

Jump to

- lottie-player Web Component
- Demo
- Documentation
- Installation
  - In HTML, import from CDN or from the local Installation:
  - In Javascript or TypeScript:
- Usage
  - Lottie-Player
  - TGS-Player
  - ReactJS & VueJS
  - Typescript ReactJS
  - NuxtJS
  - NextJS
- Properties
- Methods
  - getLottie() => Promise<any>
    - Returns
  - load(src: string | object) => void
    - Parameters
    - Returns
  - pause() => void
    - Returns
  - play() => void
    - Returns
  - setDirection(value: number) => void
    - Parameters
    - Returns
  - setLooping(value: boolean) => void
    - Parameters
    - Returns
  - setSpeed(value?: number) => void
    - Parameters
    - Returns
  - stop() => void
    - Returns
  - seek(value: number | string) => void
    - Returns
  - snapshot(download?: boolean) => string
    - Returns
  - toggleLooping() => void
    - Returns
  - togglePlay() => void
    - Returns
  - resize() => void
    - Returns
- Events
- Styling
- License

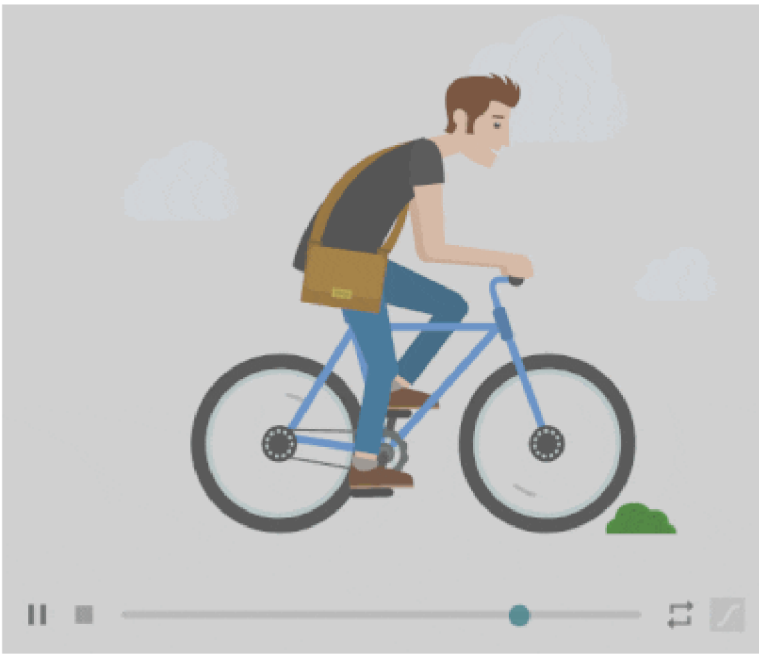
Readme

## lottie-player Web Component

This is a Web Component for easily embedding and playing Lottie animations and the Lottie-based Telegram Sticker (tgs) animations in websites.

npm v1.1.1 webcomponents.org published

## Demo



- [Basic usage examples](#)
- [Scrolling effects demo](#)

## Documentation

- [View documentation](#)

## Installation

In HTML, import from CDN or from the local Installation:

Lottie Player:

- Import from CDN.

```
<script src="https://unpkg.com/@lottiefles/lottie-player@0.4.0/dist/lottie-player.js"></script>
```

- Import from local node\_modules directory.

```
<script src="/node_modules/@lottiefles/lottie-player/dist/lottie-player.js"></script>
```

Telegram Sticker (TGS) Player:

- Import from CDN.

```
<script src="https://unpkg.com/@lottiefles/lottie-player@0.4.0/dist/tgs-player.js"></script>
```

- Import from local node\_modules directory.

```
<script src="/node_modules/@lottiefles/lottie-player/dist/tgs-player.js"></script>
```

In Javascript or TypeScript:

1. Install package using npm or yarn.

```
npm install --save @lottiefles/lottie-player
```

2. Import package in your code.

```
import "@lottiefles/lottie-player";
```

## Usage

### Lottie-Player


Add the element `lottie-player` and set the `src` property to a URL pointing to a valid Bodymovin JSON.

