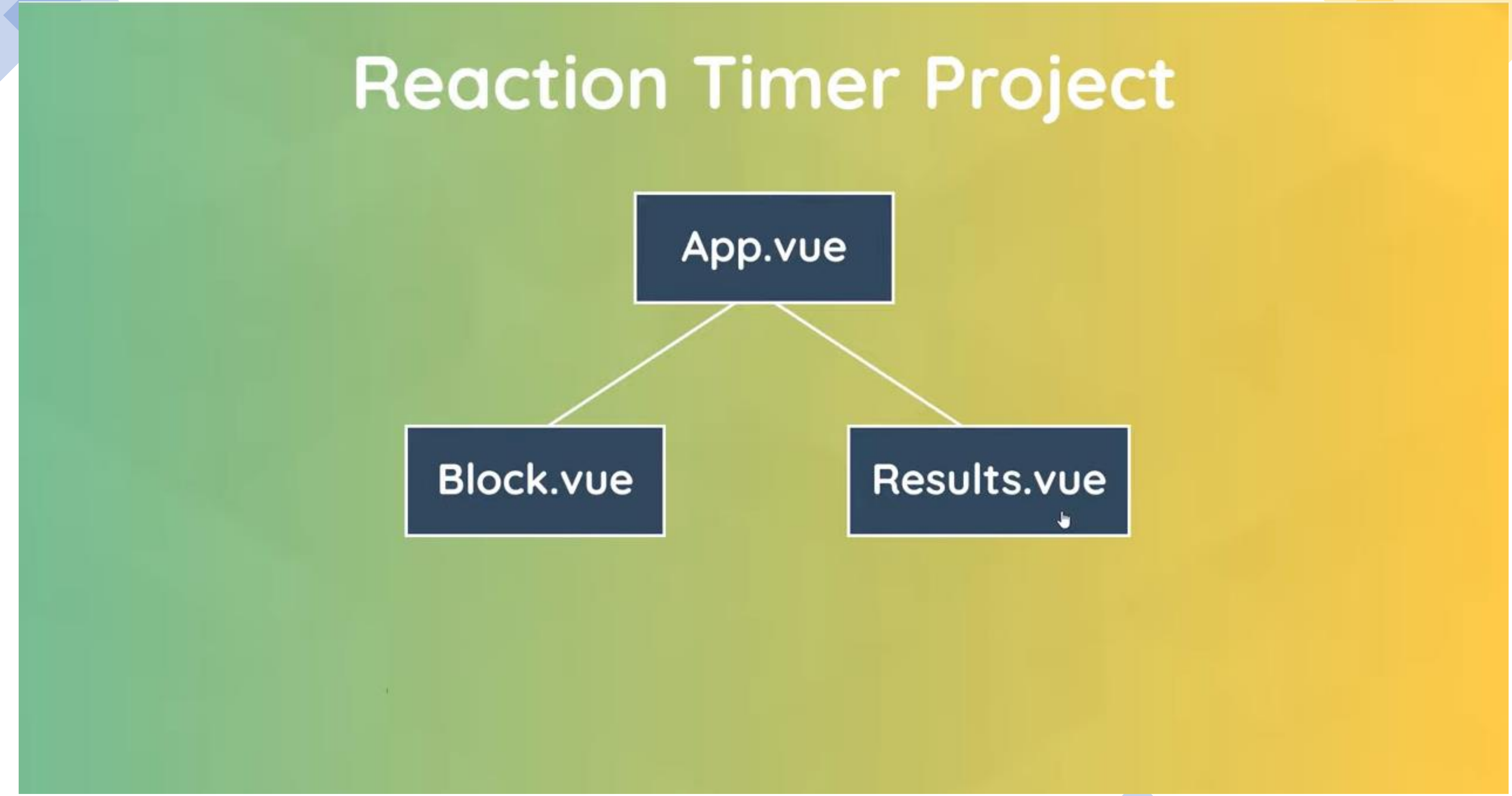
Build a Reaction Timer Game

In this Vue 3 tutorial we'll build a reaction timer game from scratch using everything that we've learnt about Vue.js so far - props, events, data, methods, etc. We'll also learn about lifecycle methods too.



Reaction Timer Project

App.vue

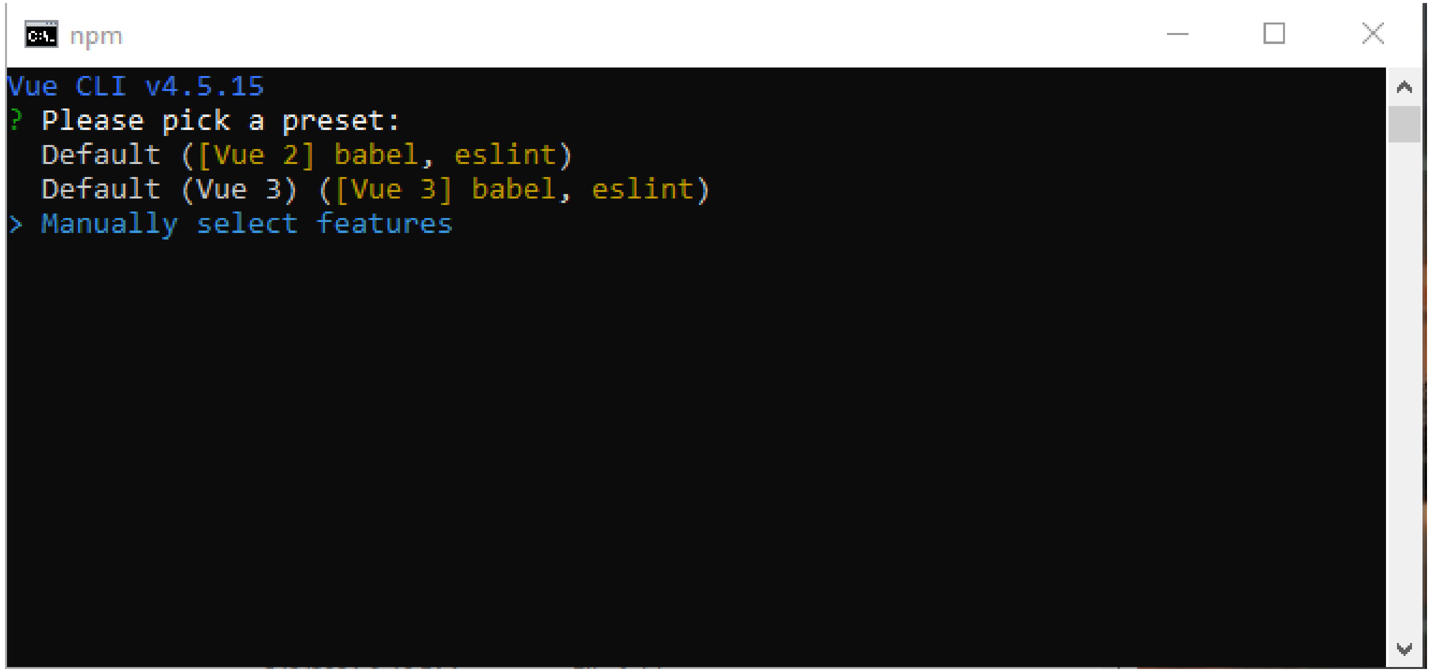
Block.vue

Results.vue

Open cmd and write next commands:

