PIPSTA011 –SENDING A PRINT-JOB TO THE WEB

Contents

Revision History

Revision	Author	Date	Description
1.0	AH	27/11/14	First Release

Difficulty Level: ●○○○○

- There is no pre-requisite technical understanding to working through this tutorial

Time to Complete: ●●○○○

- Aside from moving between Python and web-hosted databases, this tutorial should not take more than half an hour to complete.

Who Should Read This Document

This tutorial is suitable for anyone wishing to demonstrate printing over the web as a basis for Internet of Things and/or Internet of People applications. This document also provides a good grounding for subsequent web printing tutorials.

Warning

Although this is a primitive implementation, it is quick, simple to understand and portable between web hosting companies. This is not necessarily the most robust nor secure implementation, so please be aware of these limitations and DO NOT use this as the basis for anything other than a demonstration system!

Prerequisites

- It is expected that you have a fully-working, web-connected Raspberry Pi and Pipsta
- You should have the following settings from your web hosting company by having followed ***PIPSTA010- Simple WebPrinting – Pre-requisites*** :
  - Username
  - Password
  - Host IP address
  - Database name
  - Port

__Additional Configuration Step__

- Open LXTerminal
- At the $ prompt, enter:

```
sudo apt-get install python-mysqldb
```

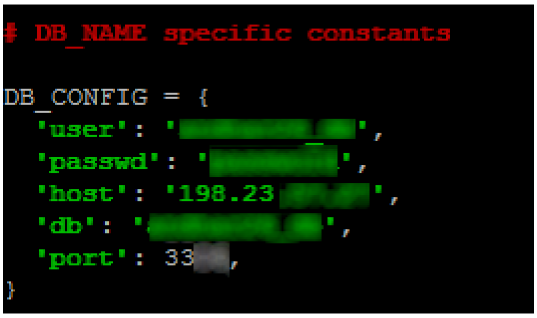
to install the library that allows Python to interact with MySQL databases.

Sending Data to Your Database

1) On the Pi, navigate to **/home/pi/pipsta/3_Web_Send**
2) In LXTerminal, enter:

```
nano WebSend.py
```

3) Locating the following DB_CONFIG block, now enter the you details you collected in the previous tutorial (including the password *for the database)*, e.g.:



```
# DB_NAME specific constants

DB_CONFIG = {
  'user': '[redacted]',
  'passwd': '[redacted]',
  'host': '198.23[redacted]',
  'db': '[redacted]',
  'port': 33[redacted],
}
```

4) Save and exit nano in the usual way, i.e. **[CTRL]+[X], [Y], [return]**
5) Use nano again to create a file called data.txt, in the same folder, with the contents:

Hello from Pipsta over the Web!

with a trailing carriage return.

6) Look on the rear of your printer to determine its serial number. It should start with a G, followed by several digits
7) In LXTerminal, enter:

```
python WebSend.py "<your serial number>" "data.txt"
```

8) If all is well, you should see the message "Print job created!" appear on the next line in LXTerminal
9) If all is *not* well, first check ALL of the details: user, password, host, db and port. Then check your internet connection is okay.

10) Once you have found the success message, go to FreeHostia, log-in, and in phpMyAdmin look at the *printdata* table by clicking on the *printdata* hyperlink:

