



Happy Birthday
QueensJS!



Taking the party online.

8.5.20 — QueensJS

IDEO

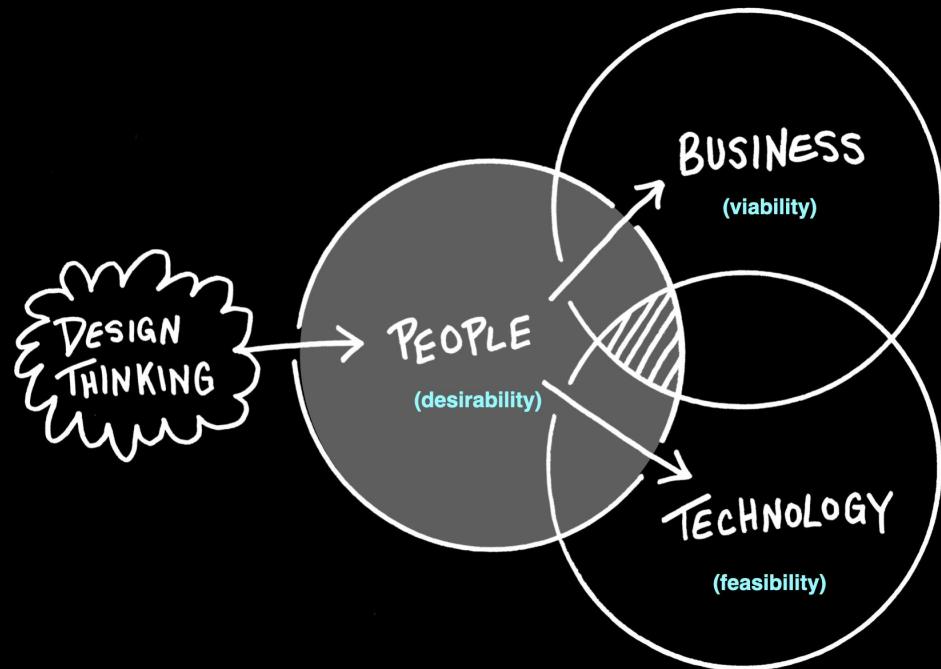






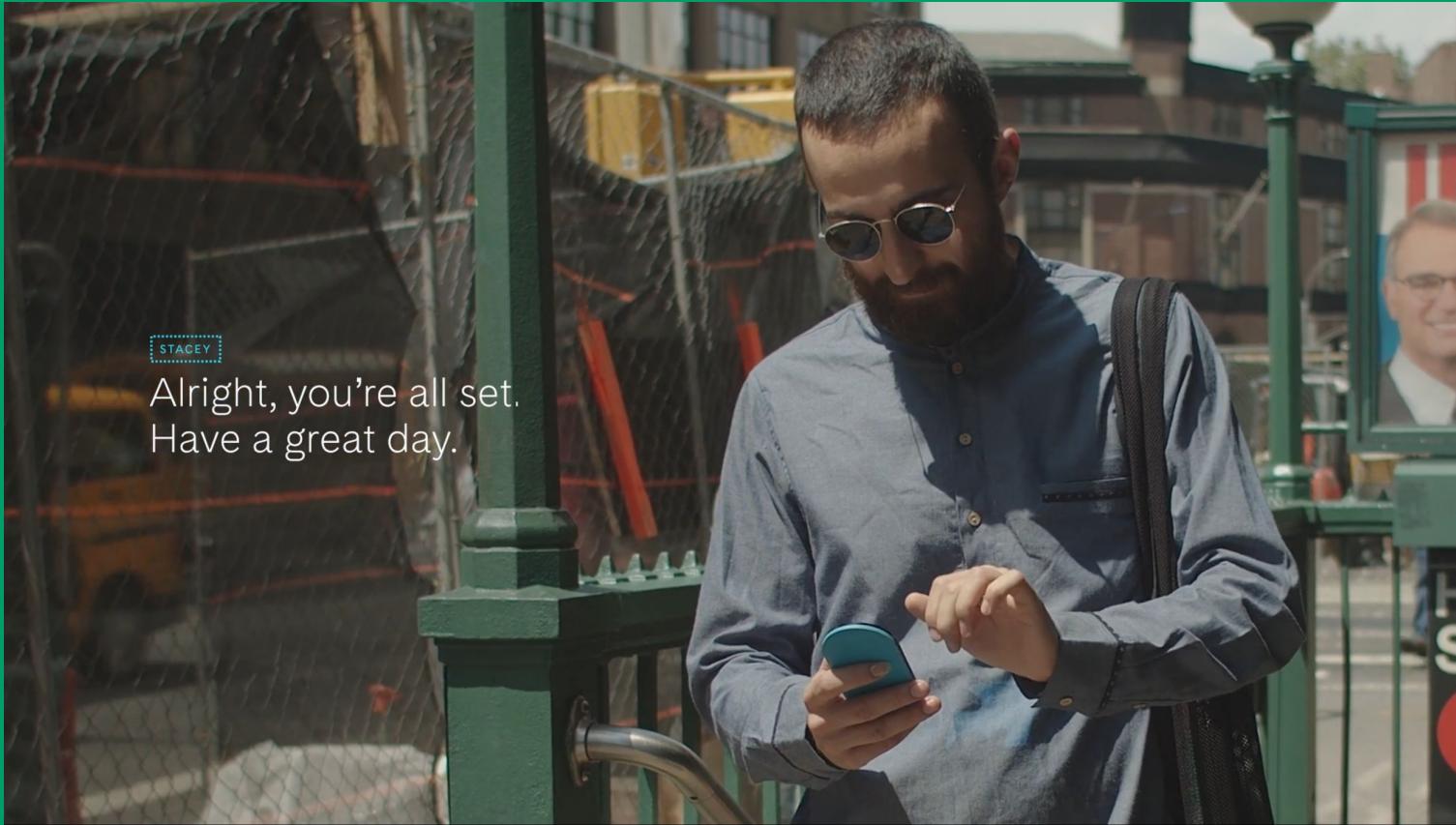






STACEY

Alright, you're all set.
Have a great day.









Reimagining the '70s Tehran Music Scene, One Party at a Time



What Now?

Candidate 1: WebRTC

<https://janus.conf.meetecho.com/>

About

Janus: the general purpose WebRTC server

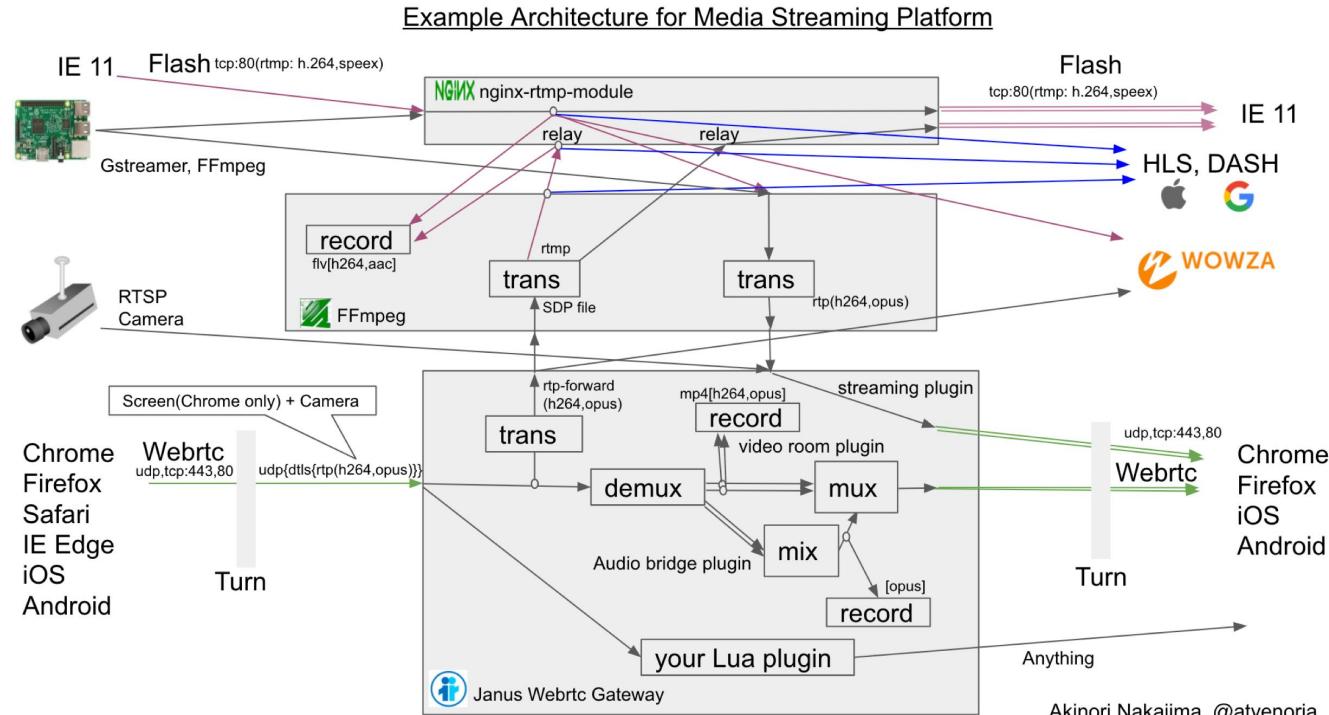
Janus is a WebRTC Server developed by [Meetecho](#) conceived to be a general purpose one. As such, it doesn't provide any functionality per se other than implementing the means to set up a WebRTC media communication with a browser, exchanging JSON messages with it, and relaying RTP/RTCP and messages between browsers and the server-side application logic they're attached to. Any specific feature/application is provided by server side plugins, that browsers can then contact via Janus to take advantage of the functionality they provide. Examples of such plugins can be implementations of applications like echo tests, conference bridges, media recorders, SIP gateways and the like.

