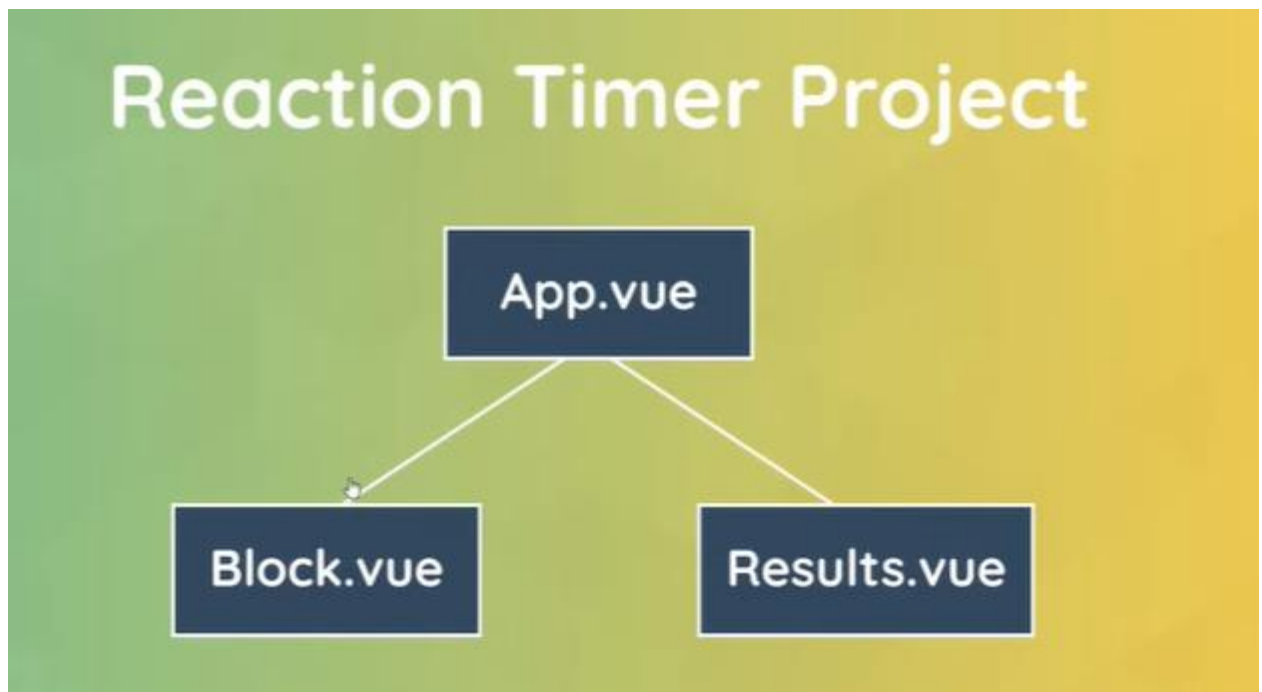
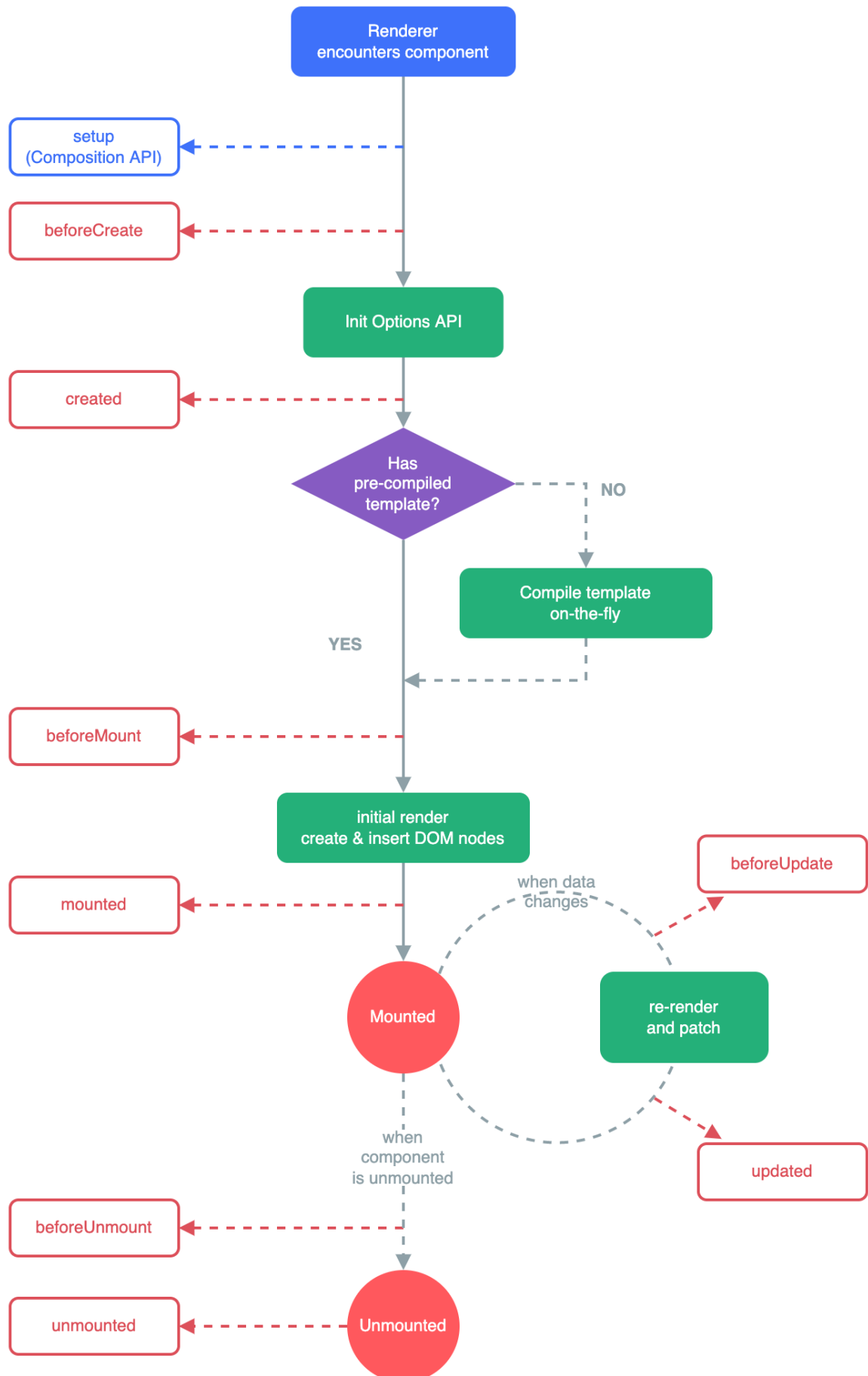


