

微信小程序定时器回收



本文用于描述 uni-app 中定时器的回收以及设计过程

一、背景

定时器是全局的，并不是跟页面绑定的，当小程序从一个页面路由到另一个页面之后，前一个页面定时器应注意手动回收。

问题(场景):

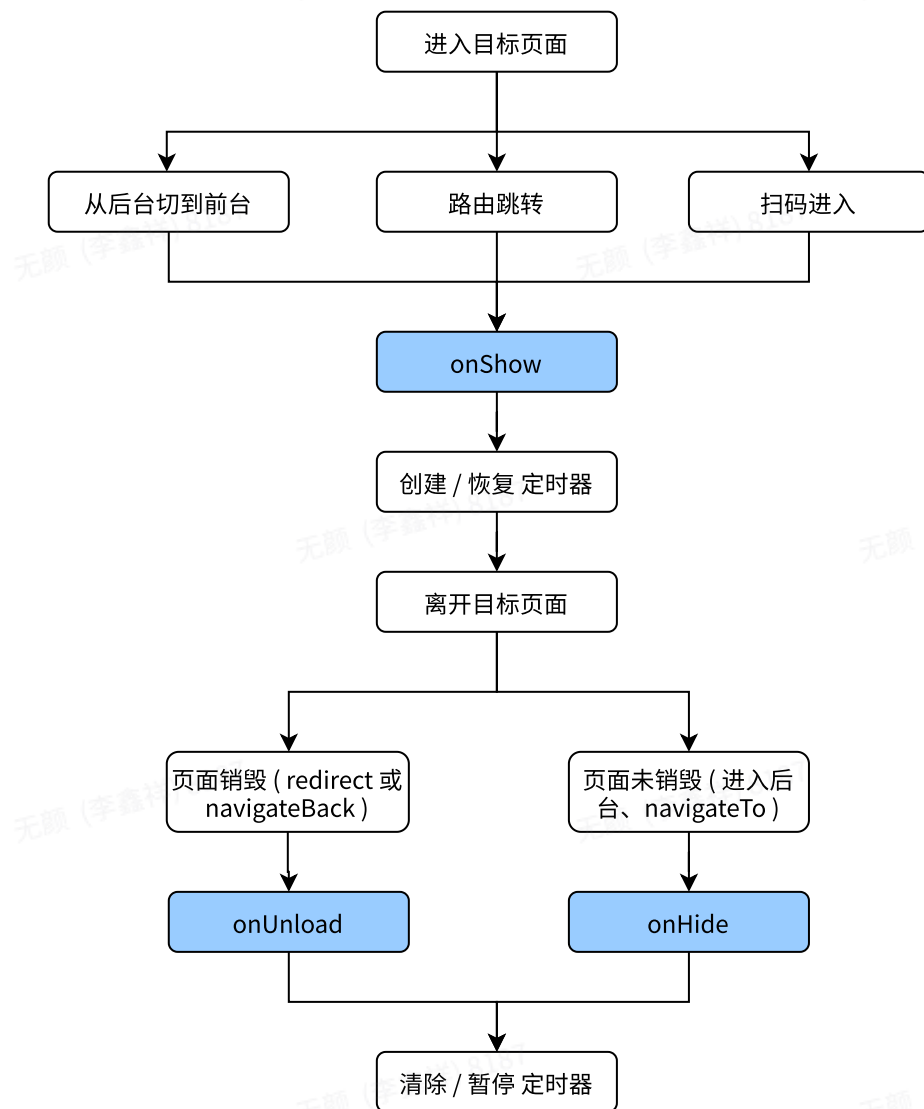
- 页面隐藏在后台时，定时器需手动回收
- 页面再次显示时需恢复原有定时器
- 页面销毁后，定时器需手动回收

目标:

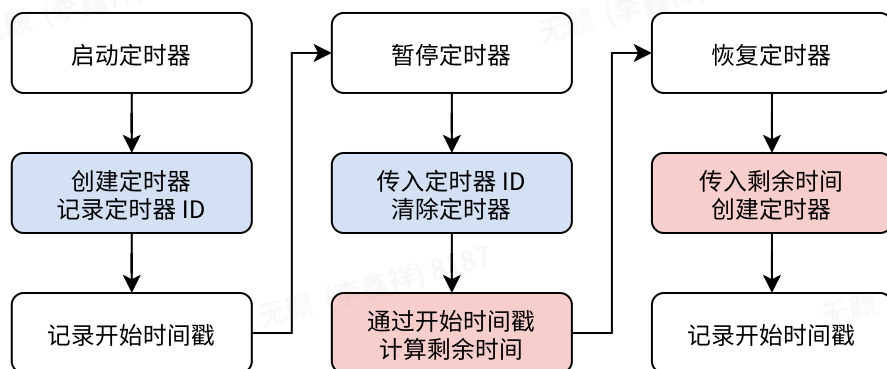
- 实现定时器的暂停与恢复功能
- 统一页面、组件中的定时器回收方式

二、关键流程

结合背景与微信小程序实际场景，关键流程描述如下:



启动、暂停、恢复流程：



注意：以上流程适用于页面、组件的定时器回收。

三、关键技术

流程中涉及到的关键技术主要是定时器、生命周期，**需要特别注意 uni-app 与微信小程序原生的区别。**

1. 定时器

流程中使用的所有定时器依赖于微信小程序提供的 setTimeout、clearTimeout、setInterval、clearInterval。其中 setInterval 可通过 setTimeout 来实现。

参见 [微信小程序定时器官方文档](#)

2. 生命周期

流程中需要在对应的生命周期中回收定时器，其中涉及 uni-app 中页面和组件的生命周期。

页面生命周期：

- onShow
- onHide
- onUnload

组件生命周期：

- onPageShow
- onPageHide

注意： uni-app 自定义组件中没有 onShow、onHide 钩子，参见 [微信小程序生命周期说明](#)

3. Mixin

统一项目中的定时器回收方式，在 uni-app 中可以选择 Mixin。它提供了一种非常灵活的方式，来分发 Vue 组件中的可复用功能。

参加 [Vue Mixin 官方文档](#)

四、功能划分

功能分为两部分，Timer 类负责实现定时器基本功能，Mixin 用于组件、页面生命周期中统一管理定时器的回收。

Timer 类：

- 管理定时器参数（callback、timeout 等）
- 提供启动、暂停方法（start、stop）
- 支持 setTimeout、setInterval（isInterval）

Mixin：

- 在对应生命周期（created、destroyed、onShow、onHide）调用 Timer 实例提供方法
- 管理多个定时器

五、具体实现

这里仅呈现伪代码，想深入了解的可以看 [源码](#)

1. Timer：

JavaScript

```
1  class Timer {
2      static id = 0
3      constructor (isInterval = false, fn = () => {}, timeout = 0, ...arg) {
4          this.id = ++Timer.count // 定时器递增 id
5          // 初始化参数
6      }
7      start() {
8          // 记录开始时间
9          // 调用 setTimeout API 实现 setTimeout 或 setInterval
10     }
11     stop () {
12         // 计算剩余时间
13         // 清除计时器
14     }
15 }
```

2. TimerMixin：

JavaScript

```
1 export default {
2   created(){} // 初始化定时器数组
3   destroyed(){} // 暂定所有定时器
4   onShow(){} // 恢复所有定时器
5   onHide(){} // 暂定所有定时器
6   onUnload(){} // 清除所有定时器
7   onPageShow(){} // 恢复所有定时器
8   onPageHide(){} // 暂定所有定时器
9   methods:{
10     setTimeout(){} // 创建定时器, 新增
11     clearTimeout(){} // 清除定时器, 移除
12     setInterval(){} // 创建定时器, 新增
13     clearInterval(){} // 清除定时器, 移除
14   }
15 }
```

六、使用方式

最终以 mixin 的方式提供给项目使用

具体使用方式如下:

JavaScript

```
1 import timerMixin from '@mixin/timerMixin'
2
3 export default {
4   mixins:[timerMixin],
5   created(){
6     const timer1 = this.setTimeout(() => {
7       console.log('setTimeout')
8     })
9     this.clearTimeout(timer1)
10
11     const timer2 = this.setInterval(() => {
12       console.log('setInterval')
13     })
14     this.clearInterval(timer2)
15   }
16 }
```