The reason for this is simple: we wanted something that would have a [small footprint](#) (hence a C implementation) and that we could only equip with what was [really needed](#) (hence pluggable modules). That is, something that would allow us to deploy either a full-fledged WebRTC gateway on the cloud, or a small nettop/box to handle a specific use case.



Check the [Documentation](#) for more details about Janus, or check out the [Demos](#) to see it in action.

<https://github.com/atyenoria/janus-webrtc-gateway-docker>



Candidate 2: RTMP

<https://github.com/illuspas/Node-Media-Server>

illuspas / Node-Media-Server

Code Issues Pull requests Actions Projects Wiki Security Insights

master 6 branches 61 tags Go to file Add file Code

A Node.js implementation of RTMP/HTTP-FLV/WS-FLV/HLS/DASH/MP4 Media Server

www.npmjs.com/package/node-me...

nodejs flv rtmp websocket livestream media-server hls mpeg-dash

Readme MIT License

Releases 61 tags

Used by 2,076

Contributors 13 + 2 contributors

Languages

JavaScript 98.6% HTML 1.3%

File	Description	Last Commit
api	fc	20 days ago
public	移除web根目录的文件	14 months ago
.dockerignore	add Dockerfile	2 years ago
.eslintrc.js	Add eslint support	20 days ago
.gitignore	Add eslint support	20 days ago
Dockerfile	add Dockerfile	2 years ago
LICENSE	Update LICENSE	20 days ago
README.md	Update README.md	20 days ago
README_CN.md	u	20 days ago
app.js	u	20 days ago
certificate.pem	Add https and wss support	3 years ago
certrequest.csr	Add https and wss support	3 years ago
node_core_amf.js	Add eslint support	20 days ago
node_core_av.js	u	2 years ago
node_core_bitop.js	Add eslint support	20 days ago
node_core_ctx.js	f	2 years ago
node_core_logger.js	Implement the multicore cluster mode.	2 years ago
node_core_utils.js	Add ffmpeg version check	2 years ago
node_fission_server.js	f	20 days ago
node_fission_session.js	f	20 days ago
node_flv_session.js	Fix #173 "Cannot read property 'cork' of undefined"	17 months ago



Getting Started

Introduction

- What is Strapi?
- Features
- The Strapi Ecosystem
- Installation
- Deployment
- Contributing
- Troubleshooting
- Telemetry
- Quick Start Guide

Content API

Concepts

Guides

Admin Panel

Plugins

Local Plugins

Command Line Interface

Global strapi

Resources

- Contributing guide
- Migrations guides

Welcome to the Strapi documentation!

Strapi is the open-source [Headless CMS](#) developers love.

What is Strapi?

Strapi is a flexible, open-source Headless CMS that gives developers the freedom to choose their favorite tools and frameworks while also allowing editors to easily manage and distribute their content. By making the admin panel and API extensible through a plugin system, Strapi enables the world's largest companies to accelerate content delivery while building beautiful digital experiences.

Where does the name come from?

Strapi comes from the word Bootstrap, and helps Bootstrap your API. This was the original purpose of this project.

Features

Custom Content Structure

You can generate the admin panel in a few clicks and get your whole CMS setup in a few minutes.

Easy Content Management

Strapi's admin panel gives you an intuitive interface to create, edit and delete your content.

