

# 项目申请书

项目名称: TinyVue组件增加全局配置动效方案

项目主导师: 吴义平

申请人: 程锴

日期: 2025.06.06

邮箱: 1170158548@qq.com

## 一、项目背景

- 1.项目基本需求
- 2.项目相关仓库

## 二、技术方法及可行性

1. Vue相关
2. 组件库相关
3. TypeScript, CSS及其他相关

## 三、项目实施细节梳理

1. 全局动效配置设计
2. 动效函数库 (支持 JS 与 Less)
  - JS 动效工具函数
  - Less 通用类支持
3. 在组件中集成动效
4. 支持动态调整配置
5. 自动化测试用例 (使用 Vitest + Vue Test Utils)

## 四、规划

1. 项目研发第一阶段 (07 月 01 日 - 08 月 15 日) :
2. 项目研发第二阶段 (08 月 16 日 - 09 月 30 日) :
3. 期望:

# 一、项目背景

## 1.项目基本需求

项目仓库地址: <https://github.com/opentiny/tiny-vue>

1. 定义全局动效配置, 需要定义一个全局的动效配置对象, 这个对象可以包含各种动效相关的属性, 比如动画类型、持续时间、延迟等。

2. 创建动效函数库，创建一些通用的动效函数（js函数或者less通用类），这些函数可以根据全局配置来执行特定类型的动画。例如，你可以为淡入淡出、滑动和缩放等常见动画创建函数。
3. 组件集成，每个组件都可以根据需要调用上述动效函数，并传入全局配置或局部配置。
4. 动态调整配置，可以希望提供一种方式来动态地改变全局动效配置，以便于在不同的场景下使用不同的动画效果。
5. 补充相应的文档和自动化测试用例。
6. 输出TinyVue组件增加全局配置动效方案的介绍文章和视频

## 2.项目相关仓库

uiverse-io: <https://github.com/uiverse-io/galaxy>

vue: <https://github.com/vuejs/>

## 二、技术方法及可行性

### 1. Vue相关

我系统学习并掌握了 Vue 3.0 及其相关技术栈。在此基础上，我参加了蓝桥杯软件赛 Web 开发赛道，并荣获国家级一等奖，积累了丰富的实战经验。因此，我有信心胜任与 Vue 相关的开发任务。

### 2.组件库相关

我曾参加字节跳动青训营 Web 组进阶班，并获得“优秀营员”称号。在最终项目中，我们组队并开发了一个简单的组件库，对组件的封装、复用与文档编写有深入理解，具备一定的组件开发与工程化能力。

### 3.TypeScript,CSS及其他相关

在多个独立项目中，我广泛使用 TypeScript 进行类型安全开发，并具备扎实的 CSS 样式实现能力。此外，我曾在同元软控实习，参与 SysLab 网页端的前端功能开发，进一步巩固了我在前端领域的实际开发能力，对相关技术有较为全面的掌握。

## 三、项目实施细节梳理

### 1.全局动效配置设计

```
// animation-config.ts
export type AnimationType = 'fade' | 'slide' | 'zoom';

export interface GlobalAnimationConfig {
  type: AnimationType;
  duration: number;
```

```

    delay: number;
    easing: string;
}

export let globalAnimationConfig: GlobalAnimationConfig = {
  type: 'fade',
  duration: 300,
  delay: 0,
  easing: 'ease-in-out',
};

// 提供更新接口
export function updateGlobalAnimationConfig(config:
Partial<GlobalAnimationConfig>) {
  globalAnimationConfig = { ...globalAnimationConfig, ...config };
}

```

## 2. 动效函数库（支持 JS 与 Less）

### JS 动效工具函数

```

// animation-utils.ts
import { globalAnimationConfig, GlobalAnimationConfig } from './animation-
config';

export function applyAnimation(
  el: HTMLElement,
  type: 'fade' | 'slide' | 'zoom',
  config?: Partial<GlobalAnimationConfig>
) {
  const { duration, delay, easing } = { ...globalAnimationConfig, ...config };
  el.style.transition = `all ${duration}ms ${easing} ${delay}ms`;

  switch (type) {
    case 'fade':
      el.style.opacity = '1';
      break;
    case 'slide':
      el.style.transform = 'translateY(0)';
      break;
    case 'zoom':
      el.style.transform = 'scale(1)';
      break;
  }
}

```

## Less 通用类支持

```
// animation.less
@duration: 300ms;
@easing: ease-in-out;

.fade-in {
  opacity: 0;
  animation: fadeIn @duration @easing forwards;
}

@keyframes fadeIn {
  to {
    opacity: 1;
  }
}
```

## 3.在组件中集成动效

```
// ExampleComponent.vue
<template>
  <div ref="box" class="animated-box">内容</div>
</template>

<script lang="ts" setup>
import { onMounted, ref } from 'vue';
import { applyAnimation } from '@/utils/animation-utils';

const box = ref<HTMLElement | null>(null);

onMounted(() => {
  if (box.value) {
    applyAnimation(box.value, 'fade');
  }
});
</script>

<style scoped>
.animated-box {
  opacity: 0;
}
</style>
```

## 4.支持动态调整配置

```
// 动态设置全局动画
import { updateGlobalAnimationConfig } from '@utils/animation-config';

updateGlobalAnimationConfig({
  duration: 500,
  easing: 'ease-out',
});
```

## 5.自动化测试用例（使用 Vitest + Vue Test Utils）

```
import { mount } from '@vue/test-utils';
import ExampleComponent from '../ExampleComponent.vue';

describe('ExampleComponent', () => {
  it('should apply fade animation on mount', async () => {
    const wrapper = mount(ExampleComponent);
    const el = wrapper.find('.animated-box').element;
    await new Promise((resolve) => setTimeout(resolve, 400)); // 等待动画结束
    expect(el.style.opacity).toBe('1');
  });
});
```

## 四、规划

因为是保研，大四毕业这个暑假空闲时间非常多，可以保证较快的进度，也希望通过参与开源的项目自己能学到很多新的东西。

### 1.项目研发第一阶段（07月01日 - 08月15日）：

- 完成全局动效配置对象的设计和封装，支持类型、时长、延迟、缓动等基础参数，以及 loop、repeat 等扩展字段
- 完成动效函数库开发，支持 JS 动效函数和 Less 通用类，适配常见的淡入淡出、滑动、缩放动画
- 实现组件动效集成，在组件挂载时应用全局配置的动画效果，并支持局部覆盖
- 补充基础自动化测试用例，验证动效逻辑正确性
- 编写文档说明各动效函数的用法和配置参数

## 2.项目研发第二阶段（08 月 16 日 - 09 月 30 日）：

- 解决在中期验收阶段中发现的问题
- 对第一阶段完成的内容进行更详细的测试
- 对第一阶段的完成内容进行总结，并输出相关文档内容
- 思考可以改进或者补充的地方

## 3. 期望：

希望通过这个项目，把自己之前零散接触的动效知识系统化，真正从动效配置设计到函数封装再到组件集成，完整实践一遍，提升前端动画系统设计的能力。同时也希望借这个机会锻炼代码抽象能力和可复用性思维，为以后在团队项目或开源项目中设计更通用的工具打下基础。

另外我对前端动效和用户体验方面一直挺感兴趣，也想在这个过程中尝试接触一些更复杂的动画场景（比如组合动画、入场离场动画等），看看能不能把动效系统做得更灵活一些。项目结束后，也希望有机会继续优化和维护这套动效方案，作为长期可用的工具积累下来。