- vue create reaction-timer

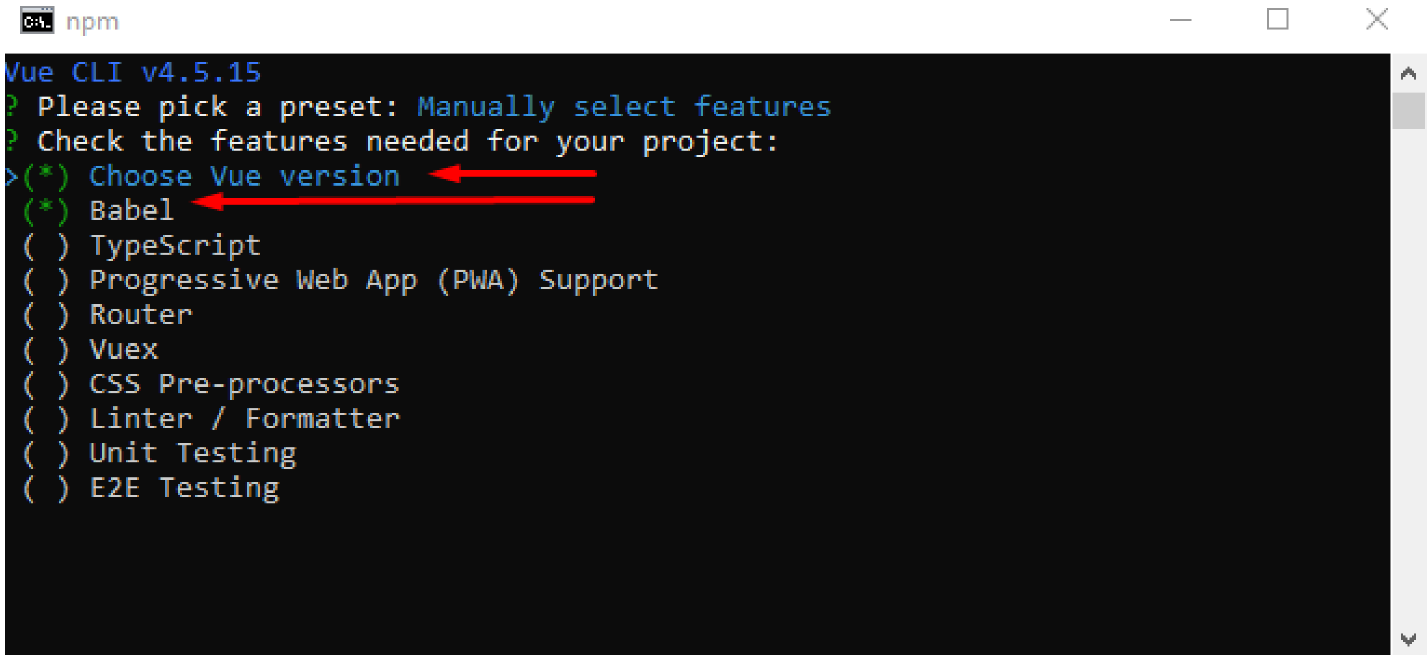
Choose row 'Manually select features' and click 'Enter'



```
Vue CLI v4.5.15
? Please pick a preset:
  Default ([Vue 2] babel, eslint)
  Default (Vue 3) ([Vue 3] babel, eslint)
> Manually select features
```

Select
Choose Vue version
Babel

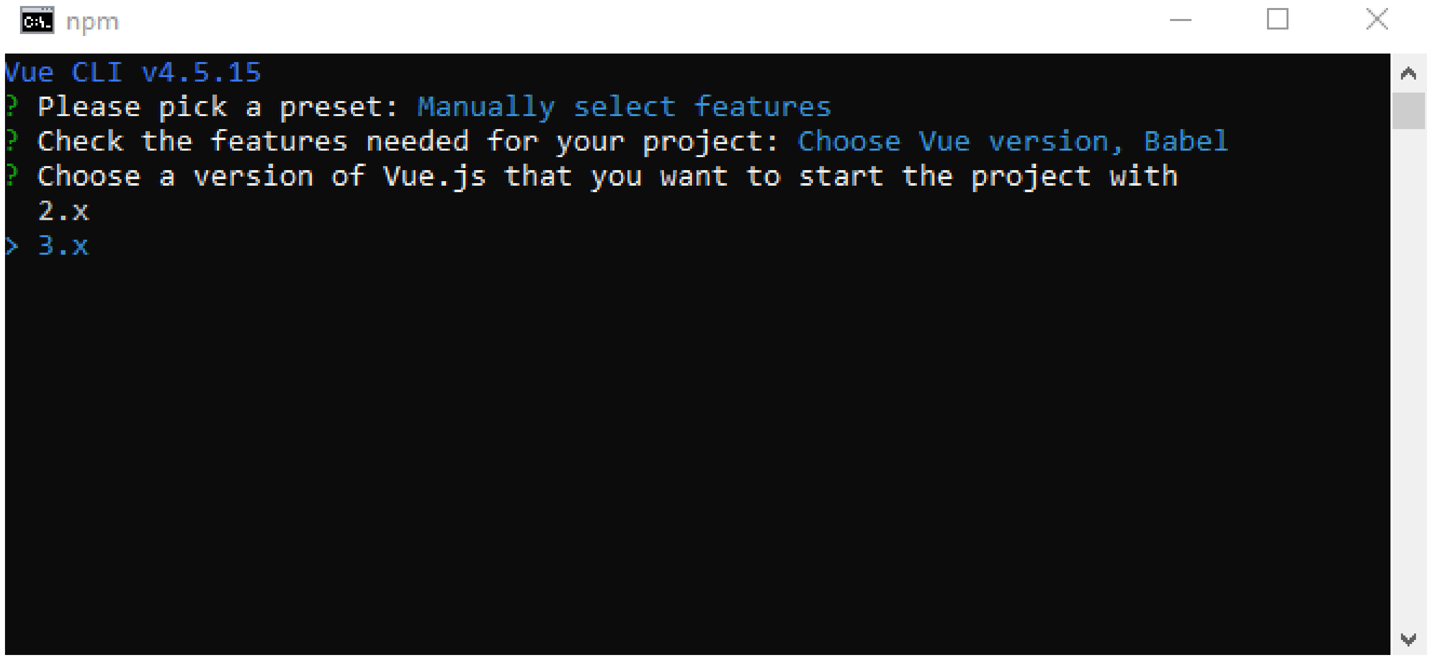
Click 'Enter'



```
Vue CLI v4.5.15
? Please pick a preset: Manually select features
? Check the features needed for your project:
>(*) Choose Vue version
 (*) Babel
 ( ) TypeScript
 ( ) Progressive Web App (PWA) Support
 ( ) Router
 ( ) Vuex
 ( ) CSS Pre-processors
 ( ) Linter / Formatter
 ( ) Unit Testing
 ( ) E2E Testing
```