@lottiefles/lottie-player

★ Rate

🔖 Save

npm i @lottiefles/lottie-player



npm • GitHub • CDN

JS

Overview

Documentation

Tutorials

Reviews

Maintenance

Dependencies

Versions

Alternatives

</lottie-player>

You may set and load animations programatically as well.

Jump to

- lottie-player Web Component
- Demo
- Documentation
- Installation
  - In HTML, import from CDN or from the local Installation:
  - In Javascript or TypeScript:
- Usage
  - Lottie-Player
  - TGS-Player
  - ReactJS & VueJS
  - Typescript ReactJS
  - NuxtJS
  - NextJS
- Properties
- Methods
  - getLottie() => Promise<any>
    - Returns
  - load(src: string | object) => void
    - Parameters
    - Returns
  - pause() => void
    - Returns
  - play() => void
    - Returns
  - setDirection(value: number) => void
    - Parameters
    - Returns
  - setLooping(value: boolean) => void
    - Parameters
    - Returns
  - setSpeed(value?: number) => void
    - Parameters
    - Returns
  - stop() => void
    - Returns
  - seek(value: number | string) => void
    - Returns
  - snapshot(download?: boolean) => string
    - Returns
  - toggleLooping() => void
    - Returns
  - togglePlay() => void
    - Returns
  - resize() => void
    - Returns
- Events
- Styling
- License

<lottie-player autoplay controls loop mode="normal" style="width: 320px">

const player = document.querySelector("lottie-player");

player.load("https://assets3.lottiefles.com/packages/1f20\_UJNc2t.json");

// or Load via a Bodymovin JSON string/object

player.load(

'{"v": "5.3.4", "fr": 30, "ip": 0, "op": 38, "w": 315, "h": 600, "nm": "new", ... }'

);

TGS-Player

Add the element tgs-player and set the src property to a URL pointing to a valid TGS JSON.

<tgs-player autoplay loop mode="normal" src="https://domain/example.tgs">

ReactJS & VueJS

Import the player either as

import \* as LottiePlayer from "@lottiefles/lottie-player";

or

require("@lottiefles/lottie-player");

Use as follows

<lottie-player

autoplay

controls

loop

mode="normal"

src="https://assets3.lottiefles.com/packages/1f20\_UJNc2t.json"

style="width: 320px"

></lottie-player>

Typescript ReactJS

Import the player either as

import \* as LottiePlayer from "@lottiefles/lottie-player";

or

require("@lottiefles/lottie-player");

Use as follows

<lottie-player

autoplay

controls

loop

mode="normal"

src="https://assets3.lottiefles.com/packages/1f20\_UJNc2t.json"

style="width: 320px"

></lottie-player>

For typescript projects an added step is required. The component must be declared as a JSX intrinsic element. Create a file 'declarations.d.ts' in the root of your project and add the code below to the file.

declare namespace JSX {

interface IntrinsicElements {

"lottie-player": any;

}

}

NuxtJS

The process for NuxtJS is slightly different. Create a lottie-player.js file in project root inside a folder named 'plugins'. Add the code below to the file

import \* as LottiePlayer from "@lottiefles/lottie-player";

Open nuxt.config.js file and adjust the plugins array as shown below

plugins: [{ src: "~/plugins/lottie-player.js", mode: "client" }],

https://openbase.com/js/@lottiefles/lottie-player/documentation#events


2/5

@lottiefles/lottie-player

★ Rate

🔖 Save

npm i @lottiefles/lottie-player



npm • GitHub • CDN

JS

Overview

Documentation

Tutorials

Reviews

Maintenance

Dependencies

Versions

Alternatives

Jump to

lottie-player Web Component

Demo

Documentation

Installation

In HTML, import from CDN or from the local Installation:

In Javascript or TypeScript:

