



谈谈lottie

START



# 目录

---

WHAT

WHY

HOW

在手百中使用lottie

# WHAT



Easily add high-quality animation to any native app.

Lottie is an iOS, Android, and React Native library that renders After Effects animations in real time, allowing apps to use animations as easily as they use static images.

轻松为任何app程序添加高质量动画。

lottie是一个用来在IOS，Android，RN上实时渲染AE动效的库，app使用动效就像用一张静态图一样简单

# WHY

- 😊 使用方便 - 提供完整的开发&使用方案
- 🚀 高性能 - 采用svg/canvas渲染,
- 🙄 高还原度 - 所见即所得, 可使用矢量
- 🌀 跨平台 - 支持IOS, Android, H5, RN
- ✨ 可交互 - 像骑自行车一样控制你的动画

状态



方向

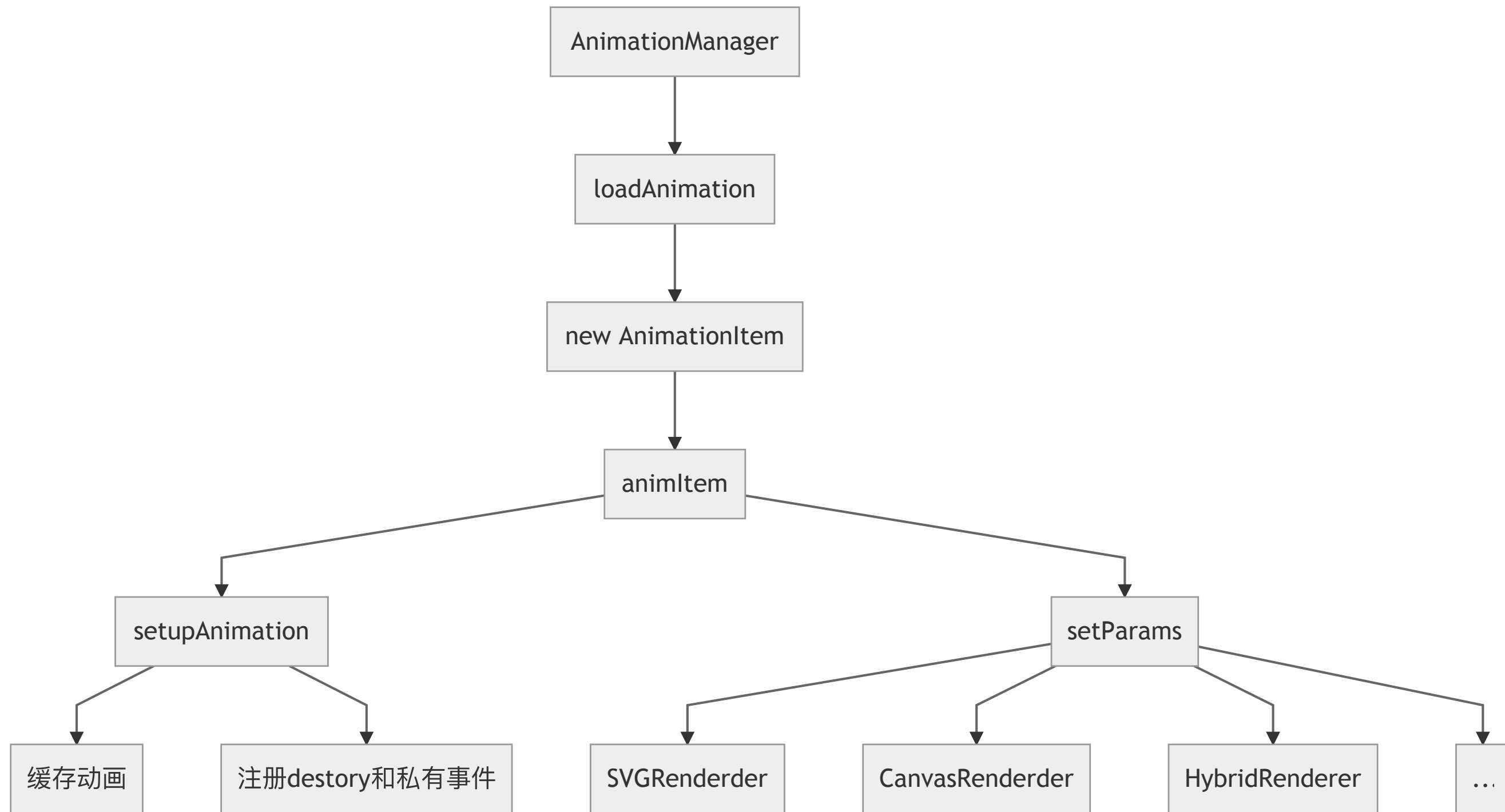


速度



# HOW

## lottie-web 运行流程图



# HOW

## 常用api

```
let lottieInstance = lottie.loadAnimation({
  container: document.getElementById('domId'), // 元素挂载节点
  renderer: 'svg', // 渲染方式 svg | canvas | html
  loop: true, // 是否循环
  autoplay: true, // 是否自动播放
  animationData: {}, // lottie json数据
  path: 'http://path/to/lottie.json' // lottie远程数据, 优先级低于animationData
});
```

api	描述
<code>lottie.loadAnimation</code>	lottie初始化方法，生成一个lottie实例lottieInstance
