



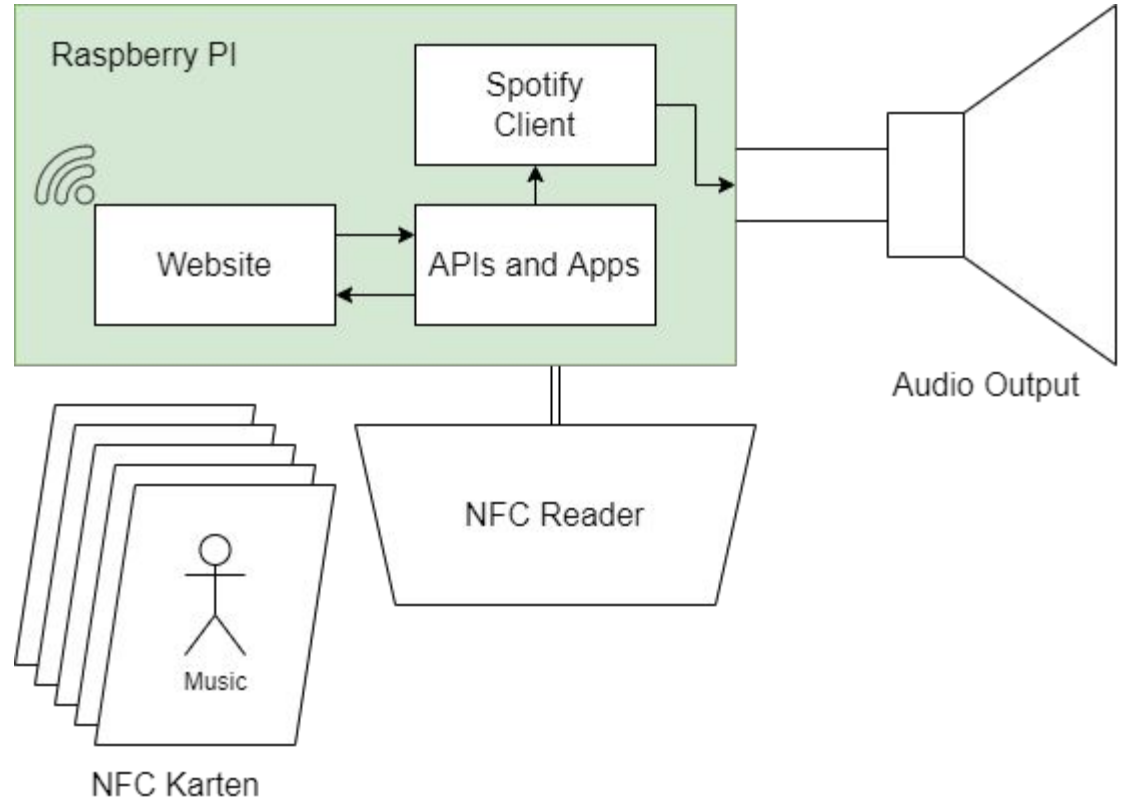
# Digital Record Player

Paul Bachinger, Maurer Florian, Kejda Domi, Hangoebel Justin



# Project Idea

- Raspberry PI
- RFID
- ExpressJS
- React
- Spotify API
- Apache Kafka



# RaspberryPI and MFRC522

- Input via NFC Cards
- Python Script Reads the Song ID
- HTTP request to ExpressJS server

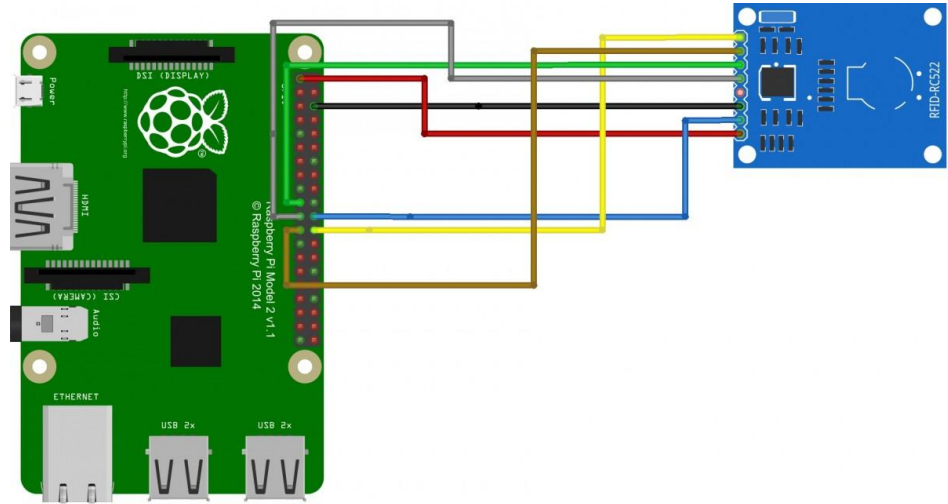
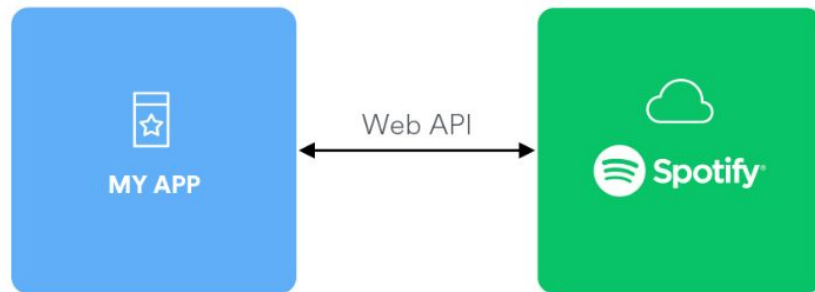