Developer-Friendly API

If your content has to be displayed to the world, an API is required. Strapi provides you with an API that will easily match your needs. Fetch any data you might need via a REST API or GraphQL endpoint.

 discotehranEN 

COLLECTION TYPES 

- Artists
- Donations
- Episodes
- Merchandises
- Minecraft Club Admissions

SINGLE TYPES 

- CT About
- CT Site Preferences
- Code of conduct
- DT About
- DT Club

Mani Nilchiani

[Delete](#)[Reset](#)[Save](#)

Name

Instagram

[Configure the view](#)

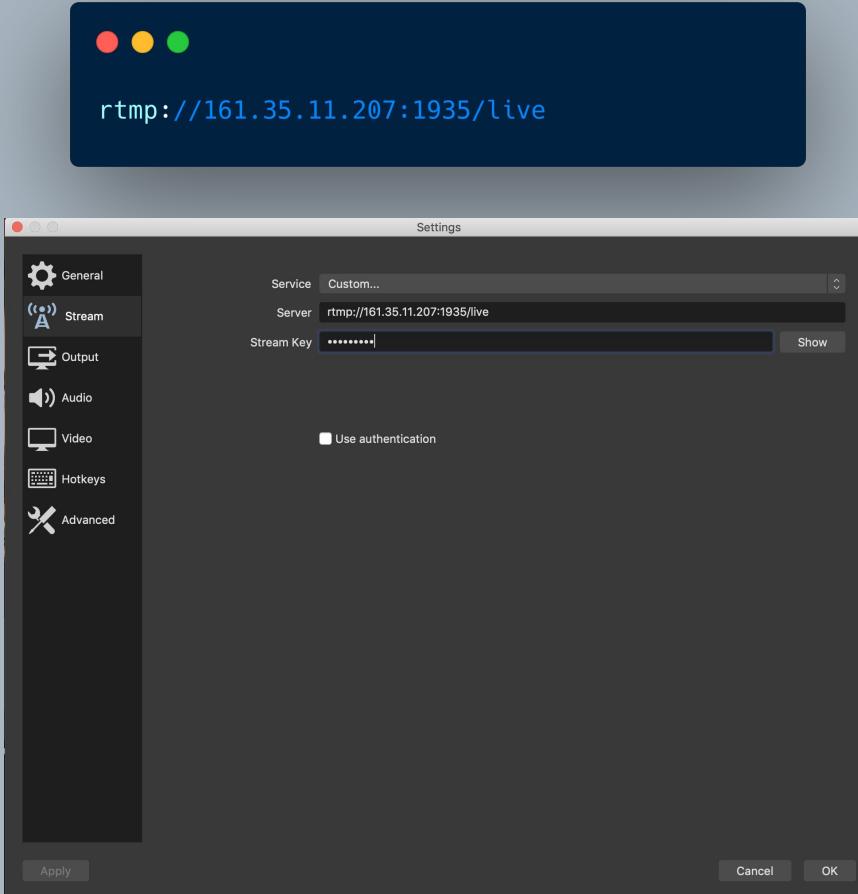
StreamingKey





```
const shortid = require('shortid');

// Before creating a value.
// Fired before an `insert` query.
beforeCreate: async (model) => {
  model.set('streamingKey', shortid.generate());
}
```



```
/*
  Setup Node Media Server
 */
const NodeMediaServer = require('node-media-server');
const nms = new NodeMediaServer(strapi.config.currentEnvironment.rtmp_server);
```

```
publicRuntimeConfig: {
  dev: {
    HOST: 'http://127.0.0.1',
    BASE_URL: 'http://127.0.0.1:1337',
    RTMP_SERVER: {
      http: {
        host: 'http://127.0.0.1',
        port: 8888,
        allow_origin: '*'
      }
    },
    prod: {
      HOST: 'https://discotehran.nyc',
      BASE_URL: 'https://api.discotehran.nyc',
      RTMP_SERVER: {
        https: {
          host: 'https://nms.discotehran.nyc',
          allow_origin: '*',
          port: 443,
          key:'[redacted]',
          cert:'[[redacted]]',
        }
      }
    }
}
```

```
nms.on('prePublish', async (id, StreamPath, args) => {
  const streamingKey = getStreamKeyFromStreamPath(StreamPath);

  let artist = null
  try {
    artist = await strapi.services.artist.findOne({ streamingKey });
    if (!artist) {
      const session = nms.getSession(id);
      session.reject();
    }
  } catch (e) {
    console.log('Failed to find artist with stream key ', streamingKey, ' : ', e);
  }
});
```

```
