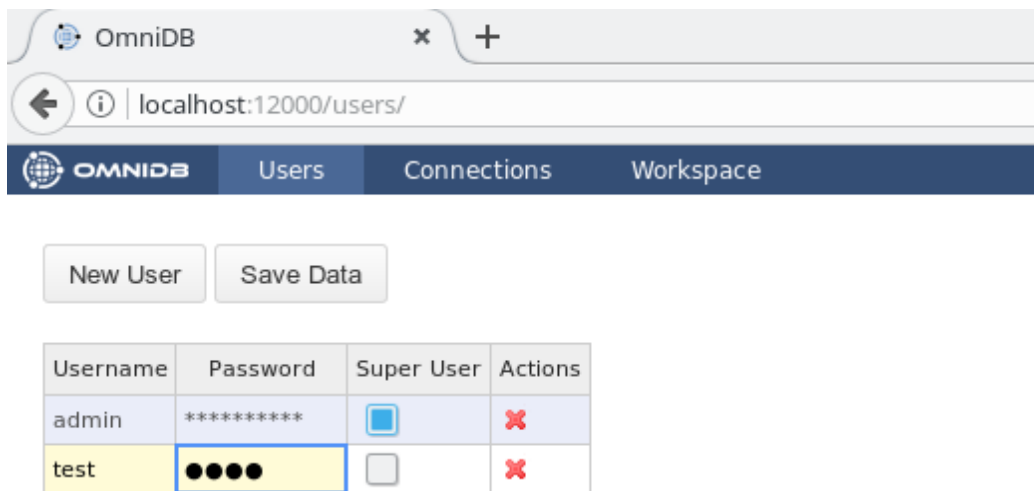


Figure 6: Changing new user attributes

4.3 Signing in as the new user

Let us sign in as the user we just created.



Figure 7: Signing in as the new user

And we can see the **Connections** window again. Note that now there is no *Users* tab, because the *test* user is not a super user.

4.4 Creating connections

OmniDB C# version supported several DBMS. At the moment, OmniDB Python version, or OmniDB 2.0, supports only PostgreSQL. More DBMS support is being added as you read this.

We will now create two connections to PostgreSQL databases. To create the connections you have to click on the button *New Connection* and then choose the connection and fill the other fields. After filling all the fields for both connections, click on the *Save Data* button.

For each connection there is an *Actions* column where you can delete and test them. Go ahead and test one of the connections.

Notice the *Password Expired* popup. This is happening because OmniDB does not store the database user password on disk. When the user types a password in this popup, the password is encrypted and stored in memory.

After you type the password and hit *Enter*, if the connection to the database is successful you will see a confirmation popup.

