

uniapp 安卓SVGA动画原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
插件方法	2
开始播放	2.1
暂停播放	2.2
继续播放	2.3
停止播放	2.4
插件事件	3
加载完成	3.1
加载错误	3.2
开始播放	3.3
暂停播放	3.4
重复播放	3.5
播放进度	3.6
播放结束	3.7
网络地址开始加载	3.8
网络地址加载中	3.9
网络地址加载完成	3.10

使用方法

介绍

安卓SVGA插件是原生组件式插件，支持SVGA动画文件格式播放，支持网络地址播放

本插件只适用于nvue页面，不支持vue页面

联系作者

关注微信公众号可联系作者



插件地址

<https://ext.dcloud.net.cn/plugin?id=14139>

所需权限

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

用法

在需要使用插件的页面加载以下代码

```
<leven-svga ref="refLevenSvg" style="height: 500rpx; width: 750rpx;" @onComplete=
@onPause="onPause" @onFinished="onFinished" @onRepeat="onRepeat" @onStep="onStep"
@onLoadingSuccess="onLoadingSuccess"></leven-svga>
```

完整页面内容

```
<template>
  <view>
    <view>
      <uni-card title="uniappSVG"原生插件">
        <leven-svga ref="refLevenSvg" style="height: 500rpx; width: 750rpx;" @onComplete=
          @onPause="onPause" @onFinished="onFinished" @onRepeat="onRepeat" @onStep="onStep"
          @onLoadingSuccess="onLoadingSuccess"></leven-svga>
        <button type="primary" @click="logStr = ''">清空日志</button>
        <button type="primary" @click="play">开始播放</button>
        <button type="primary" @click="pause">暂停播放</button>
        <button type="primary" @click="resumePlay">继续播放</button>
        <button type="primary" @click="stop">结束播放</button>
      </uni-card>
    </view>
    <view>
      <uni-card class="uni-card-box" title="日志">
        <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></view>
      </uni-card>
    </view>
  </view>
</template>

<script>
  export default {
    data() {
      return {
        logStr: ""
      }
    },
    methods: {
      // 开始播放
      play() {
        if (this.$refs.refLevenSvg) {
          this.$refs.refLevenSvg.play({
            path: "/static/1.svg",
            loop: 1, //循环播放次数, 0.无限循环, 1.循环一次
          }, res => {
            this.writeLog(JSON.stringify(res));
          });
        }
      }
    }
  }
}
```

```
    }  
  },  
  // 暂停播放  
  pause() {  
    if (this.$refs.refLevenSvg) {  
      this.$refs.refLevenSvg.pause();  
    }  
  },  
  // 继续播放  
  resumePlay() {  
    if (this.$refs.refLevenSvg) {  
      this.$refs.refLevenSvg.resumePlay();  
    }  
  },  
  // 暂停播放  
  stop() {  
    if (this.$refs.refLevenSvg) {  
      this.$refs.refLevenSvg.stop({  
        clear: false  
      }, res => {  
        this.writeLog(JSON.stringify(res));  
      });  
    }  
  },  
  // 加载完成  
  onComplete(e) {  
    this.writeLog(JSON.stringify(e));  
  },  
  // 加载错误  
  onError(e) {  
    this.writeLog(JSON.stringify(e));  
  },  
  // 开始播放  
  onPlay(e) {  
    this.writeLog(JSON.stringify(e));  
  },  
  // 暂停播放  
  onPause(e) {  
    this.writeLog(JSON.stringify(e));  
  },  
  // 播放完成  
  onFinished(e) {  
    this.writeLog(JSON.stringify(e));  
  },  
  // 重复播放  
  onRepeat(e) {  
    this.writeLog(JSON.stringify(e));  
  },  
  // 停止播放  
  onStep(e) {
```

```
        this.writeLog(JSON.stringify(e));
    },
    // 开始加载 (网络地址有效)
    onLoadingStart(e) {
        this.writeLog(JSON.stringify(e));
    },
    // 加载中 (网络地址有效)
    onLoading(e) {
        // this.writeLog(JSON.stringify(e));
    },
    // 加载完成 (网络地址有效)
    onLoadingSuccess(e) {
        this.writeLog(JSON.stringify(e));
    },
    // 写日志
    writeLog(str) {
        // let logStr = uni.$lv.date.format(null, "yyyy-mm-dd hh:MM:ss") + " " + s
        // this.logStr = logStr + this.logStr;
        console.log(str)
    }
}
}
</script>

<style>

</style>
```

