

<https://rapidapi.com/>

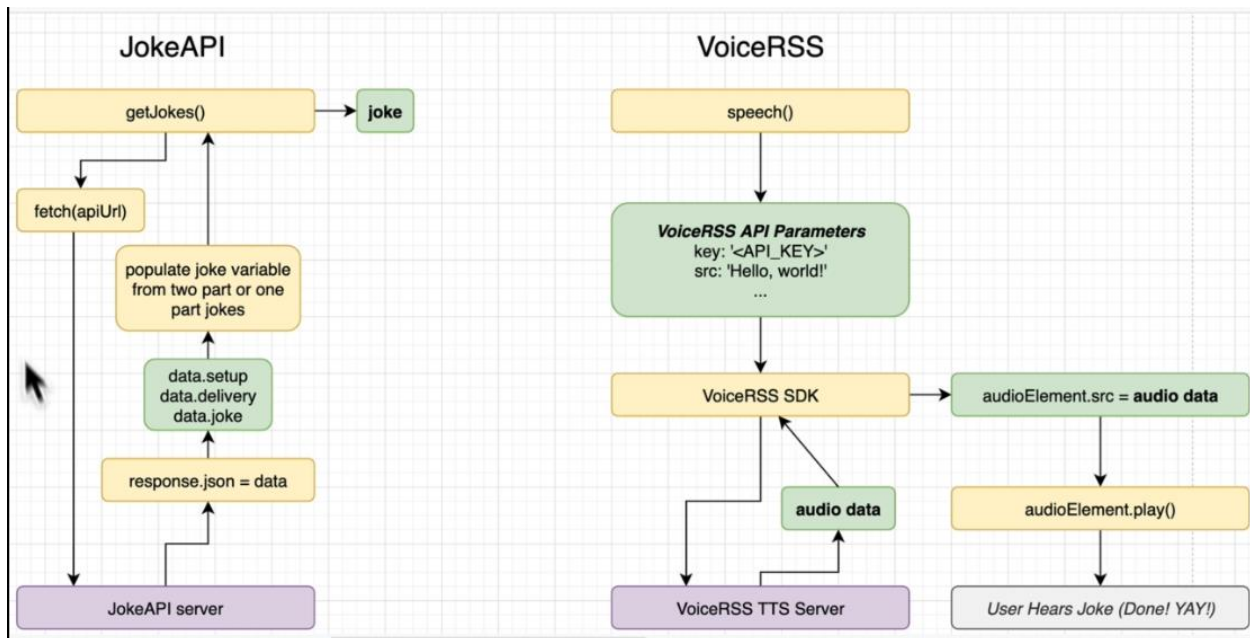
Raid API = text-to-speech

<http://www.voicerss.org/personel/> == API key: dcc18cda0fa34ee1bd1689e0a3e857c4

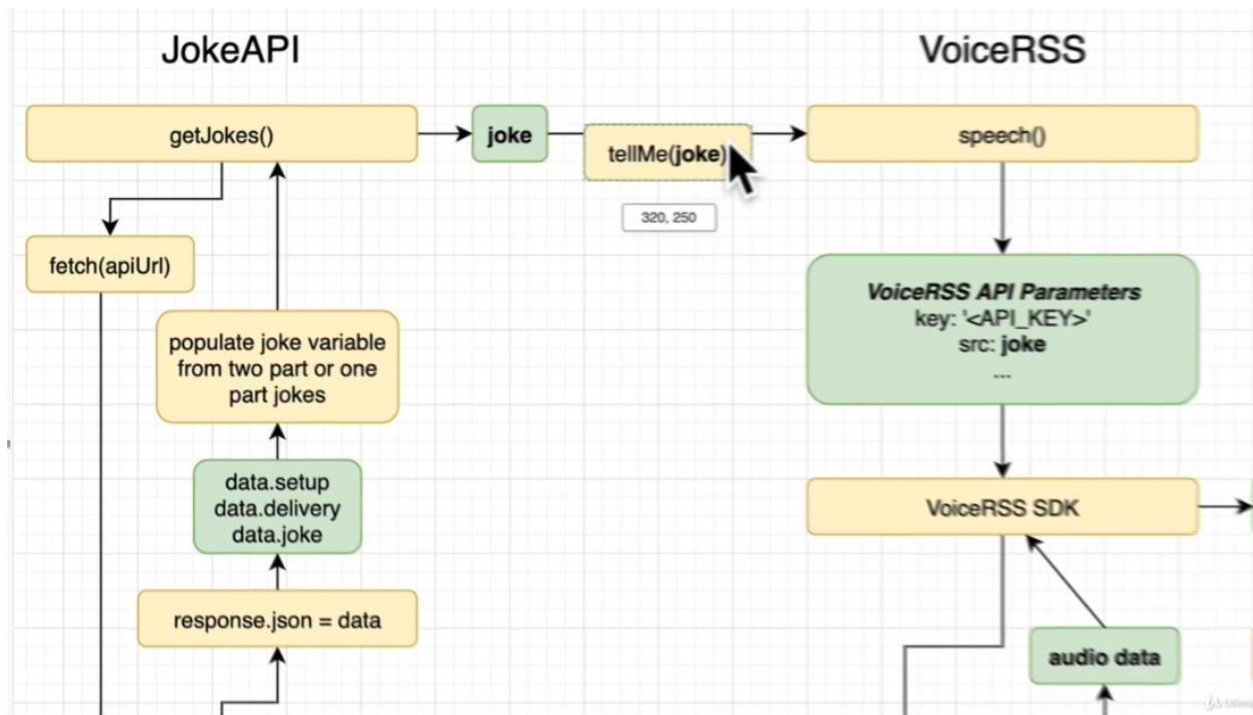
rapidapi.com a voicerss.org er API Key boshabo.