Select 3.x

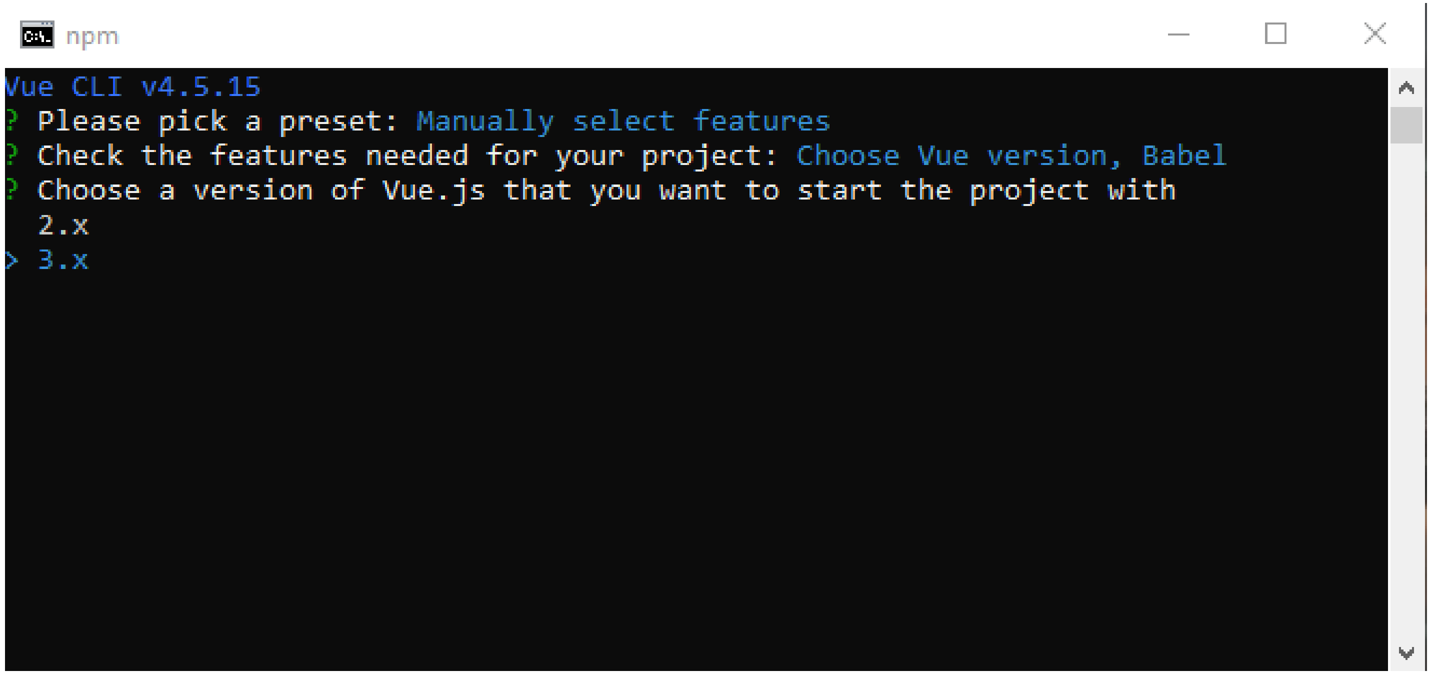
Click 'Enter'



```
Vue CLI v4.5.15
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel
? Choose a version of Vue.js that you want to start the project with
  2.x
> 3.x
```

Select 3.x

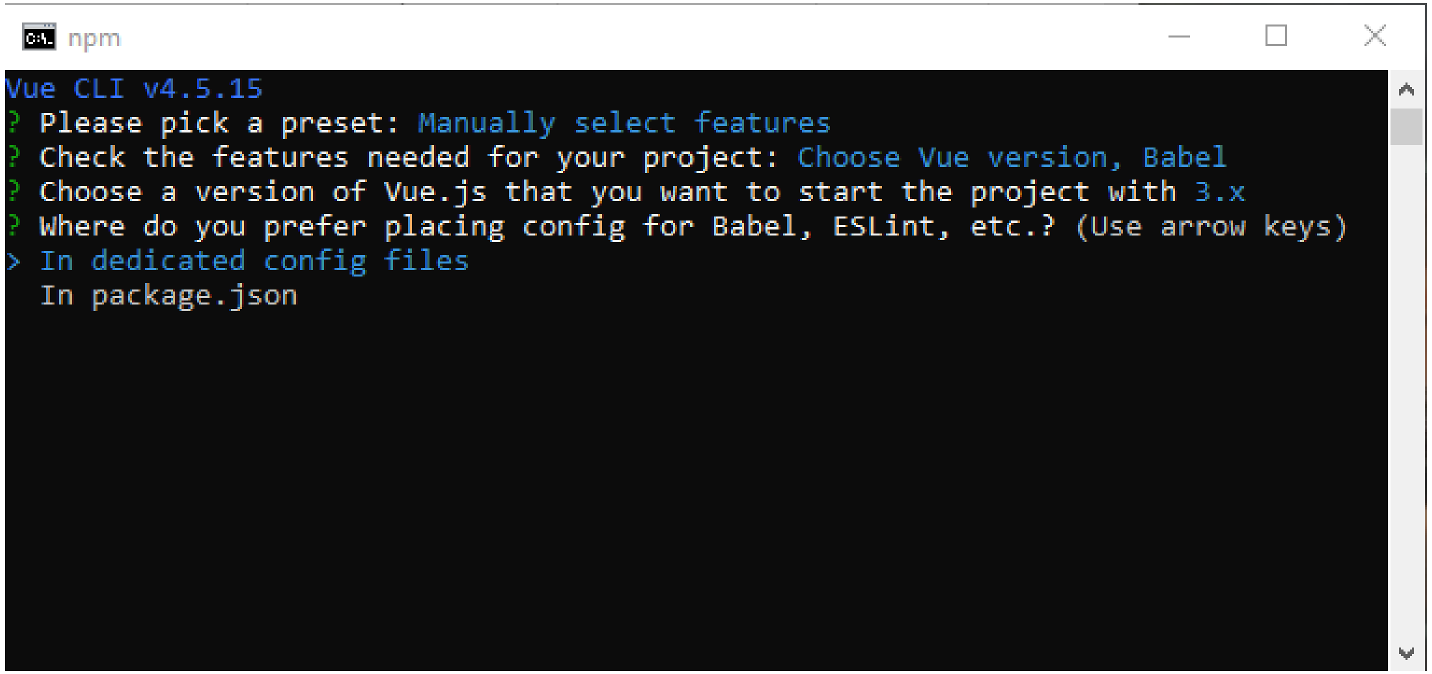
Click 'Enter'



```
Vue CLI v4.5.15
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel
? Choose a version of Vue.js that you want to start the project with
  2.x
> 3.x
```

Select row 'In dedicated config files'

