

# bodymovin

After Effects plugin for exporting animations to svg/canvas/html + js or natively on Android and iOS through [Lobla](https://medium.com/@ibob-engineering/introducing-lobla-4ff4a9afac0e) <https://medium.com/@ibob-engineering/introducing-lobla-4ff4a9afac0e>

Installation instructions:

- Extract content and search for the .zip file from 'build/extension'
- Use the [ZIP installer](http://descriptslabs.com/learn/zip-installer/) <http://descriptslabs.com/learn/zip-installer/> from aescrpts.com.

## V 5.5.3

- FEATURE: Add SVG support for Gaussian Blur effects (thanks fmalita)
- FEATURE: Add TypeScript type definitions (thanks D34THWINGS)
- FIX: removed warning of text when created by text formatted
- FIX: fixed merged characters in fonts
- FEATURE: added multiple settings functionality to extension
- FEATURE: added support for mask opacity in expressions
- FIX: fixes #1552 html renderer wrong font measurement
- FIX: Fix maskProperties key name in JSON docs (thanks john-predon)

## V 5.5.2

- FIX: Fix loader issue on pre-KitKat Android webview
- FIX: eroded masks in svg renderer
- FIX: text alignment in text boxes

## V 5.5.1

- FIX: key expression for new json format
- IMPROVEMENT: forcing a renderer when resize is called in canvas
- ACCESSIBILITY: added title, description and aria label for text layers for svg renderer

## V 5.5.0

- IMPROVEMENT: JSON filesize reduction. Around 25 to 33% filesize reduction with this new version.
- FEATURE: Bodymovin panel with new advanced settings to reduce filesize.
- FIX: honor `fontSize` when calling `playSegments` (thanks @mmms)
- FIX: added source over as default blending mode for canvas renderer setting
- FIX: fixed bezier easing property validation
- FIX: fixed property caching by index instead of bezier data
- FIX: added `thisProperty` expression value

## V 5.4.4

- FIX: viewport meta tag on android with mask fix
- FIX: fixed enterFrame direction property
- FIX: explicitly iterating math methods
- FIX: added missing expression properties
- FIX: setting class attribute via `setAttribute`
- FIX: added value property in key method expression
- FIX: text animator expression fix
- FIX: added smooth expression support
- FIX: hcamera expression fix
- FIX: fix animated dash property canvas
- NEW: `sourceRectAtTime` for images fix
- NEW: Advanced option in Bodymovin extension to skip expression properties
- NEW: removed some unused json properties

## V 5.4.3

- NEW: added blend mode support for shapes in svg renderer
- NEW: removed `random ids` in favour of incremental ids
- NEW: added new players
- NEW: updated build process. Removed vulnerable dependencies.
- FIX: canvas repeater fix
- FIX: Replaced typetk with Adobe Fonts support
- FIX: subtracting `offsetTime` for valueAtTime calculation on shapes
- FIX: expressions targeting keys fix
- FIX: multidimensional easing using first dimension value when set to 0

## V 5.4.2

- FIX: added enabled property on effect interfaces for expression
- FIX: font measuring fix for white spaces
- FIX: supporting Effects property on layers in expressions
- FIX: canvas filling shapes with same style fix
- FIX: caching valueAtTime correctly
- FIX: AVD exporter fix on clip paths
- FIX: AVD exporter fix on initial and end values
- FIX: formatResponse if responseText - thanks @sanmonjol
- NEW: renders class name for nested group elements - thanks @russelgoldenberg

## V 5.4.1

- FIX: missing variable declaration

## V 5.4.0

- EXPRESSIONS: easing functions refactored
- FIX: stroke effect fix with multiple children elements
- FIX: adding max and min values to trim path start and end
- EXPRESSIONS: added velocity property
- FIX(text): Fix charCode assignment and optimize FontManager (thanks @williams-curago)
- FIX: fix for small trim paths
- EXPRESSIONS: added valueAtTime property to expression `thisProperty` variable
- FIX: added locationHref to gradient data
- EXPRESSIONS: big performance improvement for all expressions that use Expression Values heavily

## V 5.3.4

- FIX: font load fix
- TEXT: End of Text character support

## V 5.3.3

- FIX: `playSegments` when forcing a new segment correctly removes the previous list
- FIX: loading external assets before rendering the first frame of the animation on canvas renderer
- FIX: clearing caching spatial bezier data when previous frame is reached
- FIX: promoting text property to dynamic properties when using text update methods
- FIX: extra comps width and height properties supported
- FIX: multiple trims fixed

