# Overview

A sonos can be controlled for basic things from an ISY directly ([see wiki](http://wiki.universal-devices.com/index.php?title=ISY-994i_Series_INSTEON:Networking:Network_Resources#Sonos)). To get a sonos to stop playing, play an alert sound/message, and resume playing takes a bit more brains. There may be a different want to do it but this is a starting place to explore the options.

In this setup, there are 3-4 devices:

1. ISY
2. Sonos
3. Raspberry Pi (or similar device that can run python)
4. A device somewhere must be accessible by the sonos to download the alert/message/sound. This could be the raspberry pi running an http server (like nginx) or it could be a nas or pc (you will have to find the right syntax for the URL).

The ISY sends a request to the raspberry pi. The raspberry pi determines if music is already playing by querying the sonos. If it is, it stores the current song playing and position. It then tells the sonos where to download the mp3 file (the alert/message/sound). It then waits for the file to be done playing and then tells the sonos to resume playing its current song at the stored position.

# Tools used

[Soco](https://github.com/SoCo) – python class for controlling sonos devices

Flask – python class for a micro web service

Nginx – can be used as a lightweight web server to serve up static files

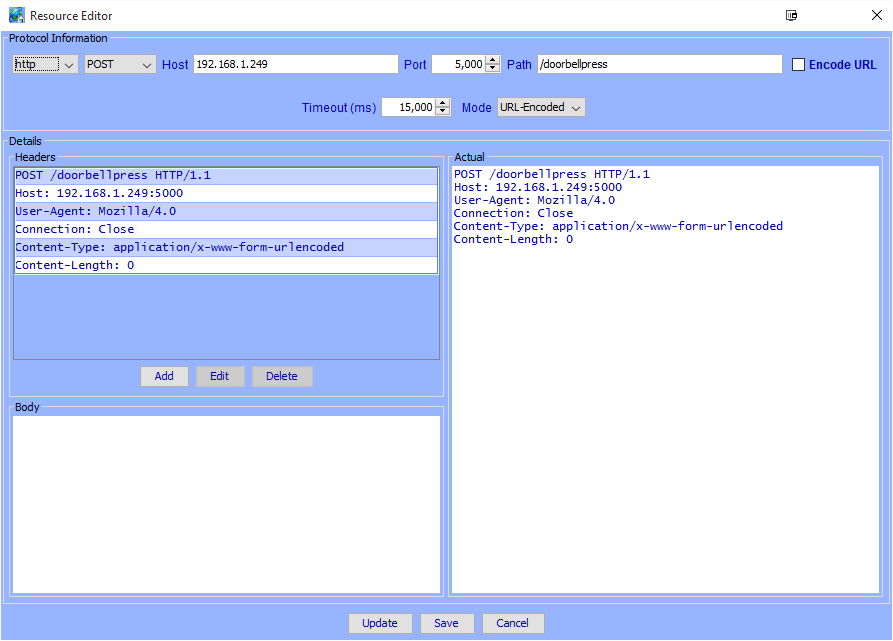
# Network Resource configuration

Host: the IP of the machine hosting the sonos-isy.py script (e.g. raspberry pi)

Port: Must match port in sonos-isy.ini file

Path: Must match path in sonos-isy.py file

Timeout: Must be set high to avoid timeout error on the ISY – the mp3 file takes a few seconds for the sonos to download a file and play it.



# Other footnotes

Here are some people discussing using SoCo with [text to speech](https://github.com/SoCo/SoCo/issues/99).

VictorOps github repo for their [version](https://github.com/charlie-vo/victorops-sonos).

[Why sonos can’t play local files when using SoCo](https://github.com/SoCo/SoCo/pull/307).