

Cyclic broadcasting on Dial Radio

<https://dialradio.live>

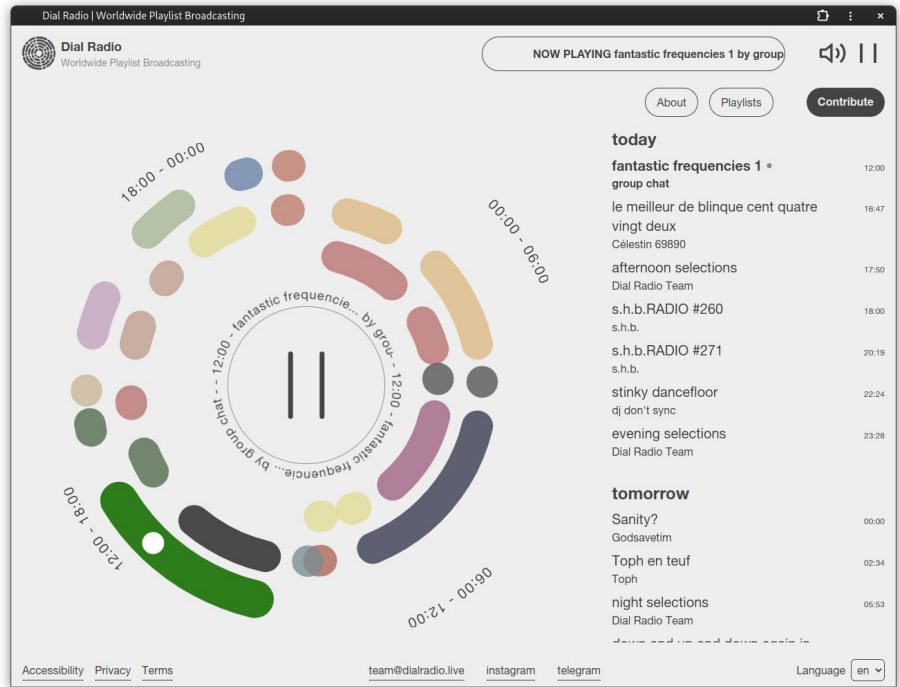
Pierre Depaz (@période)



Concept

Dial Radio broadcasts anybody's playlists!

- since June '23
- three people team
- 150+ playlists





About

Playlists

Contribute

Contribute a playlist!

? Additional information

☰ **what is your playlist called? ***

e.g. zumba café

🔗 **where can we find it? ***

Soundcloud | Spotify | Youtube | Deezer | Apple Music | Mixcloud
Playlists should be between 1h and 6h.