import { initFirebase } from '../lib.firebaseio'

const { database } = initFirebase()
const chat = database.ref(FIREBASE_REALTIME_CHAT)
const live = database.ref(FIREBASE_REALTIME_LIVE)

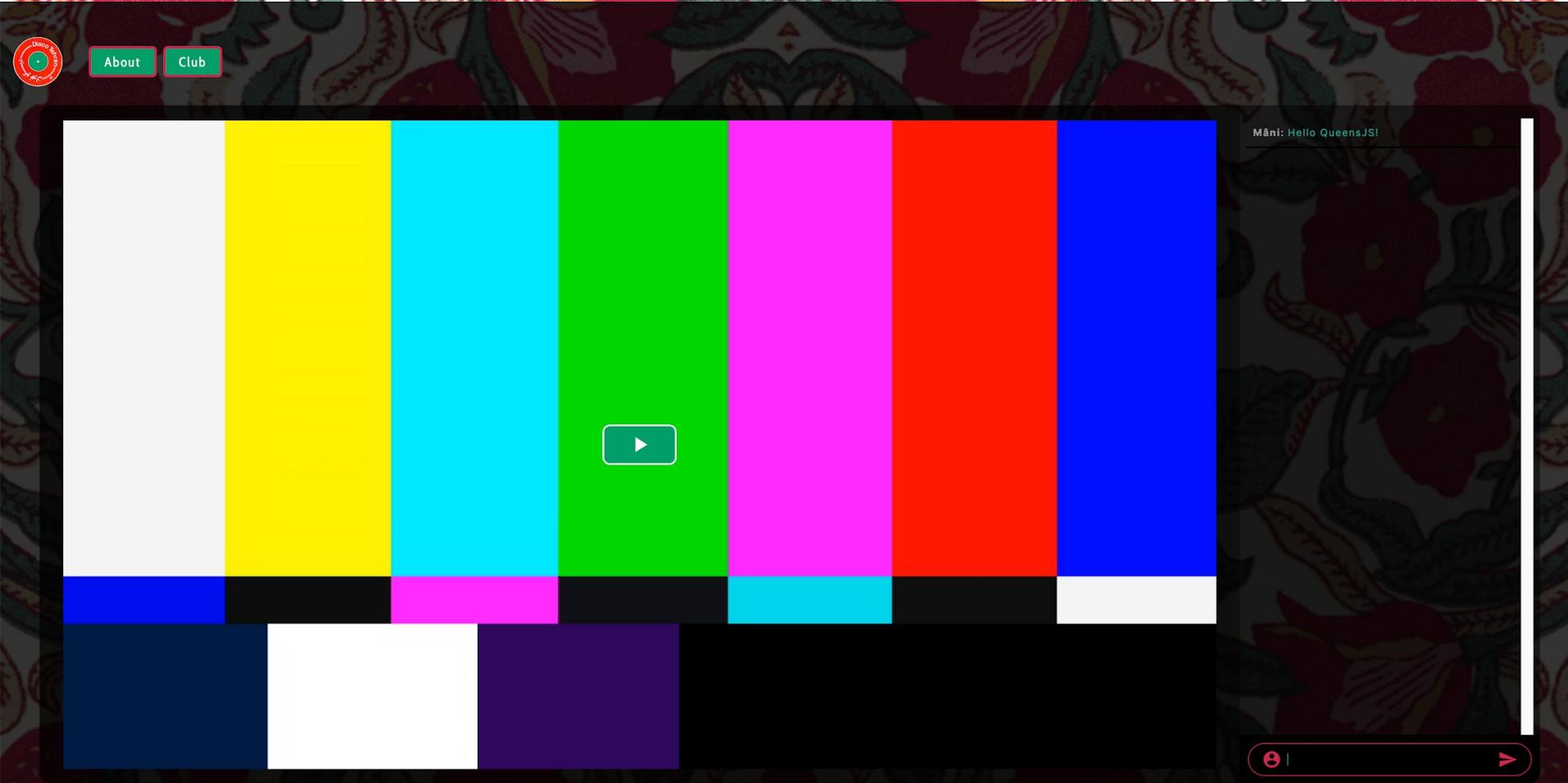
subscribeToLive = () => {
  this.live.on('value', live => {
    const data = live.val()
    if (!data) {
      const { onAir, activeStreamingKey, bannerText } = data

      if (this.state.bannerText !== bannerText) {
        this.setState({ bannerText })
      }

      if (this.state.onAir !== onAir) {
        this.setState({ onAir }, () => {
          const { onAir, activeStreamingKey } = this.state
          if (onAir) {
            if (activeStreamingKey) {
              this.updateVideoPlayerSource([this.generateSourceObject(activeStreamingKey)]).assets/pressplay.jpg")
            } else {
              this.updateVideoPlayerSource(this.defaultVideoJsOptions.sources)
            }
          } else {
            this.updateVideoPlayerSource(this.defaultVideoJsOptions.sources)
          }
        })
      }

      if (this.state.activeStreamingKey !== activeStreamingKey) {
        this.setState({ activeStreamingKey }, () => {
          const { activeStreamingKey, onAir } = this.state
          if (activeStreamingKey) {
            if (onAir) {
              this.updateVideoPlayerSource([this.generateSourceObject(activeStreamingKey)])
            } else {
              this.updateVideoPlayerSource(this.defaultVideoJsOptions.sources)
            }
          } else {
            this.updateVideoPlayerSource(this.defaultVideoJsOptions.sources)
          }
        })
      }
    }
  })
}
```

```
● ● ●  
#EXTM3U  
#EXT-X-VERSION:3  
#EXT-X-MEDIA-SEQUENCE:0  
#EXT-X-ALLOW-CACHE:YES  
#EXT-X-TARGETDURATION:11  
#EXTINF:5.215111,  
00000.ts  
#EXTINF:10.344822,  
00001.ts  
#EXTINF:10.344822,  
00002.ts  
#EXTINF:9.310344,  
00003.ts  
#EXTINF:10.344822,  
00004.ts  
...  
#EXT-X-ENDLIST
```



