

setTimeout



```
// DaysCounter.vue
methods: {
  ...mapMutations(['increaseDayCount']),
  startGameLoop () {
    this.startDayTimer()
  },
  startDayTimer () {
    const day = 1000 * 60
    this.loop = setTimeout(() => {
      if (this.isActive) {
        this.increaseDayCount()
        this.startDayTimer()
      }
    }, day)
  }
}
```



```
// Stats.vue
methods: {
  ...mapMutations(['decrease']),
  startGameLoop () {
    this.decreaseStats()
  },
  decreaseStats () {
    const decreaseInterval = 12 * 1000
    this.loop = setTimeout(() => {
      if (this.isActive) {
        // decrease all stats

        this.decreaseStats()
      }
    }, decreaseInterval)
  }
}
```



```
// ...
mounted () {
  EventBus.$on('gameStatusChange', this.handleGameStatusChange)
  this.startGameLoop()
},
beforeDestroy () {
  this.resetGameLoop()
},
methods: {
  startGameLoop () {
    // to be overwritten by component
  },
  resetGameLoop () {
    clearTimeout(this.loop)
    this.loop = null
  },
  handleGameStatusChange (isPaused) {
    if (isPaused) {
      this.resetGameLoop()
    } else {
      this.startGameLoop()
    }
  }
}
```

Mixin

- Reutilizar funcionalidades entre componentes
- Permite setear data, lifecycle hooks, propiedades computadas...
- *Merge* con las propiedades definidas en el componente.

[Documentación](#)