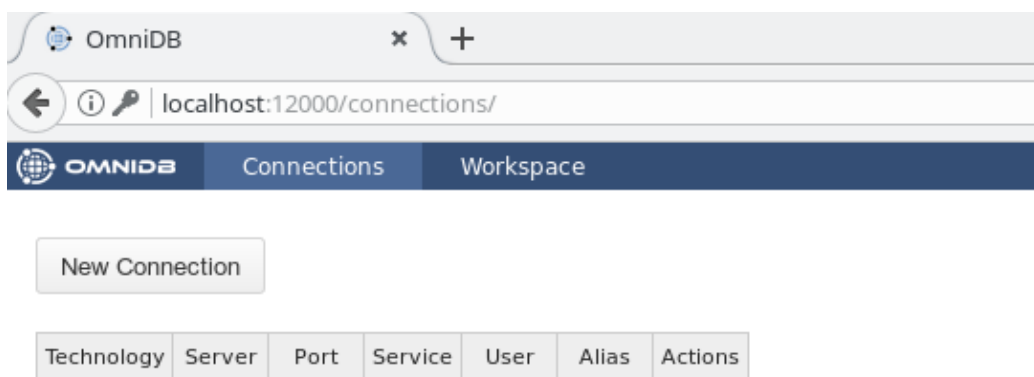


Figure 8: Connections window

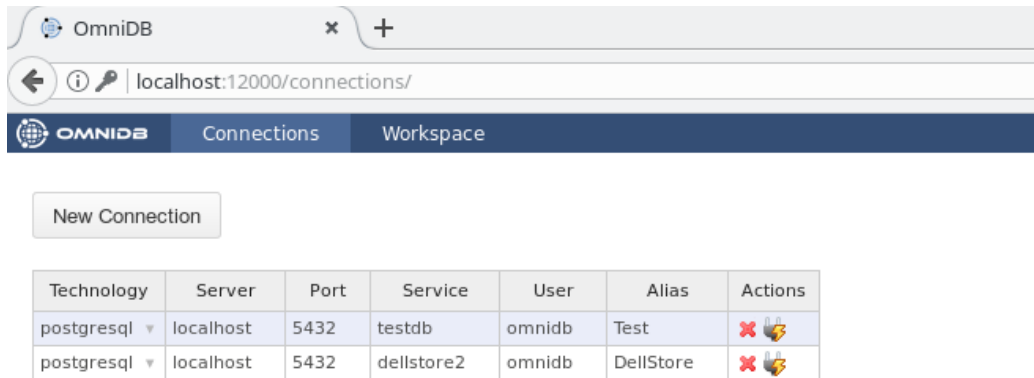


Figure 9: Some connections created

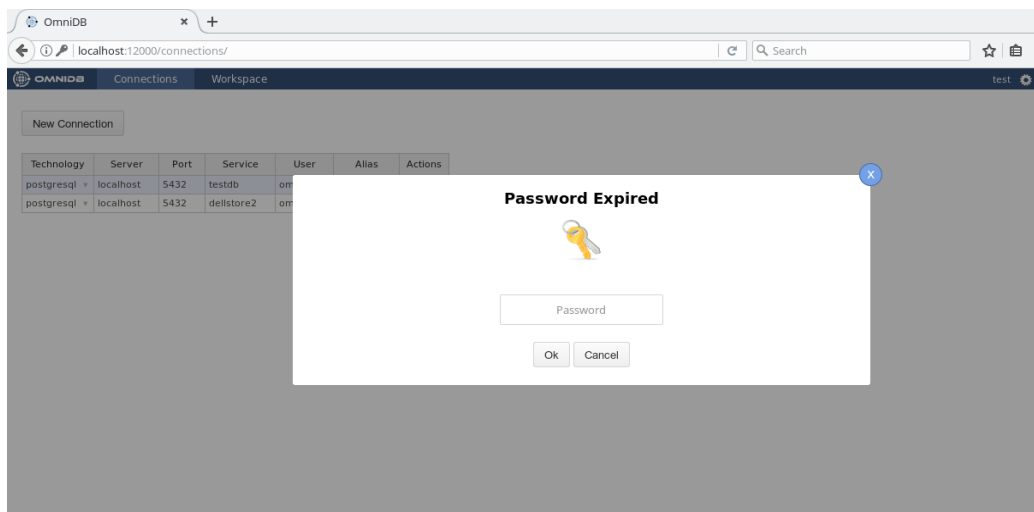


Figure 10: Some connections created

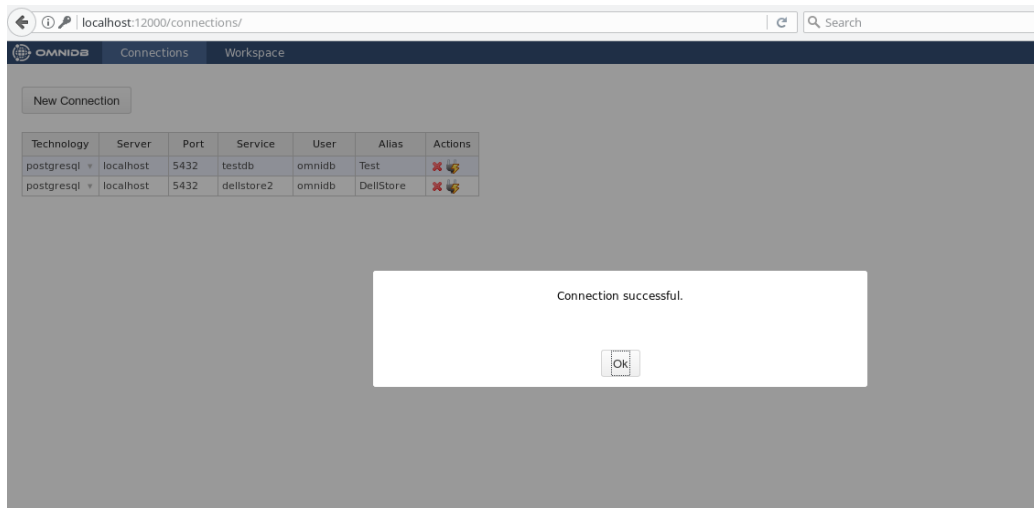


Figure 11: Connection successful

But, if you have trouble of any kind connecting to your PostgreSQL database, the *Password Expired* popup will remain showing the error OmniDB got.

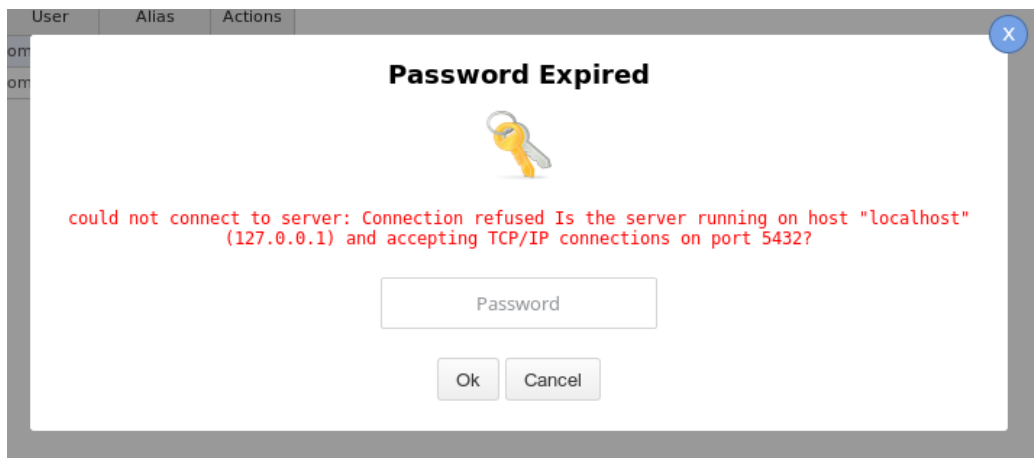


Figure 12: Connection unsuccessful