

# Google Physical Web

Publish "hot deals" through the Google Physical Web

**The Physical Web**, a project from Google's Chrome team, is an open source approach to allow physical devices (e.g. a vending machine, a poster, a toy, a bus stop, a rental car) to broadcast URL around providing an access to information or remote control.

Google provides phone Apps that allow the user to discover the brocasted URL and use them to access web pages, javascript apps or nativ apps to get information about the device or remote control them.

Get more info about the physical web here:

<http://google.github.io/physical-web/>

- Download the Physical web Android app here  
[https://play.google.com/store/apps/details?id=physical\\_web.org.physicalweb](https://play.google.com/store/apps/details?id=physical_web.org.physicalweb)
- Download the Physical web from the Apple Store here  
<https://itunes.apple.com/fr/app/physical-web/id927653608?mt=8>

More on this **Google video**:



<http://www.youtube.com/watch?v=1yaLPRgtIR0>

**This tutorial** describes steps to publish a store "hot deals" as a picture on the Physical Web. Without any technical knowledge, you'll be able to use a Raspberry Pi to brodcast URLs around that can be seen with the nativ apps.

We'll use two steps:

- publish a picture on the Internet with **postimage.org**
- broadcast that picture URL with the Pi using the Physical Web bluetooth protocol.

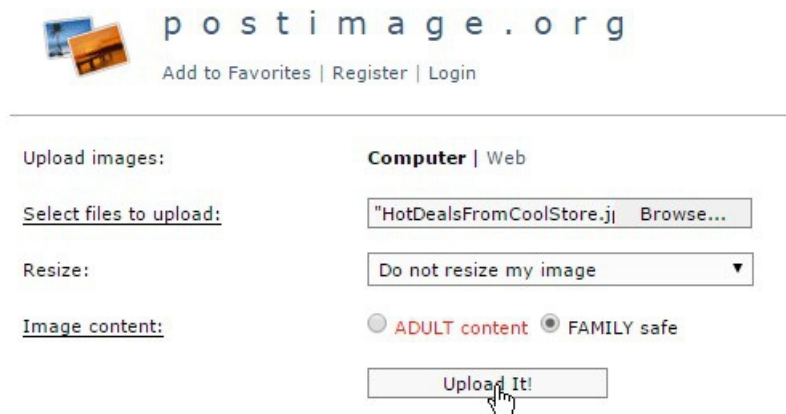
After that, any customer using the Physical web app on his phone will have access at the URL and then the picture, displaying the "Hot Deals" of your store.



Except for Pi3, you need a Bluetooth 4.0 dongle (you can buy it here <http://adafru.it/1327>) and it should be plugged in a USB port of the TimeSquAir

## Publish a "hot deals" picture on the internet

Your hot deals web page must be available through internet. There are several solutions and the chosen one here is **postimage.org**:



The screenshot shows the postimage.org website interface. At the top, there is a logo with a speech bubble and the text "postimage.org". Below the logo, there are links for "Add to Favorites", "Register", and "Login". The main content area is titled "Upload images:" and has a sub-header "Computer | Web". There are four rows of form elements: "Select files to upload:" with a text input field containing "HotDealsFromCoolStore.jl" and a "Browse..." button; "Resize:" with a dropdown menu set to "Do not resize my image"; "Image content:" with two radio buttons, "ADULT content" (unselected) and "FAMILY safe" (selected); and an "Upload It!" button at the bottom.

### Upload your hot deals image on postimage.

Copy the `Direct Link` of your hot deals. Also copy the given URLs to reuse them later if needed (as the deletion link).



**Import node**

---

Paste node name or node url here

---

Ok
Cancel
Reboot

This imports a third party software. The installation may take up to 5 minutes (a progress bar shows the progress), depending on what is already installed on your Pi.

At the end, click the `reboot` button to restart the Raspberry Pi.

## Step 8 : Build a flow.

Use the menu `Import / lib / PhysicalWeb demo 1` to add an already available flow

Here is the flow:



"Eddystone" is the name of the protocol used to broadcast the URL with Bluetooth.

## Enter the URL to broadcast

Double click the Eddystone node to enter the URL to broadcast:

**Edit eddystone-url node**

---

Name

URL

---

OK
Annuler

## Activate and broadcast your URL

Click the top-right "Activate" button.



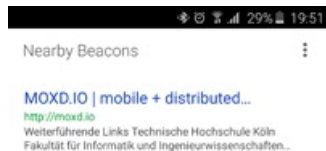
A "Stopped" status appears on the eddystone node to indicate that it's

actually not broadcasting.

Then, click the "Go" button on the flow and the status on the Eddystone node becomes "broadcasting".

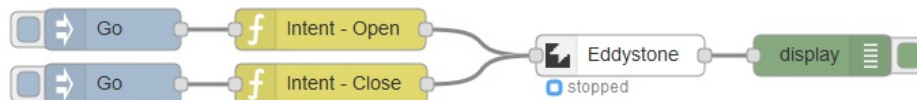
## Access your hot deals thanks to the Physical Web

Use the Physical Web application (available on **Android** and **iOS**) to browse the broadcasted URL and look at the hot deals picture.



## More

The second demo `PhysicalWeb demo 2` shows how to stop the broadcasting:



## Credits

This instructable uses the (great) work

- of **Matthias Böhmer** ( <https://github.com/matboehmer/node-red-contrib-eddystone> )
- and of **Don Coleman** ( <https://github.com/don/node-eddystone-beacon> )