**Database / PDO / PHP Information**

For use with an Apache server. Install Bitnami WAPP or equivalent.

During the installation, there will be a prompt for username & password, that is for the Postgres database that you are installing. This is not asking for the credentials for the database that we are using, you may submit anything here.

The setup will also ask you if the WAPP stack should be launched in the cloud, this is not required.

Browse to the installation directory -> apache2 -> htdocs, and place the api folder there.

This file path is what is currently used in the app (http://localhost/api). For quick access to the application folder or if you can’t find the installation directory, you can launch the stack manager tool and click “Open Application Folder”.

In order for the php to connect with the database, we use PDO, which requires installation of PDO drivers. see http://php.net/manual/en/ref.pdo-sqlsrv.php#ref.pdo-sqlsrv.installation

Follow the link for SQLSRV, and download PHP driver for Windows.

Download SQLSRV40.exe, run it to extract the dll files to a temp folder.

We need to put 1 dll into the php installation directory, in Bitnami WAPP’s case, this is at WAPP installation dir -> php -> ext (you should see a bunch of other dll files here).

The dll file we need depends on your environment. The WAPP I used is the 64-bit version, so we only look at the filenames containing x64.

The filenames without PDO are also not needed.

This leaves us with php\_pdo\_sqlsrv\_7\_nts\_x64 & php\_pdo\_sqlsrv\_7\_ts\_x64.

ts = thread-safe, nts = not thread-safe.

I am currently using ts, which seems to work for now. Unsure of the exact differences.

Also, note the ‘7’ in the filename. The number refers to the php version you are using. The php included with the WAPP I am using has version of 7.x. If you are using a different php version, I assume you will have to download a different SQLSRVxx.exe.

Copy the file (php\_pdo\_sqlsrv\_7\_ts\_x64 in my case) into the ext folder.

Now we have to configure php to include this extension. Navigate up 1 folder from the ext folder. Open php.ini.

Each line starting with ; is ignored (commented out).

Add "extension=php\_pdo\_sqlsrv\_7\_ts\_x64.dll” without the quotation marks.

If you are using a different dll file, match the filename accordingly.

I assume the exact position where this line is inserted does not matters, but I placed it with the other “extension=php\_pdo\_xxx” lines.

(Note, the dll we installed is for connecting to a Microsoft SQL Server database. If a different type of database is used, such as SQLite, use the appropriate pdo\_sqlite.dll file etc.)

In order to make connections from the app to the database, launch the WAPP stack, click “Manage Servers”, and start “Apache Web Server”. Postgres is not used so there is no need to run it.

Note: sometimes the server will show that it is running but the app will not be able to create a connection. This may happen randomly or after the server has not been contacted for some period of time left running. In such cases, restart the server, and make sure its status is running.

At this point, you should be able to make the connection. The following text are for creating new PHP files. If you are not creating / editing the PHP files, you may skip the following sections.

Now, to connect from php to the db, I am using the db\_connect.php file (located in the api folder).

It contains the server name, password etc. Change the credentials here if there is a change in the database choice.

Add “require\_once 'db\_connect.php’;” to each php file requiring the connection (see any of the PHP files in the api folder). These files will then have access to the variable $conn, as it is now in the global scope. (If I am not wrong, should you wish to access $conn in internal functions that you create within these files, you will have to pass $conn into the function.)

Then you can use $conn to perform queries. Look at the PHP files or read the PDO online documentation for how to execute queries.