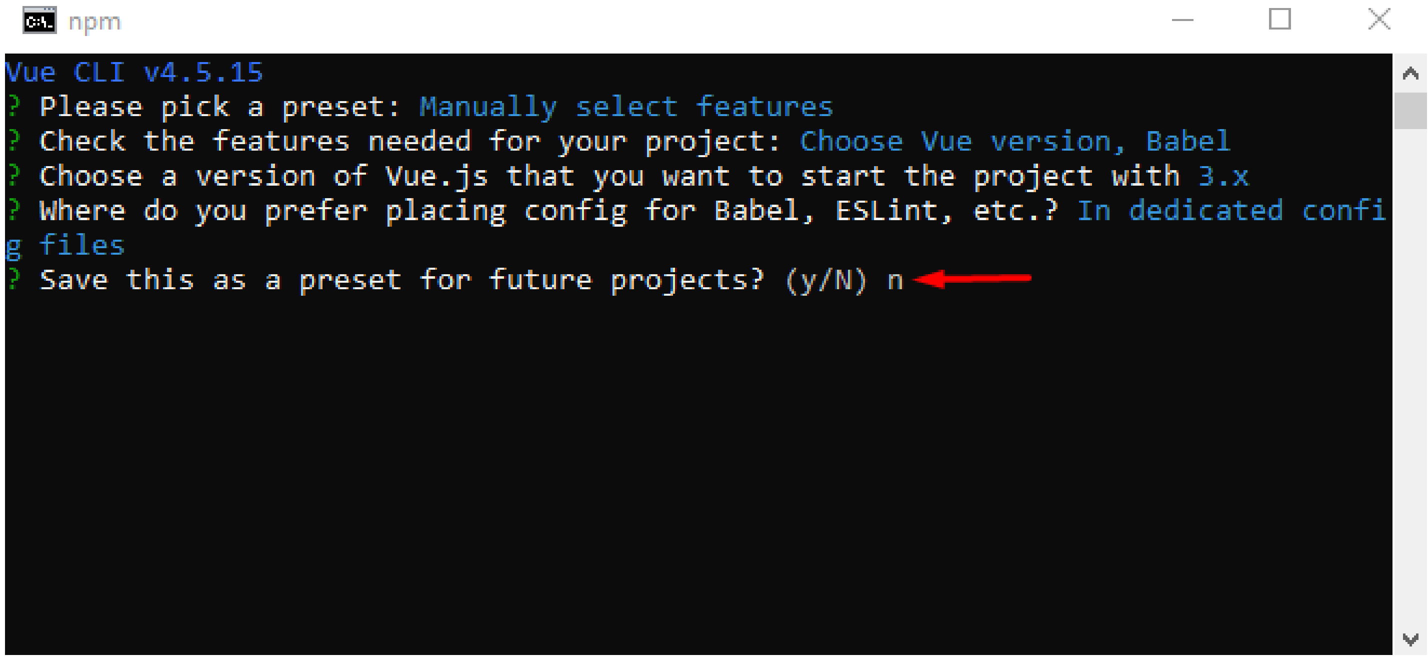
Click 'Enter'



```
Vue CLI v4.5.15
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel
? Choose a version of Vue.js that you want to start the project with 3.x
? Where do you prefer placing config for Babel, ESLint, etc.? (Use arrow keys)
> In dedicated config files
  In package.json
```

Write 'n'

Click 'Enter'



```
Vue CLI v4.5.15
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel
? Choose a version of Vue.js that you want to start the project with 3.x
? Where do you prefer placing config for Babel, ESLint, etc.? In dedicated confi
g files
? Save this as a preset for future projects? (y/N) n
```

write next commands in cmd:

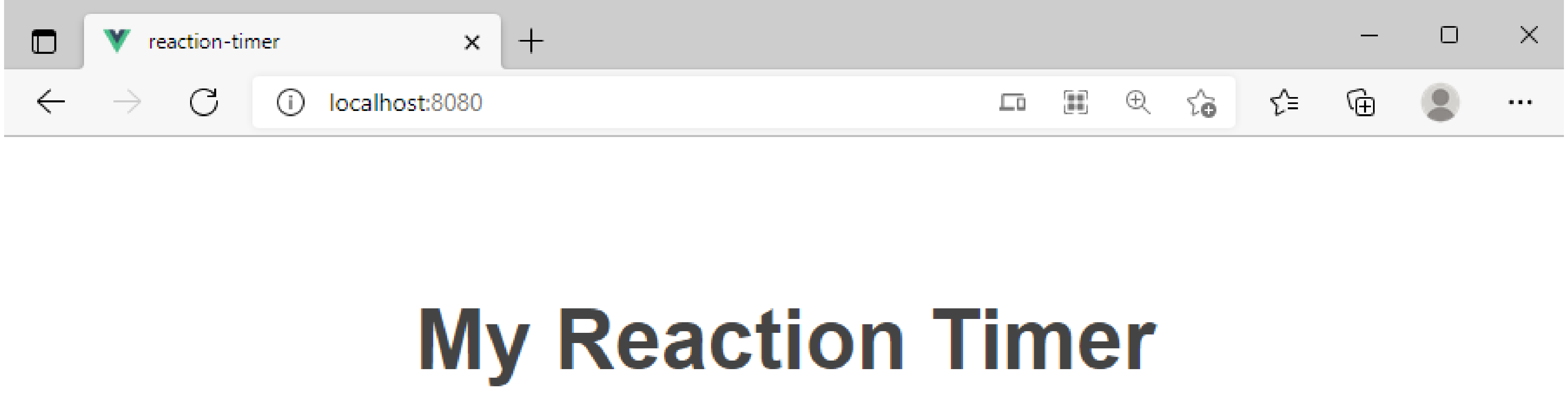
- cd reaction-timer
- code .

- Delete file HelloWorld.vue from folder 'components'
- Create file Block.vue in the folder 'components'
- Create file Results.vue in the folder 'components'
- Edit file App.vue

App.vue

```
<template>
  <h1>My Reaction Timer</h1>

</template>

<script>
export default {
  name: 'App',
  components: {},
}
</script>

<style>
  #app {
    font-family: Avenir, Helvetica, Arial, sans-serif;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
    text-align: center;
    color: #444;
    margin-top: 60px;
  }
</style>
```