Documentation pora ekta big part as a developer

Joke API: <https://sv443.net/jokeapi/v2/> == URL: <https://v2.jokeapi.dev/joke/Any>



Connect jokeAPI & speech API



[ended](#)

Returns whether the playback of the audio/video has ended or not

```
<audio id="audio" controls hidden></audio>
```

Kono Website er kaj dekte chile browser a giya right click kore View page source korle e dekte pabo.

Ami ai project a ja API use korsi kew chile e excess korte parbe. Ai ta free API ai tar khatre tai chinter kisu nai but ami Jokhon Kono API er jonno pay korbo or company er jonno ai gula sensitive jinish jar excess konno kaw k dera bane na tai aita k ai vabe rakha jabe na. So er solution holo server use kora. Amra nijeder server create korbo or ekta server use korbo.

### Hide API Key:

Where to "hide" an API key

In react app: .env

```
// .gitignore

# api keys
.env <-- add this line

# dependencies
/node_modules
...
```

Jodi heroku or kothaw publish kori ta hole git app a setting a aisha config vars a ashbo. Key add korbo.

Deployment: Hiding API Keys on Netlify

In order to use your secrets in Netlify, go to **Settings > Build & deploy > Environment > Environment variables**.

There, add your variables, just like you had in your .env file.

Index.js

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Joke Teller</title>
  <link rel="icon" type="image/png"
href="favicon.png">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <button id="button">Tell Me A Joke</button>
    <audio id="audio" controls hidden></audio>
  </div>
```

```
<!-- Script -->
<script src="voiceRSS.js"></script>
<script src="script.js"></script>
</body>
</html>
```

Script.js

```
const button = document.getElementById('button');
const audioElement = document.getElementById('audio');

// Disable/Enable Button
function toggleButton() {
  button.disabled = !button.disabled;
}

// VoiceRSS Speech Function
function tellMe(joke) {
  const jokeString = joke.trim().replace(/ /g, '%20');
  // VoiceRSS Speech Parameters
  VoiceRSS.speech({
    key: "dcc18cda0fa34ee1bd1689e0a3e857c4",
    src: jokeString,
    hl: "en-us",
    r: 0,
    c: "mp3",
    f: "44khz_16bit_stereo",
    ssml: false,
  });
}
```

```

// Get jokes from Joke API
async function getJokes() {
  let joke = '';
  const apiUrl =
'https://sv443.net/jokeapi/v2/joke/Programming?blacklist
Flags=nsfw,racist,sexist';
  try {
    const response = await fetch(apiUrl);
    const data = await response.json();
    // Assign One or Two Part Joke
    if (data.setup) {
      joke = `${data.setup} ... ${data.delivery}`;
    } else {
      joke = data.joke;
    }
    // Passing Joke to VoiceRSS API
    tellMe(joke);
    // Disable Button
    toggleButton();
  } catch (error) {
    // Catch Error Here
  }
}

// Event Listeners
button.addEventListener('click', getJokes);
audioElement.addEventListener('ended', toggleButton);

```

voiceRSS.js