<code>lottieInstance.play()</code>	播放lottie
<code>lottieInstance.pause()</code>	暂停lottie
<code>lottieInstance.stop()</code>	停止lottie，将lottie重置到首帧

# HOW

## 常用api

api	描述
<code>lottieInstance.setSpeed(speed: number)</code>	设置lottie速度
<code>lottieInstance.setDirection(direction: number)</code>	设置lottie播放反向，1 正向，-1 反向
<code>lottieInstance.playSegments(segments: Array, forceFlag: boolean)</code>	选取一个片段播放
<code>lottieInstance.destroy()</code>	销毁当前lottie实例
<code>lottieInstance.goToAndStop(value: number, isFrame: boolean)</code>	跳转到指定秒数或帧并停止
<code>lottieInstance.goToAndPlay(value: number, isFrame: boolean)</code>	跳转到指定秒数或帧并立即播放

# HOW

## 常用event

```
lottieInstance.addEventListener(eventName, callback)
```

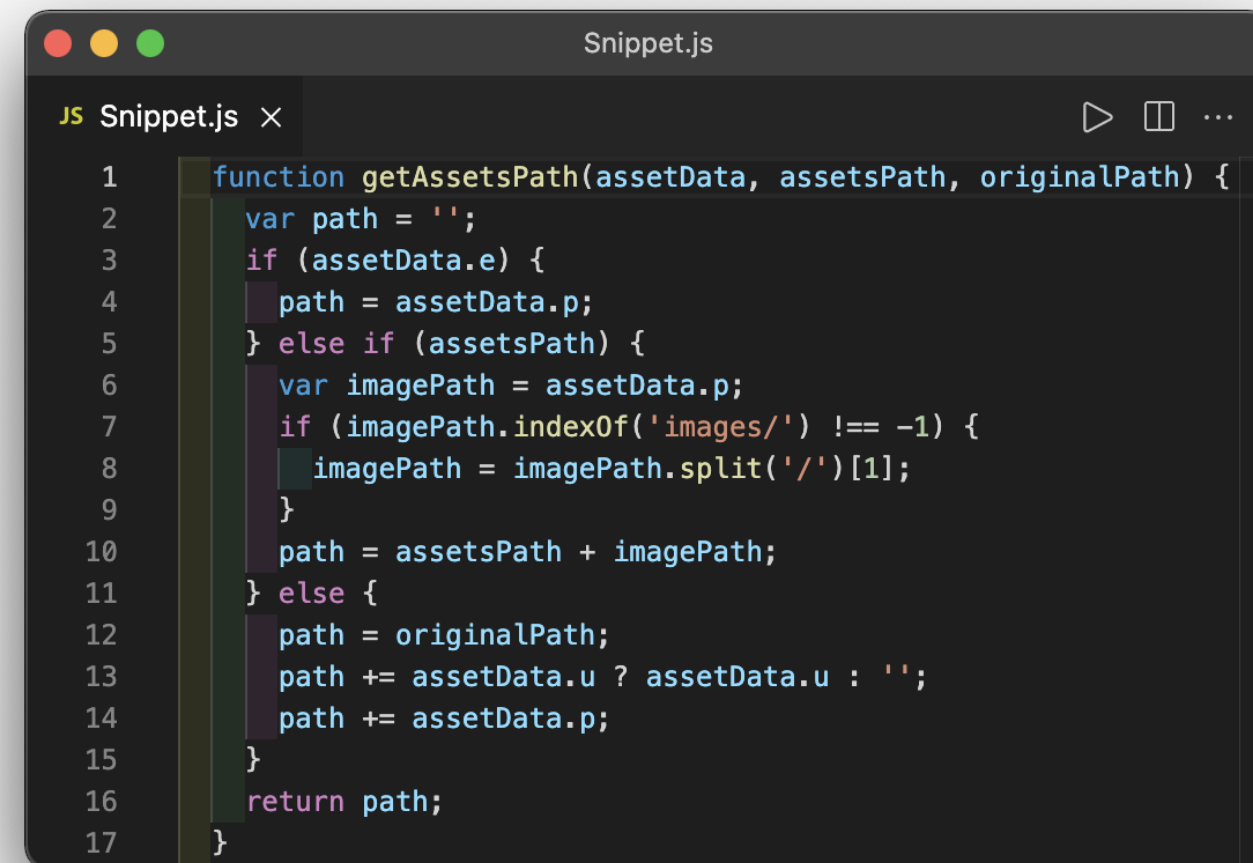
eventName	描述
complete/loopComplete	lottie播放完成/循环播放的lottie每轮播放完成
config_ready/data_ready	初始化配置成功/动画json准备完成
enterFrame	准备播放
data_failed	动画json获取失败
loaded_images	当所有图片加载成功或失败
DOMLoaded	元素插入wrap dom
destroy	销毁