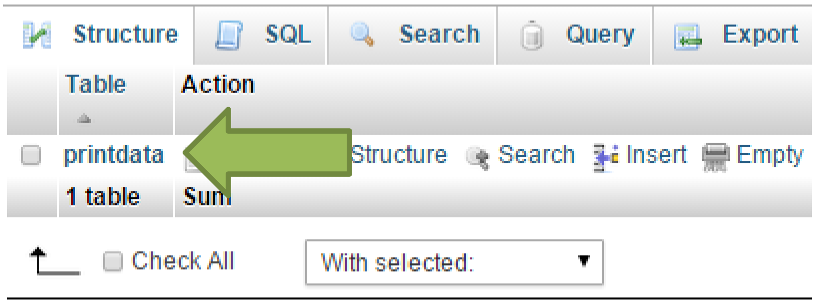
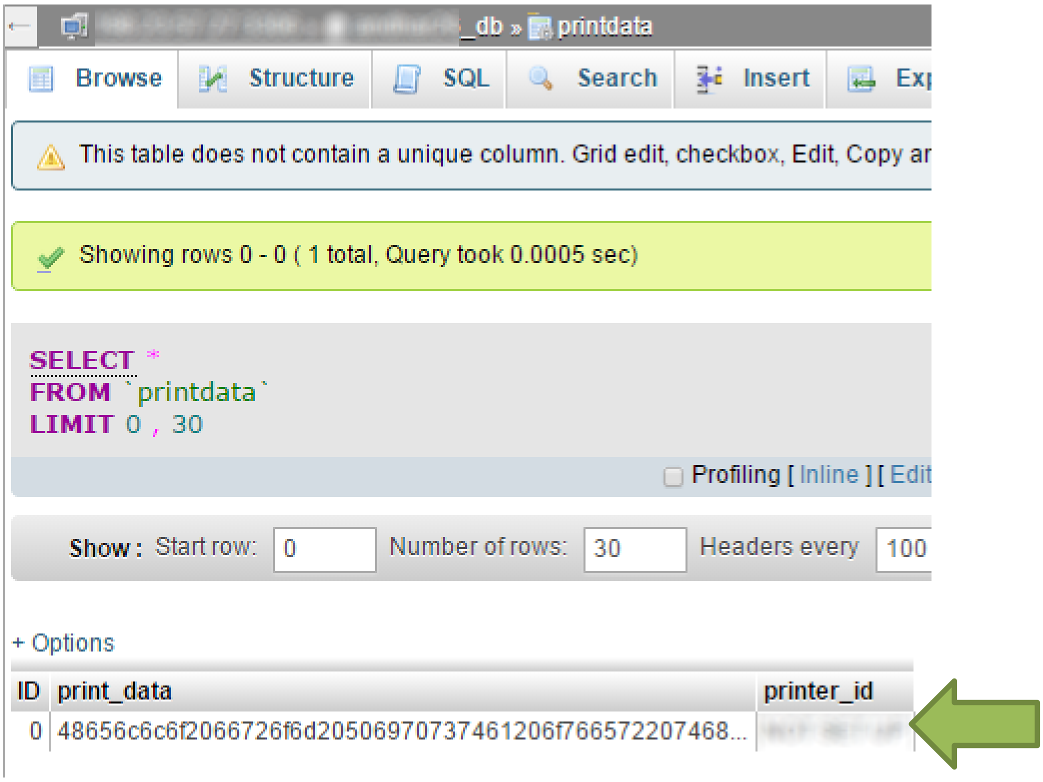


Table	Action
printdata	Structure Search Insert Empty
1 table	Sum

11) Within the table, you should now see a single line, as follows:



This table does not contain a unique column. Grid edit, checkbox, Edit, Copy ar

Showing rows 0 - 0 (1 total, Query took 0.0005 sec)

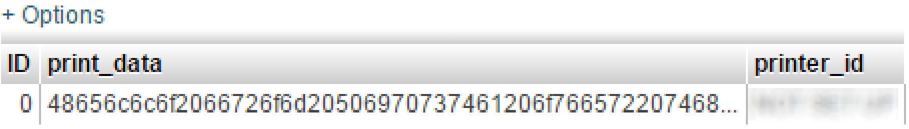
```
SELECT *
FROM `printdata`
LIMIT 0 , 30
```

Show : Start row: 0 Number of rows: 30 Headers every 100

ID	print_data	printer_id
0	48656c6c6f2066726f6d20506970737461206f766572207468...	[redacted]