## V 5.3.1

- FIX: expressions with non computed memberExpressions
- FIX: animated gradient in segs

## V 5.3.0

- REFACTOR: canvas shape renderer had an important refactor. Should support new render cases and have a performance improvement.
- EXPRESSIONS: preprocessing expressions allows to prevent getting values if not needed on expressions
- EXPRESSIONS: support for position, scale and anchorPoint variables
- EXPRESSIONS: added `numLayer` property support to comp interfaces
- REPEATERS: support for start and end opacity
- FIX: Rendering effects before masks fixes some small render cases
- FIX: added orientation support for 3d cameras
- FIX: dashed lines were not rescaled in some scenarios

## V 5.2.1

- FIX: text animations based on words and percentages
- EXPRESSIONS: Latest Dukt version support

## V 5.2.0

- FEATURE: gradient support in canvas renderer (some cases are not supported)
- EXPRESSIONS: layer name support for expressions
- FIX: reverse play on not loops
- SUPPORT: Orient along path with separate dimensions
- FIX: trim cache issue fixed

## V 5.1.20

- FIX: instanceof Array on expressions fixed
- FIX: text value on expressions fix
- FEATURE: new expressions supported

## V 5.1.19

- FIX: Trim Paths with empty paths fix
- FIX: linear method fix for inverted values
- FEATURE: Ignoring merge path's last path if square
- FIX: Camera zoom
- FIX: Validating if text data is complete on first render
- FIX: Loop counting when playing bookends
- FEATURE: Added `inPoint` and `outPoint` to layer's expressions

## V 5.1.18

- FIX: Moved defs to top to avoid Safari issues
- FEATURE: Added `crossOrigin` attribute to images to avoid tainted canvases
- FEATURE: Added `imagePreserveAspectRatio` to `rendererSettings` for image layers
- FIX: splitting animations in multiple files

## V 5.1.17

- FIX: asset error handling
- FEATURE: Rove across time support

- FIX: bevel line support

## V 5.1.16

- FIX: exporting images for older version of AE.
- FEATURE: if needed can skip images export once exported a first time.

## V 5.1.15

- FEATURE: Added freeze/unfreeze methods to stop prevent any animation from playing
- FEATURE: Added getRegisteredAnimations method to get all current animations handled by lottie
- FEATURE: Exporting PNGs with render queue in order to fix black pixels around images
- FEATURE: Support for compressed jpgs when image is not transparent
- FEATURE: Support for exporting base 64 encoded images inlined in the json file
- FIX: Some small expression fixes

## V 5.1.14

- FIX: Dash property animated
- FIX: Canvas renderer skips hidden layers
- FIX: When clearCanvas is set to true on the canvas renderer, every frame is rendered
- FIX: calculation error on mase

## V 5.1.13

- EXPRESSIONS: fix for global variables used in functions
- EXPRESSIONS: operations supported for arrays

## V 5.1.12

- EXPRESSIONS: improved memory management for declared functions
- EXPRESSIONS: speedAtTime support
- FIX: timed paths fix
- FIX: destroy method fix
- SSR: checking for navigator to create library

## Lottie + Bodymovin

Lottie is the native engine that Airbnb's awesome team built. It uses Bodymovin as the animation exporter and is the ideal complement for getting animations to play natively everywhere.

Follow these links to get each player:

- [Android's player](https://github.com/airbnb/lottie-android) <https://github.com/airbnb/lottie-android>
- [iOS's player](https://github.com/airbnb/lottie-ios) <https://github.com/airbnb/lottie-ios>
- [React Native's wrapper](https://github.com/airbnb/lottie-react-native) <https://github.com/airbnb/lottie-react-native>

## Lottie and AVD

Some animations can be exported for Android using the AVD format.

It can fit for some cases where you'll gain a performance improvement.

But Lottie brings much more features, a level of animation control and dynamic loading that couldn't be achieved with avd.

Here's a [link](http://airbnb.io/lottie-lottie-avd.html) <http://airbnb.io/lottie-lottie-avd.html> with a full comparison of both technologies.