# HOW

## lottie json文件解读

```
{
  "v": "5.5.9", // bodymovin插件版本
  "fr": 30, // 帧率
  "ip": 0, // 开始帧
  "op": 150, // 结束帧
  "w": 512, // 宽
  "h": 512, // 高
  "nm": "DEMO", // lottie名称
  "ddd": 0, // 是否为3d
  "assets": [// 静态资源信息(图片, 图层)
    {
      "id": "imgAssets_0", // 唯一id
      "w": 1000, // 宽
      "h": 1000, // 高
      "u": "", // path前缀
      "p": "", // 图片(绝对|相对)地址/base64 imageData
      "e": 1 // 是否是绝对路径
    }
  ],
  "layers": [/*.....*/] // 图层信息
}
```



```
JS Snippet.js x
1 function getAssetsPath(assetData, assetsPath, originalPath) {
2   var path = '';
3   if (assetData.e) {
4     path = assetData.p;
5   } else if (assetsPath) {
6     var imagePath = assetData.p;
7     if (imagePath.indexOf('images/') !== -1) {
8       imagePath = imagePath.split('/')[1];
9     }
10    path = assetsPath + imagePath;
11  } else {
12    path = originalPath;
13    path += assetData.u ? assetData.u : '';
14    path += assetData.p;
15  }
16  return path;
17 }
```

# HOW

## lottie json文件解读

```
{
  /*.....*/
  "layers": [ // 图层信息
    {
      "ddd": 0, // 是否3d
      "ind": 1, // 图层id
      "ty": 4, // 图层类型 0 合成图层(comp),1 固定图层(solid), 2 图片(image),3 空(null), 4 形状图层(shape), 5 文字图层(text)
      "nm": "Green Box", // 图层名称
      "sr": 1, // 图层缩放
      "ks": { // 动画信息
        "o": { /*.....*/ }, // 透明度
        "r": { /*.....*/ }, // 旋转
        "p": { /*.....*/ }, // 位置
        "a": { /*.....*/ }, // 锚点
        "s": { /*.....*/ } // 缩放
      },
      "shapes": [ /*.....*/ ], // 图形信息
      /*.....*/
    }
  ]
}
```

# HOW

## lottie json文件解读

```
{
  /*.....*/
  "layers": [ // 图层信息
    {
      /*.....*/
      "shapes": [
        {
          /***
           * 图形类型
           * gr: 图形合并,st: 图形描边,fl: 图形填充,tr: 图形变换,sh: 图形路径, el: 椭圆路径, rc: 矩形路径, tm: 剪裁路径
           ***/
          "ty": "gr",
          "it": [/*.....*/], // 混合图形的各图形信息
          /*.....*/
        },
        {/*.....*/}
      ],
      /*.....*/
    }
  ]
}
```