Reaction Timer Project



Life Cycle Hooks



```
▼ App.vue M    ▼ Block.vue U ✕    ▼ Results.vue U
src > components > ▼ Block.vue > {} "Block.vue" > script > [🔗] default > 📦
11      return {
12        showBlock: false
13      }
14    },
15    mounted() {
16      console.log('components mounted')
17      setTimeout(() => {
18        this.showBlock = true
19        console.log(this.delay)
20      }, this.delay)
21    },
22    updated() {
23      console.log('component updated')
24    },
25    unmounted(){
26      console.log('unmounted')
27    }
28  }
29  </script>
30
```

Creating a timer

```
▼ App.vue M    ▼ Block.vue U ×    ▼ Results.vue U
src > components > ▼ Block.vue > {} "Block.vue" > script > default > methods > stopTimer
9      props: ['delay'],
10     data(){
11       return {
12         showBlock: false,
13         timer: null,
14         reactionTime: 0
15       }
16     },
17     mounted() {
18       console.log('components mounted')
19       setTimeout(() => {
20         this.showBlock = true
21         this.startTimer()
22       }, this.delay)
23     },
24     methods: {
25       startTimer() {
26         this.timer = setInterval(() => {
27           this.reactionTime += 10
28         }, 10)
29       },
30       stopTimer() {
31         clearInterval(this.timer)
32         console.log(this.reactionTime)
33       }
34     }
35   }
```

Custom Events with Data

```
App.vue M X
src > App.vue > {} "App.vue" > script > default > methods > start
You, 48 seconds ago | 1 author (You)
1 <template>
2   <h1>Reaction Timer</h1>
3   <button @click="start" :disabled="isPlaying">Play</button>
4   <Block v-if="isPlaying" :delay="delay" @end="endGame"/>
5   <p v-if="showResults">Reaction Time: {{ score }} ms</p>
6 </template>
7
8 <script>
9 import Block from './components/Block.vue'
10 export default {
11   name: 'App',
12   components: { Block },
13   data(){
14     return {
15       isPlaying: false,
16       delay: null,
17       score: null,
18       showResults: false
19     }
20   },
21   methods: {
22     start(){
23       this.delay = 2000 + Math.random() * 5000
24       this.isPlaying = true
25       this.showResults = false
26     },
27     endGame(reactionTime){
28       this.score = reactionTime
29       this.isPlaying = false
30       this.showResults = true
31     }
32   }
33 }
34 </script>

Blockvue U X
src > components > Blockvue > {} "Blockvue" > script > default > metho
1 <template>
2   <div class="block" v-if="showBlock" @click="stopTimer">
3     Click Me
4   </div>
5 </template>
6
7 <script>
8 export default {
9   props: ['delay'],
10  data(){
11    return {
12      showBlock: false,
13      timer: null,
14      reactionTime: 0
15    }
16  },
17  mounted() {
18    console.log('components mounted')
19    setTimeout(() => {
20      this.showBlock = true
21      this.startTimer()
22    }, this.delay)
23  },
24  methods: {
25    startTimer() {
26      this.timer = setInterval(() => {
27        this.reactionTime += 10
28      }, 10)
29    },
30    stopTimer() {
31      clearInterval(this.timer)
32      this.$emit('end', this.reactionTime)
33    }
34  }
35 }
```