更新日志

2023-08-18

首次发布

开始播放

方法名

play

用法

- 用法如下：

```
this.$refs.refLevenSvg.play({
  path: "/static/1.svga",
  loop: 1, //循环播放次数，0.无限循环，1.循环一次
}, res => {
  console.log(res)
});
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
path	String	是	无	播放地址，支持网络地址
loop	Integer	否	0	循环播放次数，0.无限制

播放地址如果是本地地址的话就放到static目录下，如：/static/svg/1.svga

回调

- 示例

```
{
  "data": {},
  "message": "操作成功",
  "code": 0
}
```

- 回调说明：

开始播放

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败，请参考状态码

暂停播放

方法名

pause

用法

- 用法如下：

```
if (this.$refs.refLevenSvg) {  
  this.$refs.refLevenSvg.pause();  
}
```

- 参数说明

无

回调

无

继续播放

方法名

resumePlay

用法

- 用法如下：

```
if (this.$refs.refLevenSvg) {  
  this.$refs.refLevenSvg.resumePlay();  
}
```

- 参数说明

无

回调

无

停止播放

方法名

stop

用法

- 用法如下：

```
this.$refs.refLevenSvga.stop({
  clear: false
}, res => {
  this.writeLog(JSON.stringify(res));
});
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
clear	Boolean	否	无	是否清空画布

回调

- 示例

```
{
  "data": {},
  "message": "操作成功",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败，请参考状态码

加载完成

事件名称

用法

- 用法如下：

- 参数说明

无

回调

- 示例

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

加载错误

事件名称

onError

用法

- 用法如下：

```
onError(e) {  
  this.writeLog(JSON.stringify(e.detail));  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "message": "加载错误"  
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

开始播放

事件名称

onPlay

用法

- 用法如下：

```
onPlay(e) {  
  this.writeLog(JSON.stringify(e.detail));  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "message": "开始播放"  
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

暂停播放

事件名称

onPause

用法

- 用法如下：

```
onPause(e) {  
  this.writeLog(JSON.stringify(e.detail));  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "message": "暂停播放"  
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

重复播放

事件名称

onRepeat

用法

- 用法如下：

```
onRepeat(e) {
  this.writeLog(JSON.stringify(e.detail));
},
```

- 参数说明

无

回调

- 示例

```
{
  "message": "重复播放"
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

播放进度

事件名称

onStep

用法

- 用法如下：

```
onStep(e) {  
  this.writeLog(JSON.stringify(e.detail));  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "frame": 62,  
  "message": "正在播放",  
  "progress": 0.9545454545454546  
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
frame	Integer	当前播放的帧
progress	Float	当前播放的进度

播放结束

事件名称

onFinished

用法

- 用法如下：

```
onFinished(e) {  
  this.writeLog(JSON.stringify(e.detail));  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "message": "播放结束"  
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

网络地址开始加载

事件名称

onLoadingStart

用法

- 用法如下：

```
onLoadingStart(e) {  
  this.writeLog(JSON.stringify(e.detail));  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "message": "开始加载网络文件"  
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

网络地址加载中

事件名称

onLoading

用法

- 用法如下：

```
onLoading(e) {  
  this.writeLog(JSON.stringify(e.detail));  
},
```

- 参数说明

无

回调

- 示例

```
{  
  "percent": 0.998543014900872,  
  "message": "网络地址加载中",  
  "size": 1622906,  
  "totalSize": 1625274  
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
size	Integer	当前已加载的大小
percent	Float	当前加载的进度
totalSize	Integer	文件总大小

网络地址加载完成

- 用法如下：

- 参数说明

无

- 示例

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
cachePath	Integer	网络文件本地缓存地址