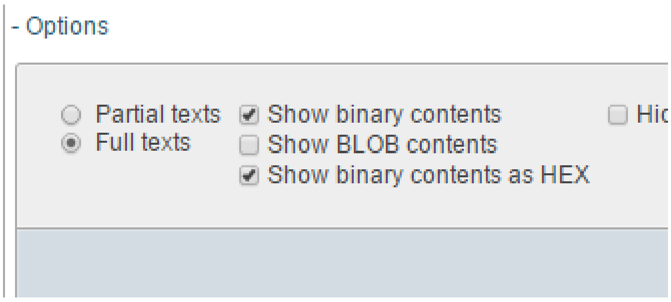
12) You should recognise your printer serial number under the **printer_id** field, however the **print_data** differs *in appearance* to the message you entered in *data.txt*. This is actually *ASCII encoded hex*, where pairs of characters represent individual characters. To confirm this for yourself:

➢ Click '+Options' above the record



ID	print_data	printer_id
0	48656c6c6f2066726f6d20506970737461206f766572207468...	[redacted]

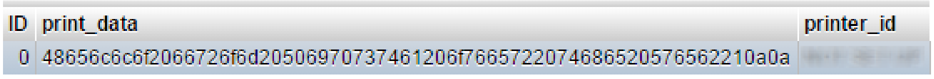
➢ Select 'Full texts' and click the 'Go' button:



- Options

○ Partial texts ☑ Show binary contents □ Hic
● Full texts □ Show BLOB contents
☑ Show binary contents as HEX

- ➢ You should now see a full line of ASCII encoded hex under **print_data:**

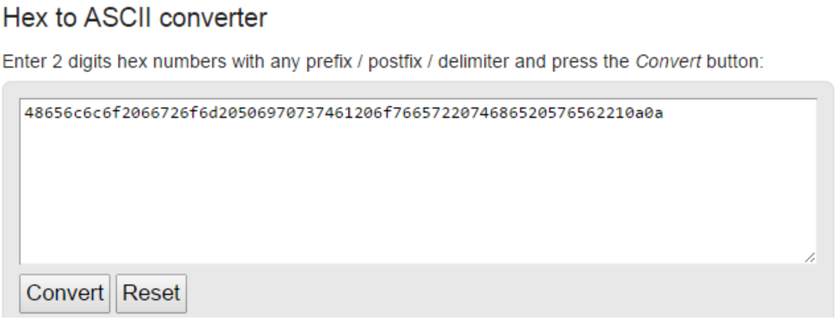


ID	print_data	printer_id
0	48656c6c6f2066726f6d20506970737461206f76657220746865205765622 10a0a	[redacted]

- ➢ Now, move the mouse pointer to the start of the data under **print_data** and click and drag the cursor over the full length of the data (from the first ‘4’ to the last ‘a’). Now copy this by pressing [CTRL][C].
- ➢ Now, in a browser, go to:

  http://www.rapidtables.com/convert/number/hex-to-ascii.htm

and paste the clipboard contents into the text field as shown below:



Hex to ASCII converter

Enter 2 digits hex numbers with any prefix / postfix / delimiter and press the *Convert* button:

```
48656c6c6f2066726f6d20506970737461206f76657220746865205765622 10a0a
```

- ➢ Now click convert and confirm that the data *is* the same as the contents of the *data.txt* file.

13) You have now successfully used **WebSend.py** to send data from your Raspberry Pi to a database on the web, and confirmed the contents of the new record created.
14) In the next tutorial, we will learn how the Raspberry Pi can retrieve this print job, print it, and mark it as printed so it does not print again.

Extending the Tutorial

You can:

- Adjust the text in *data.txt*,
- Create a new file and pass this filename as a command-line parameter in *WebSend.py*
- Obtain a friend's/colleagues serial number and send a message intended to be printed to their Pipsta!
- Discuss the necessity for using ASCII encoded hex to represent characters to access *byte values* like zero (NULL)

Shutting Pipsta Down Safely

Whilst the printer is resilient when it comes to powering down, the Raspberry Pi must undergo a strict shutdown process to avoid corrupting the Micro SD card. The most straightforward method of doing this is to double-click the 'Shutdown' icon on the desktop.

TIP:	If you are already in LXTerminal, type **sudo shutdown –h now** to shut-down the Raspberry Pi immediately.

TIP:	Always make sure ALL activity on the Raspberry Pi's green LED (the LED on the right) has stopped before removing the power!

■End of Document■