**WebIOPi - Raspberry Pi REST framework**

WebIOPi is a REST framework which allows to control Raspberry Pi’s GPIO from a browser. It’s written in Javascript for the client and in Python for the server.

You can fully customize and easy build your own webapp. You can even use all the power of WebIOPi directly in your own Python script and register your functions so you can call them from the webapp. WebIOPi also includes some other features like software PWM for all GPIO.

# Installation

Installation on the Raspberry Pi is really easy, as it only requires Python.

On Raspbian Wheezy, you can use the PiStore to download and install WebIOPi.

You can also install it using a terminal or a SSH connection, check the project page for the latest version, then:

$ wget http://webiopi.googlecode.com/files/WebIOPi-0.5.3.tar.gz

$ tar xvzf WebIOPi-0.5.3.tar.gz

$ cd WebIOPi-0.5.3

$ sudo ./setup.sh

You should see some compile and install process, to finally get a success output with usage instructions:

WebIOPi successfully installed

\* To start WebIOPi with python:   
sudo python -m webiopi

\* To start WebIOPi with python3:  
 sudo python3 -m webiopi

\* To start WebIOPi at boot:  
sudo update-rc.d webiopi defaults

\* To start WebIOPi service:  
sudo /etc/init.d/webiopi start

\* Look in /home/pi/webiopi/examples for Python library usage examples

You will have a line for each installed Python version you can use to launch WebIOPi.

It’s time to start WebIOPi, for example with Python 2.X:

$ sudo python -m webiopi

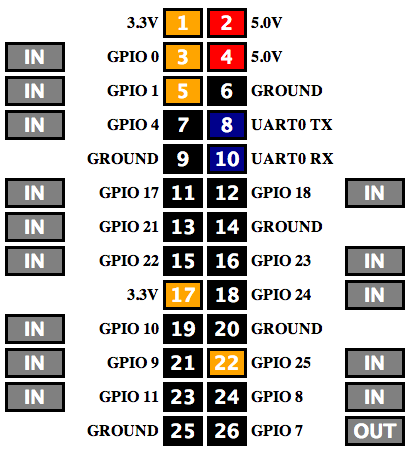
WebIOPi/Python2/0.5.3 Started at http://[IP]:8000/webiopi/

# First use

Open a browser from any device of your network, and point it to the given URL: http://raspberrypi:8000/webiopi/

Replace raspberrypi by its IP. You can use localhost if you are connected to your Pi with a keyboard and a display plugged in it.

You will be asked to log in, **default user is webiopi and password is raspberry**. You should see the default header app:



With the default header app, you can toggle GPIO functions between input and output, and toggle pin states. Just click on the IN/OUT buttons and on each pin to change their state when set as output.

All GPIO can be directly used with the REST API. For instance, to setup GPIO 23 as an output, just make a HTTP POST request on /GPIO/23/function/out then to output a logical 1, make POST on /GPIO/23/value/1

To retrieve states, make HTTP GET on /GPIO/23/function and /GPIO/23/value