## After installing

- Go to Edit > Preferences > General > and check on "Allow Scripts to Write Files and Access Network"

## HTML player installation

```
# with npm
npm install lottie-web

# with bower
bower install bodymovin
```

Or you can use the script file from here:

<https://cdnjs.com/libraries/bodymovin>

Or get it directly from the AE plugin clicking on Get Player

## Demo

[See a basic implementation here](https://codepen.io/aiman/project/editor/ZehONQO) <https://codepen.io/aiman/project/editor/ZehONQO> <br>

## Examples

[See examples on codepen](http://codepen.io/collecionn/VYVZRV) <http://codepen.io/collecionn/VYVZRV> <br>

## How it works

[Here's https://www.youtube.com/watch?v=5XMU4d0L8](https://www.youtube.com/watch?v=5XMU4d0L8) a video tutorial explaining how to export a basic animation and load it in an html page <br />

## After Effects

- Open your AE project and select the bodymovin extension on Window > Extensions > bodymovin
- A Panel will open with a Compositions tab listing all of your Project Compositions
- Select the composition you want to export.
- Select a Destination Folder:
- Click Render
- look for the exported json file (if you had images or AI layers on your animation, there will be an images folder with the exported files)

## HTML

- get the lottie.js file from the build/player/ folder for the latest build
- include the .js file on your html (remember to gzip it for production)

```
<script src="js/lottie.js" type="text/javascript"></script>
```

You can call `lottie.loadAnimation()` to start an animation.

It takes an object as a unique param with:

- `animationData`: an Object with the exported animation data.
- `path`: the relative path to the animation object. (`animationData` and `path` are mutually exclusive)
- `loop`: true / false / number
- `autoplay`: true / false it will start playing as soon as it is ready
- `name`: animation name for future reference
- `renderer`: 'svg' / 'canvas' / 'html' to set the renderer
- `container`: the dom element on which to render the animation

It returns the animation instance you can control with `play`, `pause`, `setSpeed`, etc.

```
lottie.loadAnimation({
  container: element, // the dom element that will contain the animation
  renderer: 'svg',
  loop: true,
  autoplay: true,
  path: 'data.json' // the path to the animation json
});
```

## Composition Settings:

Check this [wiki](https://github.com/airbnb/lottie-web/wiki/Composition-Settings) page for an explanation for each setting.

<https://github.com/airbnb/lottie-web/wiki/Composition-Settings>

## Usage

Animation instances have these main methods

### play

### stop

### pause

### setLocationHref(href)

- `href`: usually pass as `location.href`. Its useful when you experience `mask` issue in safari where your url does not have `#` symbol.

### setSpeed(speed)

- `speed`: 1 is normal speed.

### goToAndStop(value, isFrame)

- `value`: numeric value.
- `isFrame`: defines if first argument is a time based value or a frame based (default false).

### goToAndPlay(value, isFrame)

- `value`: numeric value.
- `isFrame`: defines if first argument is a time based value or a frame based (default false).

### setDirection(direction)

- `direction`: 1 is forward, -1 is reverse.

### playSegments(segments, forceFlag)

- `segments`: array. Can contain 2 numeric values that will be used as first and last frame of the animation. Or can contain a sequence of arrays each with 2 numeric values.
- `forceFlag`: boolean. If set to false, it will wait until the current segment is complete. If true, it will update values immediately.

### setSubFrames(useSubFrames)

- `useSubFrames`: If false, it will respect the original AE fps. If true, it will update on every `requestAnimationFrame` with intermediate values. Default is true.

destroy()

#### getDuration(inFrames)

- inFrames: If true, returns duration in frames, if false, in seconds.

#### Additional methods:

- updateTextDocumentData – updates a text layer's data  
[More info \(https://github.com/airbnb/lottie-web/wiki/Text-layer-update\(DocumentData\)\)](https://github.com/airbnb/lottie-web/wiki/Text-layer-update(DocumentData))

#### lottie has 8 global methods that will affect all animations:

**lottie.play()** – with 1 optional parameter **name** to target a specific animation <br>  
**lottie.stop()** – with 1 optional parameter **name** to target a specific animation <br>  
**lottie.setSpeed()** – first argument speed (1 is normal speed) – with 1 optional parameter **name** to target a specific animation <br>  
**lottie.setDirection()** – first argument direction (1 is normal direction.) – with 1 optional parameter **name** to target a specific animation <br>  
**lottie.searchAnimations()** – looks for elements with class "lottie" <br>  
**lottie.loadAnimation()** – Explained above. returns an animation instance to control individually. <br>  
**lottie.destroy()** – To destroy and release resources. The DOM element will be emptied.<br>  
**lottie.registerAnimation()** – you can register an element directly with registerAnimation. It must have the "data-animation-path" attribute pointing at the data.json url <br>  
**lottie.setQuality()** – default 'high', set 'high','medium','low', or a number > 1 to improve player performance. In some animations as low as 2 won't show any difference.<br>