Usage

Lottie-Player

TGS-Player

ReactJS & VueJS

Typescript ReactJS

NuxtJS

NextJS

Properties

Methods

getLottie() => Promise<any>

Returns

load(src: string | object) => void

Parameters

Returns

pause() => void

Returns

play() => void

Returns

setDirection(value: number) => void

Parameters

Returns

setLooping(value: boolean) => void

Parameters

Returns

setSpeed(value?: number) => void

Parameters

Returns

stop() => void

Returns

seek(value: number | string) => void

Returns

snapshot(download?: boolean) => string

Returns

toggleLooping() => void

Returns

togglePlay() => void

Returns

resize() => void

Returns

Events

Styling

License

loop

style="width:400px"

src="https://assets3.lottiefles.com/packages/1f20\_RItkEz.json"

speed="1"

debug

></lottie-player>

This is because the player script needs to be rendered on the browser/client side and we must configure nuxtjs to load the script on the client side only.

NextJS

The process to import in NextJS is similar to NuxtJS in the sense that on SSR mode, the library must be declared as a client side module. To do this, import the library within a react useEffect hook.

```
import React, { useRef } from "react";

export default function Home() {
  const ref = useRef(null);
  React.useEffect(() => {
    import("@lottiefles/lottie-player");
  });
  return (
    <div className={styles.container}>
      <main className={styles.main}>
        <lottie-player
          id="firstLottie"
          ref={ref}
          autoplay
          controls
          loop
          mode="normal"
          src="https://assets4.lottiefles.com/packages/1f20_gb5bmwlm.json"
          style={{ width: "300px", height: "300px" }}
        ></lottie-player>
      </main>
    </div>
  );
}
```

Do add a declaration file named declaration.d.ts to the root of the project as well

```
declare namespace JSX {
  interface IntrinsicElements {
    "lottie-player": any;
  }
}
```

Properties

Property	Attribute	Description	Type	Default
autoplay	autoplay	Autoplay animation on load.	boolean	false
background	background	Background color.	string	undefined
controls	controls	Show controls.	boolean	false
count	count	Number of times to loop animation.	number	undefined
direction	direction	Direction of animation.	number	1
hover	hover	Whether to play on mouse hover.	boolean	false
loop	loop	Whether to loop animation.	boolean	false
mode	mode	Play mode.	PlayMode.Bounce \  PlayMode.Normal	PlayMode.Normal
preserveAspectRatio	preserveAspectRatio	Valid preserve aspect ratio value.	string	'xMidYMid meet'
renderer	renderer	Renderer to use.	'svg'	'canvas'
speed	speed	Animation speed.	number	1
src <i>(required)</i>	src	Bodymovin JSON data or URL to JSON.	string	undefined

Direction value options are 1 and -1

Methods

getLottie() => Promise<any>

Returns the instance of lottie player used in the component.

Returns

Type: Promise<any>

load(src: string | object) => void

Load (and play) a given Bodymovin animation.

Parameters

Name	Type	Description
------	------	-------------

@lottiefiles/lottie-player

★ Rate

🔖 Save

npm i @lottiefiles/lottie-player

📦

npm

•

GitHub

•

CDN

•

JS

Overview

Documentation

Tutorials

Reviews

Maintenance

Dependencies

Versions

Alternatives

Jump to

•

lottie-player Web Component

•

Demo

•

Documentation

•

Installation

•

In HTML, import from CDN or from the local Installation:

•

In Javascript or TypeScript:

•

Usage

•

Lottie-Player

•

TGS-Player

•

ReactJS & VueJS

•

Typescript ReactJS

•

NuxtJS

•

NextJS

•

Properties

•

Methods

•

getLottie() => Promise<any>

•

Returns

•

load(src: string | object) => void

•

Parameters

•

Returns

•

pause() => void

•

Returns

•

play() => void

•

Returns

•

setDirection(value: number) => void

•

Parameters

•

Returns

•

setLooping(value: boolean) => void

•

Parameters

•

Returns

•

setSpeed(value?: number) => void

•

Parameters

•

Returns

•

stop() => void

•

Returns

•

seek(value: number | string) => void

•

Returns

•

snapshot(download?: boolean) => string

•

Returns

•

toggleLooping() => void

•

Returns

•

togglePlay() => void

•

Returns

•

resize() => void

•

Returns

•

Events

•

Styling

•

License

pause() => void

Pause animation play.