e.g. <https://soundcloud.com/you/the-playlist>

🕒 **when's the best time to play it? ***



morning

afternoon

evening

night

📄 **anything to say about it?**

e.g. travel playlist. credits go to the landscapes

🖼️ **what's a fitting image?**

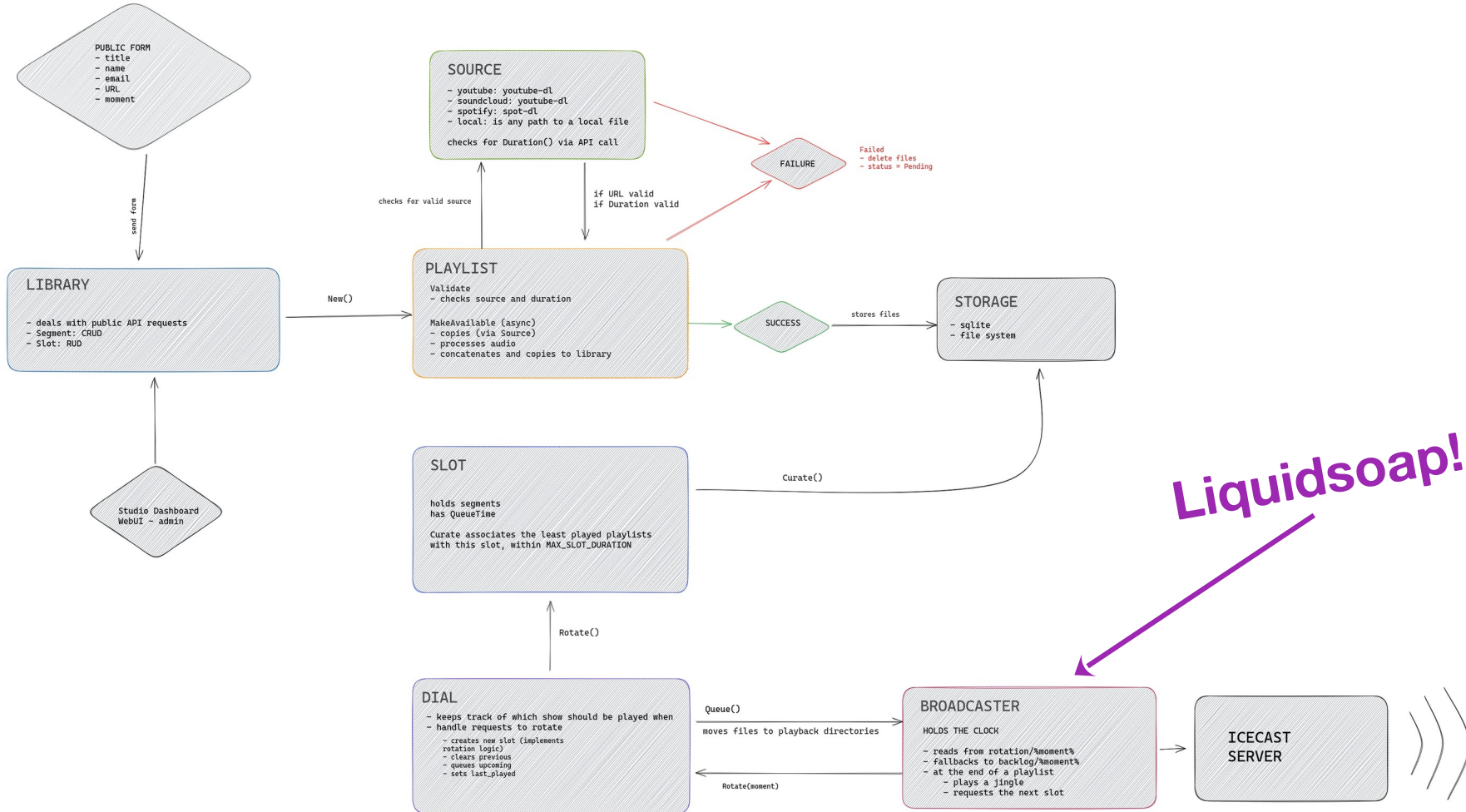
Click to select a cover image

Concept

Why did we want to do this?

- Lowering barrier to access
- Carrier bag theory of listening ([Le Guin, 1986](#))
- Radio as a time-based medium

Architecture



Architecture

A big switch for the different moments.

Slots last six hours, are made up of:

- listener-contributed playlists
- automatic playlists

```
radio =  
  switch(  
    track_sensitive=false,  
    [  
      (morning_jingle, delay(180., jingles)),  
      (morning_slot, morning),  
      (afternoon_jingle, delay(180., jingles)),  
      (afternoon_slot, afternoon),  
      (evening_jingle, delay(180., jingles)),  
      (evening_slot, evening),  
      (night_jingle, delay(180., jingles)),  
      (night_slot, night),  
      ({true}, fallback_track)  
    ]  
  )
```

Architecture

Liquidsoap acts as the main clock:

- picks up whichever sound files are in watched directories
- calls the API at the end of each part of the day
- the API moves files around accordingly

```
def request_rotate() =
  if (!is_shutting_down == false and list.mem(!moment, moments))
  then
    log("rotating for: #{!moment}")
    req =
      http.post(
        headers=[
          ("Content-Type", "application/x-www-form-urlencoded"),
          ("Authorization", "Bearer #{api_auth_key}")
        ],
        data="moment=#{!moment}",
        "localhost:#{station_port}/dial/rotate"
      )

    if
      req.status_code >= 400
    then
      log.important("failed to request rotation")
    end

    # wait 10 minutes and then skip to next track
    if
      !moment == "morning"
    then
      thread.run(delay=600., {morning.skip()})
    elsif !moment == "afternoon" then thread.run(delay=600., {afternoon.skip()})
    elsif !moment == "evening" then thread.run(delay=600., {evening.skip()})
    elsif !moment == "night" then thread.run(delay=600., {night.skip()})
    end
  else
    log("no valid moment set from last_metadata() #{!moment}, skipping...")
  end
end
```

Architecture

When the listener playlists are done, LS generates the tracklist of filler playlists.

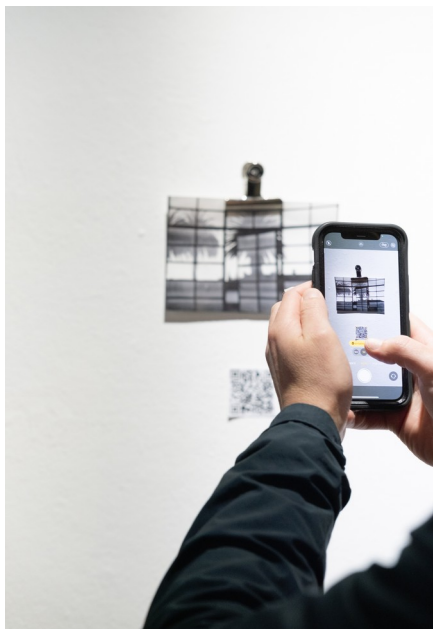
```
def signal_selection(m) =
  log("signalling selection...")
  if
    (!moment == "" or not list.mem(!moment, moments))
  then
    log.important("invalid moment!")
  else
    log(metadata.json.stringify(m))

    req =
      http.post(
        headers=[
          ("Content-Type", "application/json"),
          ("Authorization", "Bearer #{api_auth_key}")
        ],
        data=metadata.json.stringify(m),
        "localhost:#{station_port}/playlists/selections/#{!moment}"
      )

    if
      req.status_code >= 400
    then
      log.important("failed to signal selection")
    end
  end
end
```


Events!

Liquidsoap for ultra-local broadcasting



Feedback

- + Simple and reliable!
- + Community is 🔥
- Online documentation is hard to navigate

Feedback

- AAC was a bit of a struggle :/
- Still figuring out smooth deploys
- Setting a testing environment takes a while and seems very bespoke



Feedback

My mental model of how LS handles event (listeners and handlers) is not there yet.

- Jingles overlap (`append + rotate()` ?)
- `fallback.skip` has a strange interaction with crossfade?
- Sending requests `on_end` of track vs. `on_leave` of source?

Thank you!

dialradio.live/contribute

gitlab.com/dial-radio

team@dialradio.live