write commands in cmd: npm run serve and go to http://localhost:8080/

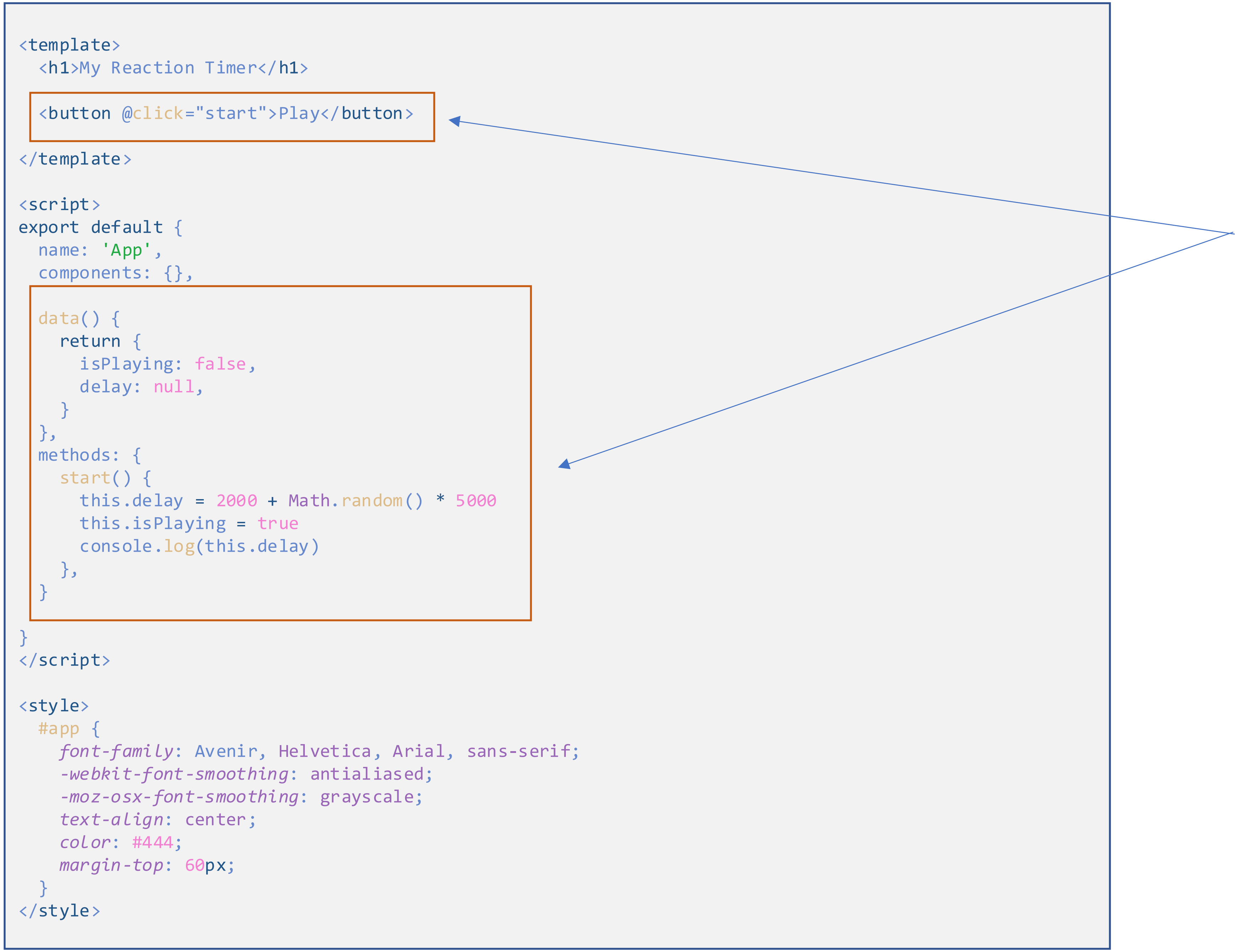


My Reaction Timer

Starting a New Game

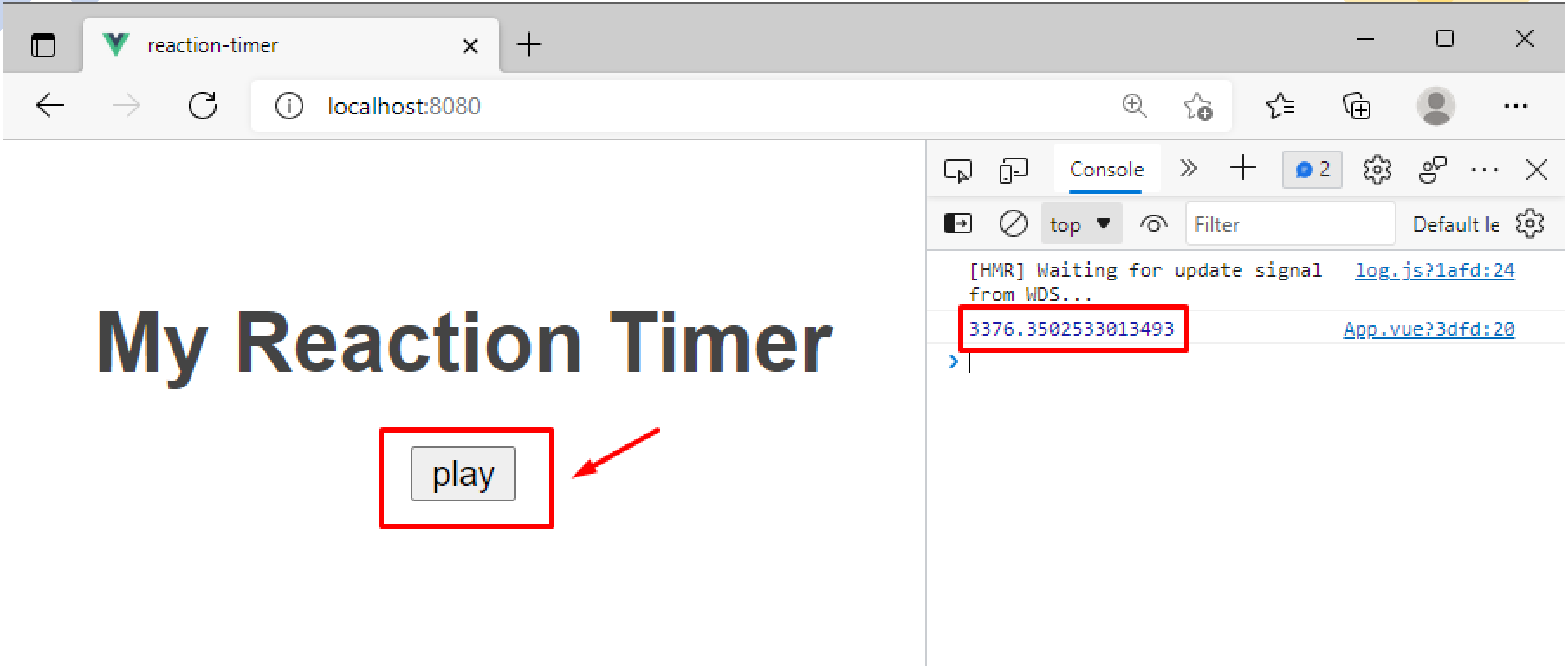
Edit file App.vue

App.vue



```
<template>
  <h1>My Reaction Timer</h1>

  <button @click="start">Play</button>

</template>

<script>
export default {
  name: 'App',
  components: {},

  data() {
    return {
      isPlaying: false,
      delay: null,
    }
  },
  methods: {
    start() {
      this.delay = 2000 + Math.random() * 5000
      this.isPlaying = true
      console.log(this.delay)
    },
  }

}
</script>

<style>
  #app {
    font-family: Avenir, Helvetica, Arial, sans-serif;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
    text-align: center;
    color: #444;
    margin-top: 60px;
  }
</style>
```

Add code



```
[HMR] Waiting for update signal    log.js?1afd:24
from WDS...
3376.3502533013493                 App.vue?3dfd:20
```