#### Events

- onComplete
- onLoopComplete
- onEnterFrame
- onSegmentStart

you can also use addEventListener with the following events:

- complete
- loopComplete
- enterFrame
- segmentStart
- config\_ready (when initial config is done)
- data\_ready (when all parts of the animation have been loaded)
- loaded\_images (when all image loads have either succeeded or errored)
- DOMLoaded (when elements have been added to the DOM)
- destroy

#### Other loading options

- if you want to use an existing canvas to draw, you can pass an extra object: 'renderer' with the following configuration:

```
lottie.loadAnimation({
  container: element, // the dom element
  renderer: 'svg',
  loop: true,
  autoplay: true,
  animationData: animationData, // the animation data
  rendererSettings: {
    context: canvasContext, // the canvas context
    scaleMode: 'noScale',
    clearCanvas: false,
    progressiveLoad: false, // Boolean, only svg renderer, loads dom elements when needed. Might speed up initialization for large number of elements.
    hideOnTransparent: true, // Boolean, only svg renderer, hides elements when opacity reaches 0 (defaults to true)
    className: 'some-css-class-name'
  }
});
```

Doing this you will have to handle the canvas clearing after each frame

<br>

Another way to load animations is adding specific attributes to a dom element.

You have to include a div and set it's class to lottie.

If you do it before page load, it will automatically search for all tags with the class "lottie".

Or you can call lottie.searchAnimations() after page load and it will search all elements with the class "lottie".

<br>

- add the data.json to a folder relative to the html
  - create a div that will contain the animation.
- ```
<br>
<div>
  Required
  <br>
  . a class called "lottie"
  . a "data-animation-path" attribute with relative path to the data.json
  <br>
  Optional
  <br>
  . a "data-loop-loop" attribute
  . a "data-name" attribute to specify a name to target play controls specifically
  <br>
  Example
  <br>
```

```
<div style="width:100px;height:60px" class="lottie" data-animation-path="animation/" data-loop="true" data-name="ninja"></div>
```

<br>

#### Preview

You can preview or take an svg snapshot of the animation to use as poster. After you render your animation, you can take a snapshot of any frame in the animation and save it to your disk. I recommend to pass the svg through an svg optimizer like <https://jakearchibald.github.io/svgomg/> and play around with their settings.<br>

#### Recommendations

##### Files

If you have any images or AI layers that you haven't converted to shapes (I recommend that you convert them, so they get exported as vectors, right click each layer and do: "Create shapes from Vector Layer"), they will be saved to an images folder relative to the destination json folder. Beware not to overwrite an existing folder on that same location.

##### Performance

This is real time rendering. Although it is pretty optimized. It always helps if you keep your AE project to what is necessary.<br>  
More optimizations are on their way, but try not to use huge shapes in AE only to mask a small part of it.<br>  
Too many nodes will also affect performance.

##### Help

If you have any animations that don't work or want me to export them, don't hesitate to write. <br>

I'm really interested in seeing what kind of problems the plugin has. <br>

my email is [hernantortisi@gmail.com](mailto:hernantortisi@gmail.com)

#### AE Feature Support

- The script supports precomps, shapes, solids, images, null objects, texts
- It supports masks and inverted masks. Maybe other modes will come but it has a huge performance hit.
- It supports time remapping.
- The script supports shapes, rectangles, ellipses and stars
- Expressions. Check the wiki page for [more info \(https://github.com/airbnb/lottie-web/wiki/Expressions\)](https://github.com/airbnb/lottie-web/wiki/Expressions)
- Not supported: image sequences, videos and audio are not supported
- No negative layer stretching! No idea why, but stretching a layer messes with all the data.

#### Development

npm install & browser install first

npm start

#### Notes

- if you want to modify the parser or the player, there are some gulp commands that can simplify the task
- look at the great animations exported on codepen [See examples on codepen \(https://codepen.io/jordedevn/VYVQZ9\)](https://codepen.io/jordedevn/VYVQZ9)
- grouping the animation .jsons and the player have a huge reduction on the filesize. I recommend doing it if you use it for a project.

#### Issues

- For missing mask in Safari browser, please `anim.setLocationHref(locationHref)` before animation is generated. It usually caused by usage of base tag in html. (see above for description of `setLocationHref`)