Returns

Type: void

play() => void

Start playing animation.

Returns

Type: void

setDirection(value: number) => void

Animation play direction.

Parameters

Name	Type	Description
value	number	Direction values.

Returns

Type: void

setLooping(value: boolean) => void

Sets the looping of the animation.

Parameters

Name	Type	Description
value	boolean	Whether to enable looping. Boolean true enables looping.

Returns

Type: void

setSpeed(value?: number) => void

Sets animation play speed.

Parameters

Name	Type	Description
value	number	Playback speed.

Returns

Type: void

stop() => void

Stops animation play.

Returns

Type: void

seek(value: number | string) => void

Seek to a given frame. Frame value can be a number or a percent string (e.g. 50%).

Returns

Type: void

snapshot(download?: boolean) => string

Snapshot the current frame as SVG. If 'download' argument is boolean true, then a download is triggered in browser.

Returns

Type: string

toggleLooping() => void

Toggles animation looping.

Returns

Type: void

togglePlay() => void

Toggle playing state.

Returns

Type: void

https://openbase.com/js/@lottiefiles/lottie-player/documentation#events

4/5



Resize animation stage and elements in response to changes in component.

Returns

Type: void

Events

The following events are exposed and can be listened to via `addEventListener` calls.

Name	Description
load	Animation data is loaded.
error	An animation source cannot be parsed, fails to load or has format errors.
ready	Animation data is loaded and player is ready.
play	Animation starts playing.
pause	Animation is paused.
stop	Animation is stopped.
freeze	Animation is paused due to player being invisible.
loop	An animation loop is completed.
complete	Animation is complete (all loops completed).
frame	A new frame is entered.

Styling

Custom property	Description	Default
--lottie-player-toolbar-height	Toolbar height	35px
--lottie-player-toolbar-background-color	Toolbar background color	transparent
--lottie-player-toolbar-icon-color	Toolbar icon color	#999
--lottie-player-toolbar-icon-hover-color	Toolbar icon hover color	#222
--lottie-player-toolbar-icon-active-color	Toolbar icon active color	#555
--lottie-player-seeker-track-color	Seeker track color	#CCC
--lottie-player-seeker-thumb-color	Seeker thumb color	rgba(0, 107, 120, 0.8)

License

MIT License © LottieFiles.com

Jump to

- lottie-player Web Component
- Demo
- Documentation
- Installation
  - In HTML, import from CDN or from the local installation:
  - In Javascript or TypeScript:
- Usage
  - Lottie-Player
  - TGS-Player
  - ReactJS & VueJS
  - Typescript ReactJS
  - NuxtJS
  - NextJS
- Properties
- Methods
  - getLottie() => Promise<any>
    - Returns
  - load(src: string | object) => void
    - Parameters
    - Returns
  - pause() => void
    - Returns
  - play() => void
    - Returns
  - setDirection(value: number) => void
    - Parameters
    - Returns
  - setLooping(value: boolean) => void
    - Parameters
    - Returns
  - setSpeed(value?: number) => void
    - Parameters

Openbase helps developers choose among and use millions of open-source packages, so they can build amazing products faster.

- Product
- Home
  - What is Openbase?
  - Categories
  - Random Package
  - Leaderboard
  - Documentation
  - Contact Us

- Top Libraries
- Top JS Libraries
  - Top Python Libraries
  - Top Go Libraries
  - Top Rust Libraries
  - Top Swift Libraries
  - Top Java Libraries

- Company
- About
  - Careers
  - Blog
  - Media Kit
  - Privacy
  - Terms

- Returns
- resize() => void
  - Returns
- Events
- Styling
- License