# 在手百小程序中使用lottie

## 小程序原生lottie组件的困境

### 1. lottie组件是**不支持同层渲染**的**原生组件**

- 原生组件的层级是最高的，所以页中的其他组件无论设置 z-index 为多少，都无法盖在原生组件上。后插入的原生组件可以覆盖之前的原生组件。
- 原生组件无法在 scroll-view、swiper、picker-view、movable-view 中使用

### 2. 由于第三方库兼容性问题，在使用 animation-view 组件时，需要将小程序的最低基础库版本设置成大于等于 3.250.12

### 3. 若开发者使用 Adobe After Effects (AE) Bodymovin 插件导出 Lottie 文件时，应该使用版本号低于 5.5 版的 Bodymovin 插件导出 Lottie 文件（推荐使用 v5.4.4），否则会出现与 Lottie SDK 兼容性问题，导致无法正常播放

### 4. path不支持远程路径

### 5. 只能调整动画状态(播放、暂停、停止)，派发的事件只有ended

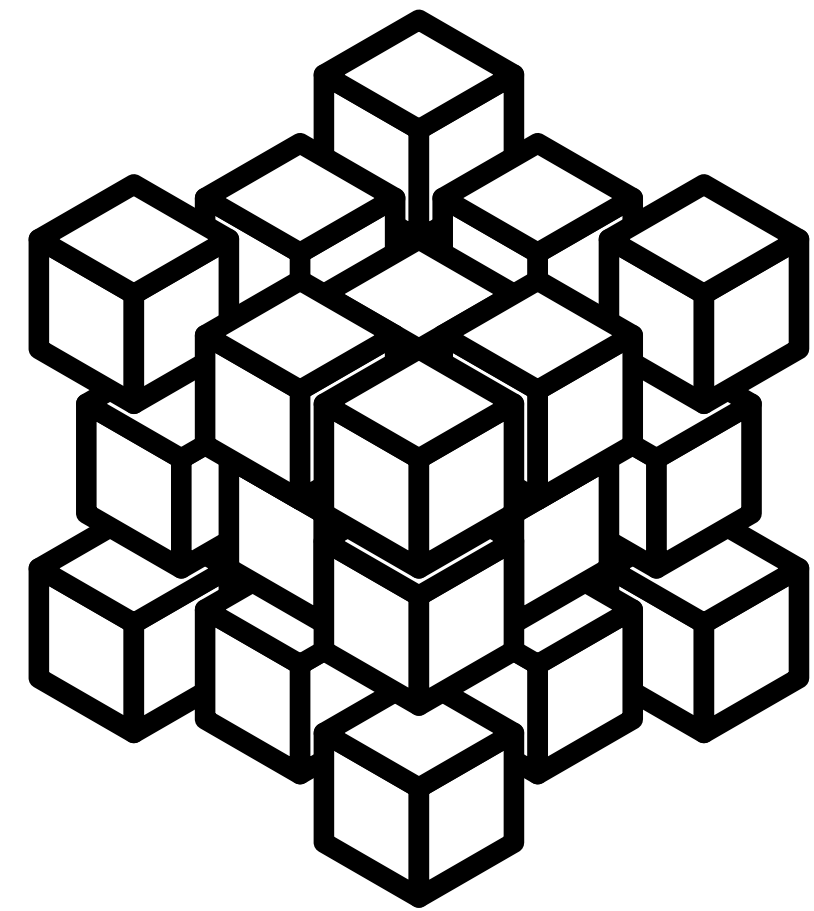
## Detail

# 怎么办?

动态库，是指可被添加到小程序内直接使用的功能组件

优势

1. 不是原生组件
2. 动态库本身没有基础库限制
3. 提供了更加底层的api，web友好



# 在使用lottie时遇到的一些问题，

[戳这里](#)，预览你的lottie



- ios 13.4.1以下的机器，手百小程序无法渲染带渐变的矢量图形
  - 解决方案：指定机型以下采用降级方案(帧动画 or css)
- 导出lottie无法正确渲染渐变图层，[点我看效果](#)
  - 解决方案：有多个图层使用了gradient fill属性，要对各个图层的该属性进行重新命名，**不支持中文**。
- 动态库无法使用canvas渲染
  - 解决方案：使用svg方式渲染
- 性能问题
  - 解决方案：尽量减少同时播放的lottie动画



# Learn More

[Documentations](#) / [GitHub Repo](#)