Image source: [www.fritzing.com/MFRC522](http://www.fritzing.com/MFRC522)

# Spotify Web API

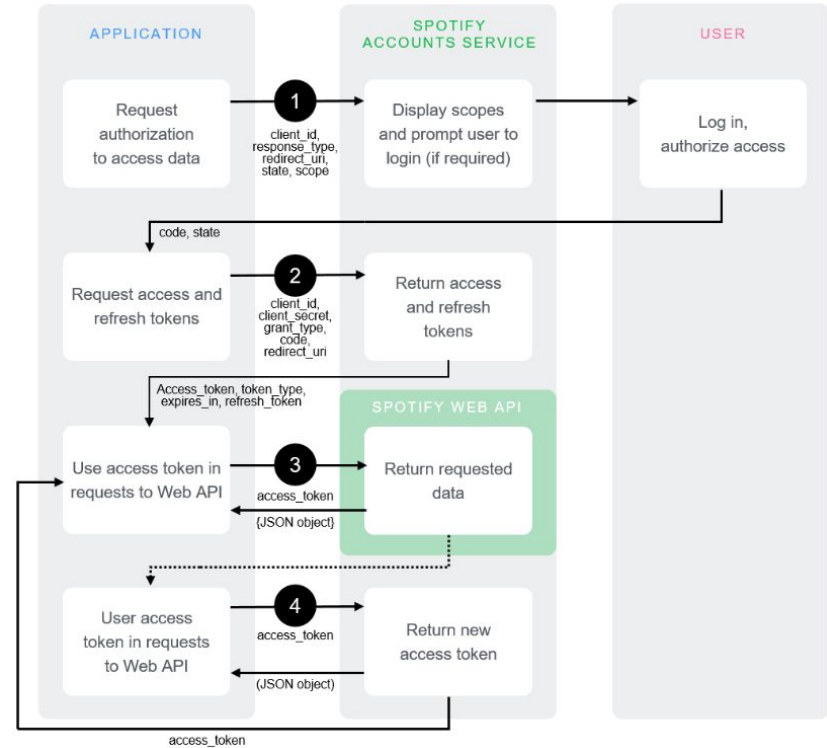


- REST API principle
- Gather public data from the Spotify database
  - Track-, Artist, Album information, ...
- Access private data from a specific Spotify account
  - Control playback, Edit Playlists, ...
  - Requires authorization via OAuth 2.0

The base address of the API is “<https://api.spotify.com/>”

# OAuth 2.0 - Authorization Flow

- Granting applications a variety of permissions
- Spotify Integration Module
  - Client credentials
- Used to call specific API endpoints



# ExpressJS

# express

- HTTP Server
- GET & POST Request
- Backend for Client and RFID Reader
- Interface to the Spotify API, handles Client Authentication

# Request to Spotify API

- PUT Requests to change Data with Body
- GET Requests to access Data
- POST Requests to change Data without a Body

## Request

---

**PUT** /me/player/play

### ✓ Query

**device\_id** string

The id of the device this command is targeting. If not supplied, the user's currently active device is the target.

Example value: "0d1841b0976bae2a3a310dd74c0f3df354899bc8"