Edit file Block.vue

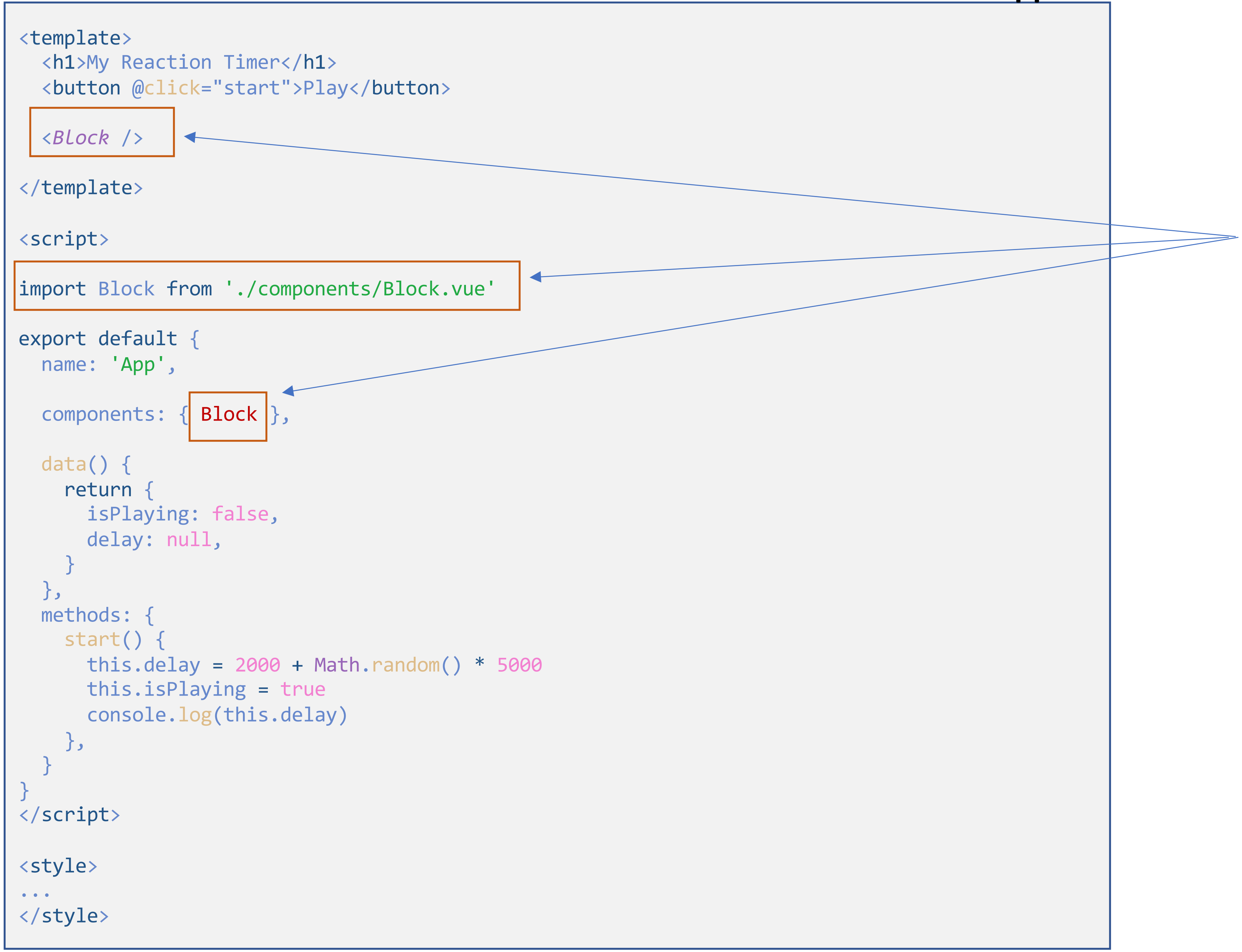
Block.vue

```
<template>
  <div class="block">
    Click Me
  </div>
</template>

<script>
export default {
  props: ['delay']
}
</script>

<style>
  .block {
    width: 400px;
    border-radius: 20px;
    background:  #0faf87;
    color: white;
    text-align: center;
    padding: 100px 0;
    margin: 40px auto;
  }
</style>
```

Add
code

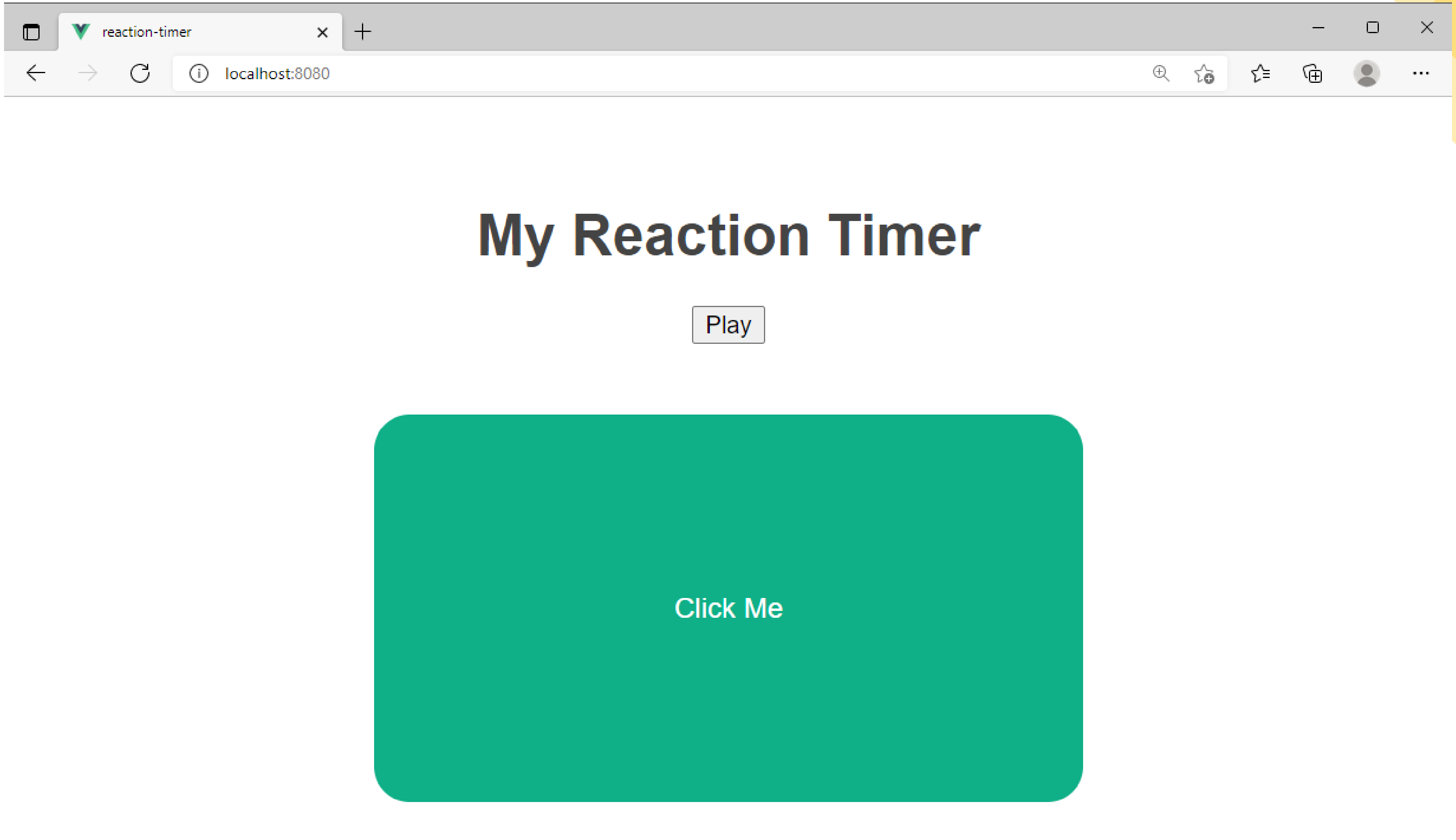
Edit file App.vue

App.vue



```
<template>
  <h1>My Reaction Timer</h1>
  <button @click="start">Play</button>

  <Block />

</template>

<script>

import Block from './components/Block.vue'

export default {
  name: 'App',

  components: { Block },

  data() {
    return {
      isPlaying: false,
      delay: null,
    }
  },
  methods: {
    start() {
      this.delay = 2000 + Math.random() * 5000
      this.isPlaying = true
      console.log(this.delay)
    },
  }
}
</script>

<style>
...
</style>
```