```
// VoiceRSS Javascript SDK
```

```
const VoiceRSS = {
  speech(e) {
    this._validate(e), this._request(e);
  },
  _validate(e) {
    if (!e) throw "The settings are undefined";
    if (!e.key) throw "The API key is undefined";
    if (!e.src) throw "The text is undefined";
    if (!e.hl) throw "The language is undefined";
    if (e.c && "auto" !== e.c.toLowerCase()) {
      let a = !1;
      switch (e.c.toLowerCase()) {
        case "mp3":
          a = new
Audio().canPlayType("audio/mpeg").replace("no", "");
          break;
        case "wav":
          a = new
Audio().canPlayType("audio/wav").replace("no", "");
          break;
        case "aac":
          a = new
Audio().canPlayType("audio/aac").replace("no", "");
          break;
        case "ogg":
          a = new
Audio().canPlayType("audio/ogg").replace("no", "");
          break;
        case "caf":
          a = new Audio().canPlayType("audio/x-
caf").replace("no", "");
      }
    }
  }
}
```

```

        if (!a) throw `The browser does not support the
audio codec ${e.c}`;
    }
},
_request(e) {
    const a = this._buildRequest(e),
        t = this._getXHR();
    (t.onreadystatechange = function () {
        if (4 == t.readyState && 200 == t.status) {
            if (0 == t.responseText.indexOf("ERROR")) throw
t.responseText;
            let e = t.responseText;
            (audioElement.src = e),
                (audioElement.onloadedmetadata = () => {
                    audioElement.play();
                });
        }
    }),
        t.open("POST", "https://api.voicerss.org/", !0),
        t.setRequestHeader(
            "Content-Type",
            "application/x-www-form-urlencoded; charset=UTF-
8"
        ),
        t.send(a);
},
_buildRequest(e) {
    const a = e.c && "auto" != e.c.toLowerCase() ? e.c :
this._detectCodec();
    return `key=${e.key || ""}&src=${e.src ||
""}&hl=${e.hl || ""}&r=${
e.r || ""

```

```

        }&c=${a || ""}&f=${e.f || ""}&ssml=${e.ssml || ""}&b64=true`;
    },
    _detectCodec() {
        const e = new Audio();
        return e.canPlayType("audio/mpeg").replace("no", "")
            ? "mp3"
            : e.canPlayType("audio/wav").replace("no", "")
            ? "wav"
            : e.canPlayType("audio/aac").replace("no", "")
            ? "aac"
            : e.canPlayType("audio/ogg").replace("no", "")
            ? "ogg"
            : e.canPlayType("audio/x-caf").replace("no", "")
            ? "caf"
            : "";
    },
    _getXHR() {
        try {
            return new XMLHttpRequest();
        } catch (e) {}
        try {
            return new ActiveXObject("Msxml3.XMLHTTP");
        } catch (e) {}
        try {
            return new ActiveXObject("Msxml2.XMLHTTP.6.0");
        } catch (e) {}
        try {
            return new ActiveXObject("Msxml2.XMLHTTP.3.0");
        } catch (e) {}
        try {
            return new ActiveXObject("Msxml2.XMLHTTP");
        }
    }
}

```



```
    } catch (e) {}  
    try {  
        return new ActiveXObject("Microsoft.XMLHTTP");  
    } catch (e) {}  
    throw "The browser does not support HTTP request";  
},  
};
```

Style.js

```
/* ROBOT.GIF from Giphy - https://giphy.com/gifs/robot-  
cinema-4d-eyedesyn-3o7abtn7DuREEpsyWY */  
  
html {  
    box-sizing: border-box;  
}  
  
body {  
    margin: 0;  
    background: #45badd;  
}  
  
.container {  
    height: 100vh;  
    width: 100vw;  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
    align-items: center;  
    background: url('./robot.gif');
```

```
background-size: contain;
background-position: left center;
background-repeat: no-repeat;
}

button {
  cursor: pointer;
  outline: none;
  width: 200px;
  height: 50px;
  font-family: Courier, monospace;
  font-size: 20px;
  color: white;
  background: #ff3482;
  border: none;
  border-radius: 5px;
  box-shadow: 2px 2px 20px 10px rgba(0, 0, 0, 0.2);
}

button:hover {
  filter: brightness(95%);
}

button:active {
  transform: scale(0.98);
}

button:disabled {
  cursor: default;
  filter: brightness(30%);
}
```

```
/* Media Query: Tablet or Smaller */
@media screen and (max-width: 1000px) {
  .container {
    background-position: center center;
    background-size: cover;
  }

  button {
    box-shadow: 5px 5px 30px 20px rgba(0, 0, 0, 0.5);
  }
}
```