### ✓ Body

application/json

**context\_uri** string

string

**uris** array of strings

Array of URIs

**offset** object

object

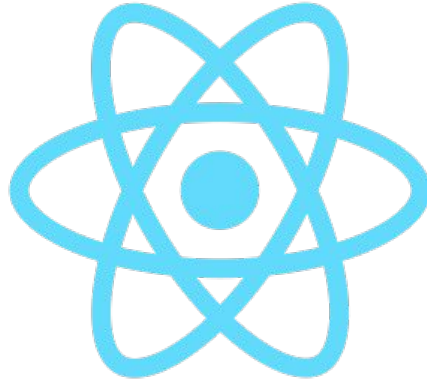
AdditionalProperties

**position\_ms** integer

integer

# Frontend

**HTML**



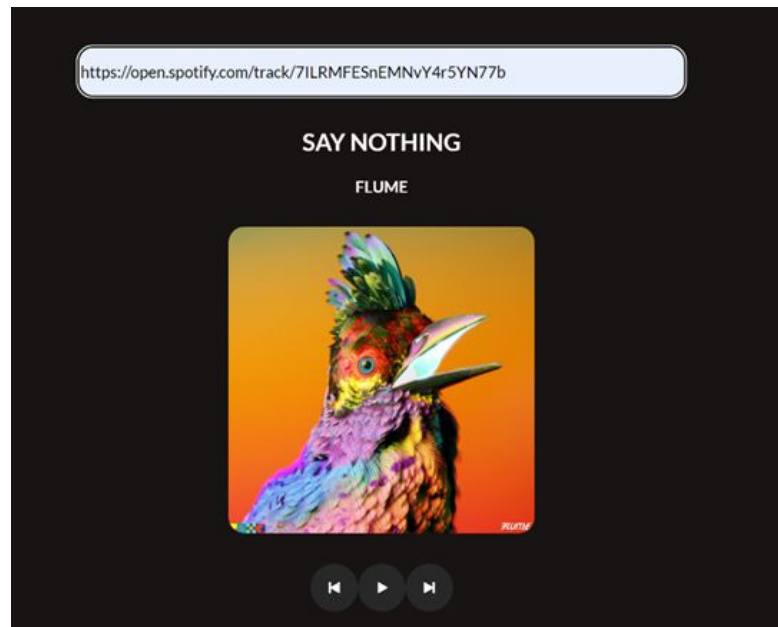
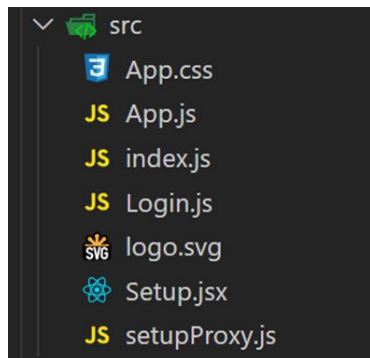
React JS

**CSS**





# React



# HTML

```
return (  
  <div className="grid">  
    <div className="search">  
      <form>  
        <label htmlFor="searchbar"></label>  
        <input className="corner-round-1" type="text" name="search" id="searchbar"  
          onChange={(value) => this.getSearchResults(value.target.value)}>  
        </input>  
      </form>  
    </div>  
  </div>  
)
```

# CSS

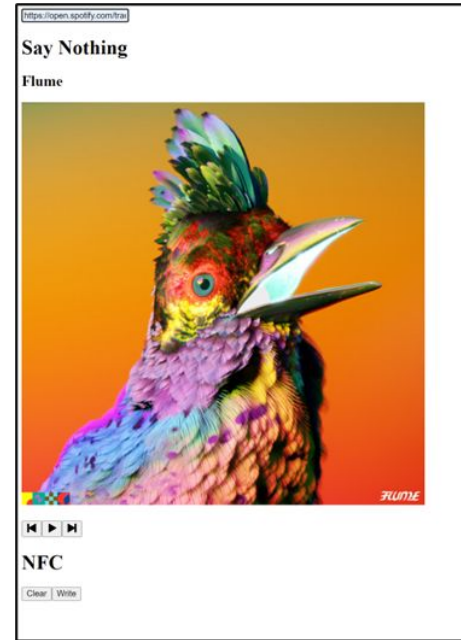
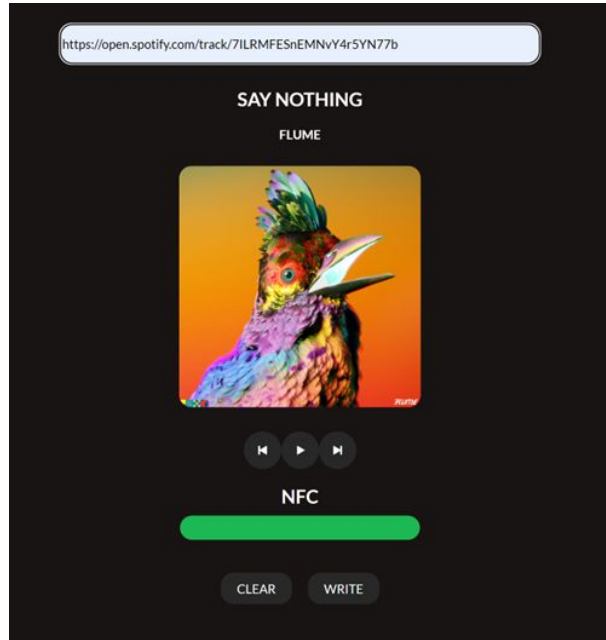
## Content in the stylesheet

```
/******ELEMENTS*****/  
html, body {  
  background-color: var(--spotify-black);  
  margin: 0;  
}  
/******CLASSES*****/  
.corner-round-1{  
  border-radius: 1rem;  
}
```

## Content in the HTML

```
<button class="button-style" type="button"></button>
```

# Finished Frontend



# Apache Kafka

- Python Thread listens to Kafka Topic
- On write Button press
- Server writes the ID to the Topic
- Python writes the Song ID