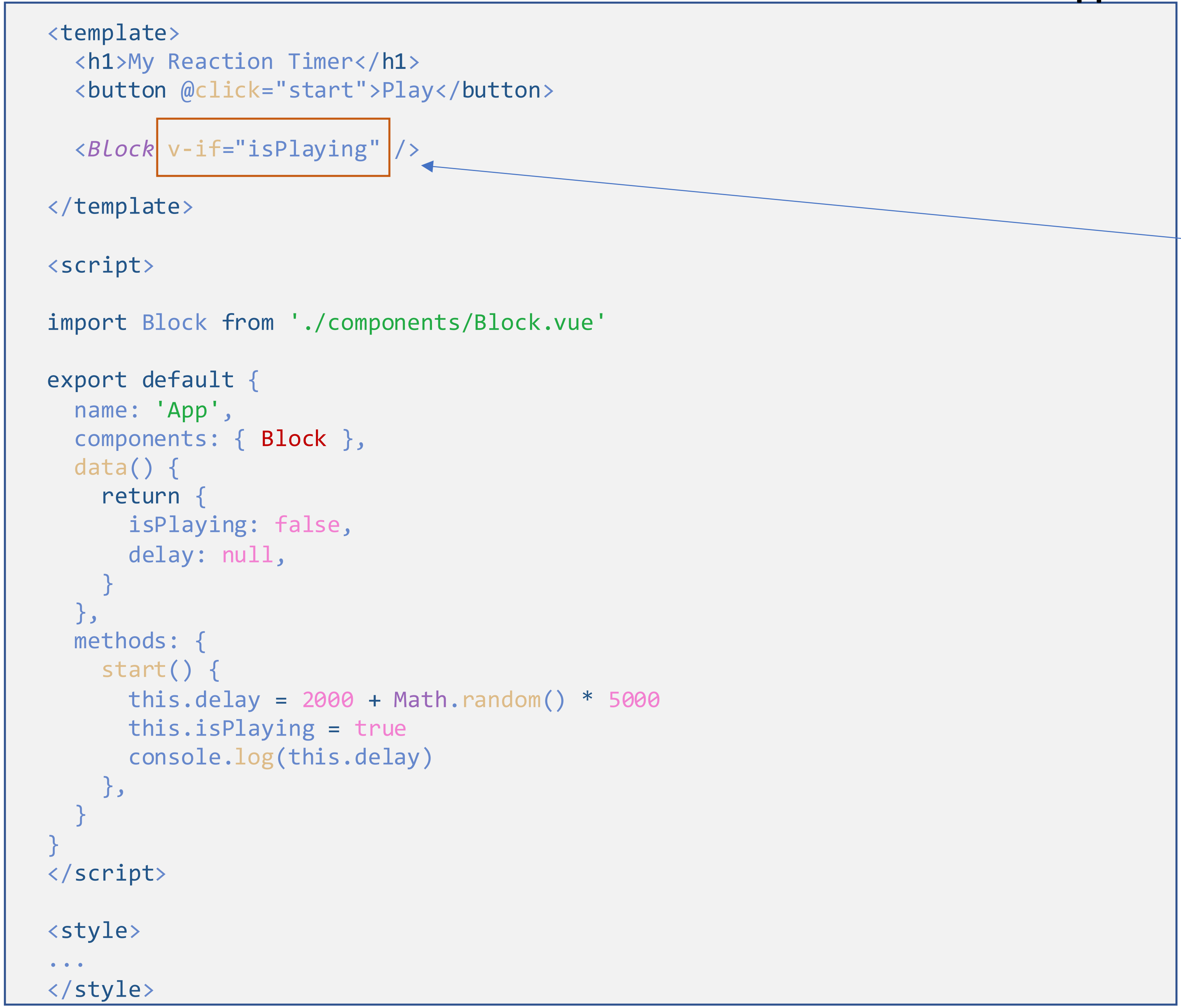
Add code



reaction-timer
localhost:8080
My Reaction Timer
Play
Click Me

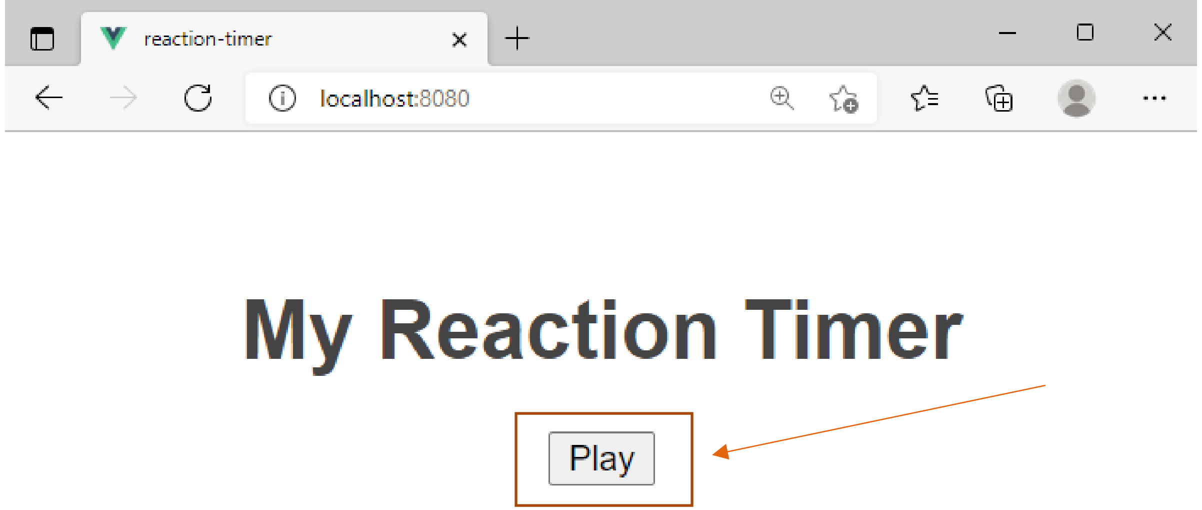
Edit file App.vue

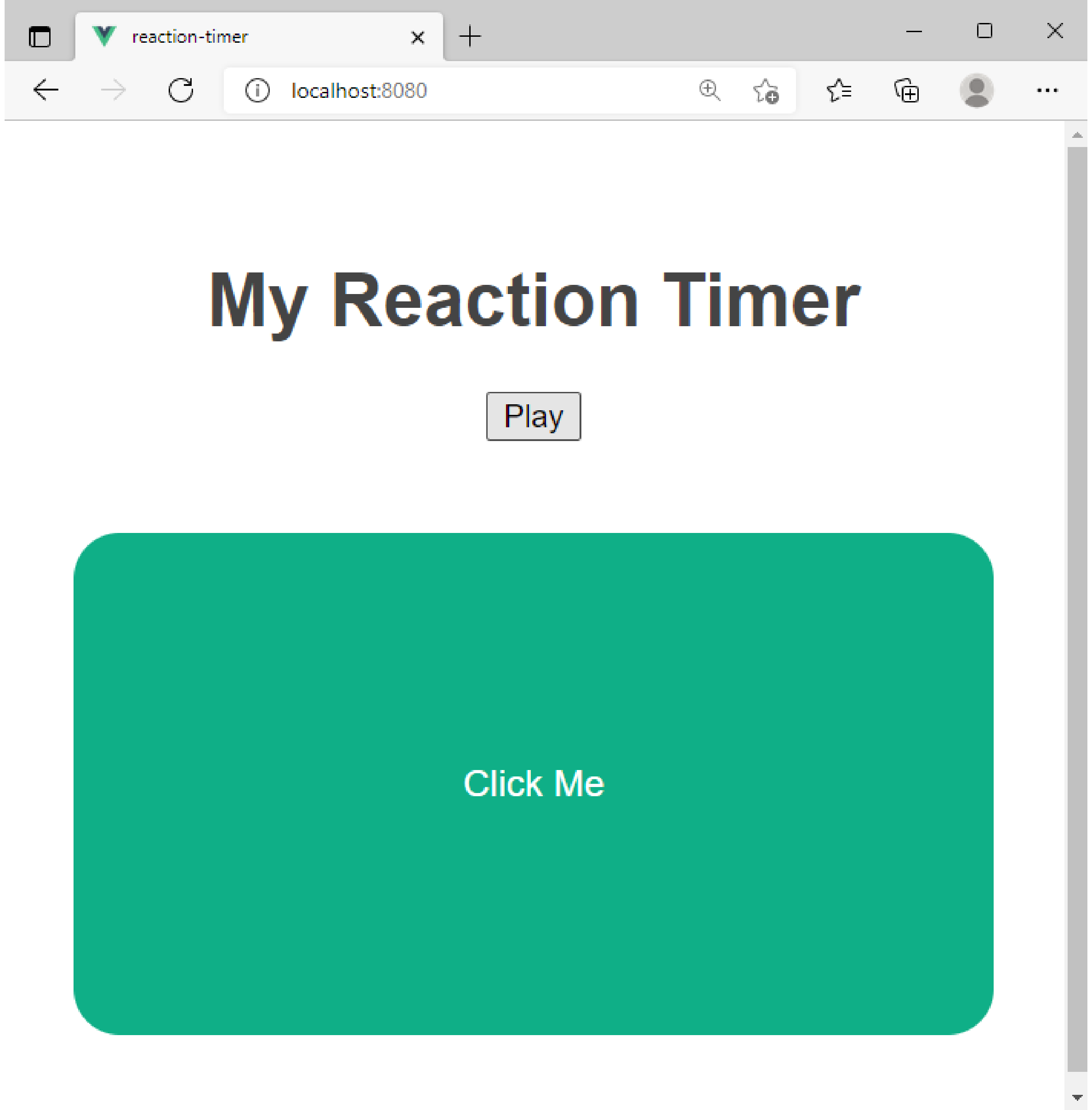
App.vue



```
<template>
  <h1>My Reaction Timer</h1>
  <button @click="start">Play</button>

  <Block v-if="isPlaying" />

</template>

<script>

import Block from './components/Block.vue'

export default {
  name: 'App',