• OFFLINE

Tehran ❤️

iSalam family! Welcome to Disco Tehran ❤️

iSalam family! Welcome to Disco Tehran ❤️

iSalam family! Welcome to Disco Tehran

Donate



🔗 https://funkywarmdata.firebaseio.com/



funkywarmdata

chat

-ME-leRgfS5U9lIT69Va

- color: "#25a18eff"
- message: "Hello QueensJS!"
- type: "message"
- userID: "Mâni"

live

- bannerText: ";Salam family! Welcome to Disco Tehran ❤"
- onAir: false

COLLECTION TYPES

- Artists
- Donations
- Episodes
- Merchandises
- Minecraft Club

SINGLE TYPES

- CT About
- CT Site Preferences
- Code of conduct
- DT About
- DT Club

LIVE STREAMS

- ≡ Switch Board

PLUGINS

- ✍ Content-Types Builder
- 📁 Files Upload
- 👤 Roles & Permissions

GENERAL

- 🛒 Marketplace
- ≡ Plugins
- 📄 Documentation
- ❓ Help

Strapi v3.0.0-beta.19.5

Live Stream switch board

Set streaming status, preview and control streams.

1. Set "On Air" status

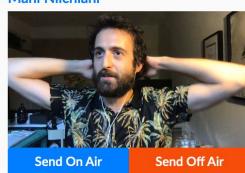
Setting this status will set the "On Air" flashing red light on the website's bottom Live Banner.
Audience will see a TV testing pattern placeholder video until you send a Streamer on air.

OFF ON

2. Send a streamer On Air

To send a streamer On or Off Air, use the buttons below the following preview video players.
Please note that this will instantly change the stream source for all of our viewers.

Mani Nilchiani



3. Update the Live Banner text

Type a message below and press "update" to instantly update the live banner text for the audience.

When going offline, you can update the text to something like:

"Currently offline. Please find our next show schedule @discotehran.nyc on Instagram."

Or just leave it empty.

:Salam family! Welcome to Disco Tehran ❤

Update



```
● ● ●

async function getStreams() {
  const streamingKeys = await fetch(`#${LIVESTREAM_BASE_PATH[env]}/apistreams`)
    .then(response => response.json())
    .then(streams => {
      let keys = []
      if (streams.hasOwnProperty('live')) {
        keys = Object.keys(streams.live)
      }
      return keys
    }).catch(e => {
      console.log('Failed to get streams ', e)
    })
}

const promises = streamingKeys.map(key => {
  return fetch(`${BASE_URL}/artists?streamingKey=${key}`)
    .then(response => response.json())
    .then(([artist]) => ({ name: artist.name, key: key }))
    .catch(e => {
      console.log('Failed to get artist details by streamingKey ', e)
    })
})

Promise.all(promises).then(_streamingArtists => {
  setStreamingArtists(_streamingArtists)
}).catch(e => {
  console.log('Failed to get streaming artists ', e)
})
}
```



Live Performance at
discotehran.nyc

Thank you!
Mani Nilchiani 8.5.20