  components: { Block },
  data() {
    return {
      isPlaying: false,
      delay: null,
    }
  },
  methods: {
    start() {
      this.delay = 2000 + Math.random() * 5000
      this.isPlaying = true
      console.log(this.delay)
    },
  }
}
</script>

<style>
...
</style>
```

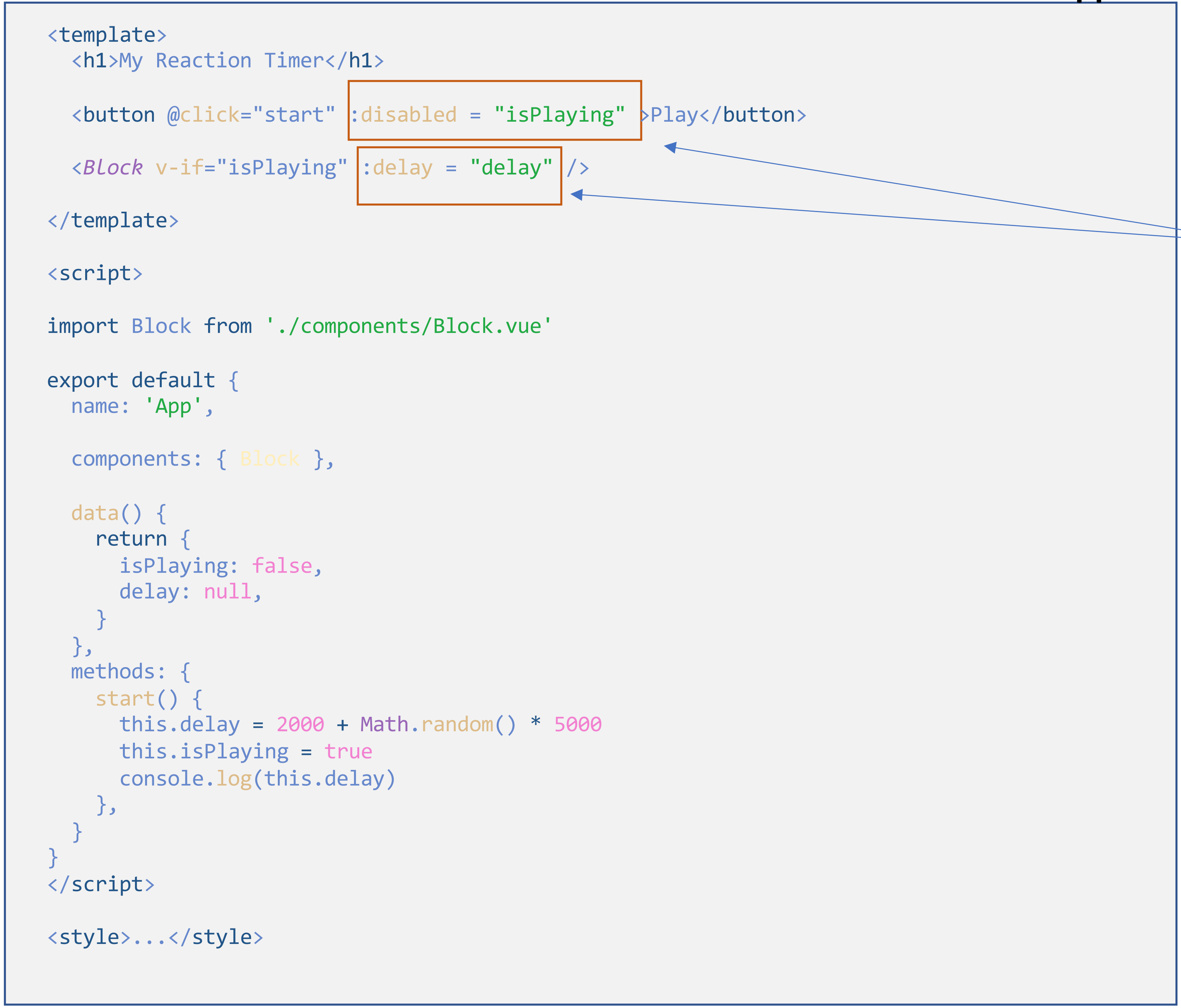
Add
code


reaction-timer
localhost:8080
My Reaction Timer
Play


reaction-timer
localhost:8080
My Reaction Timer
Play
Click Me

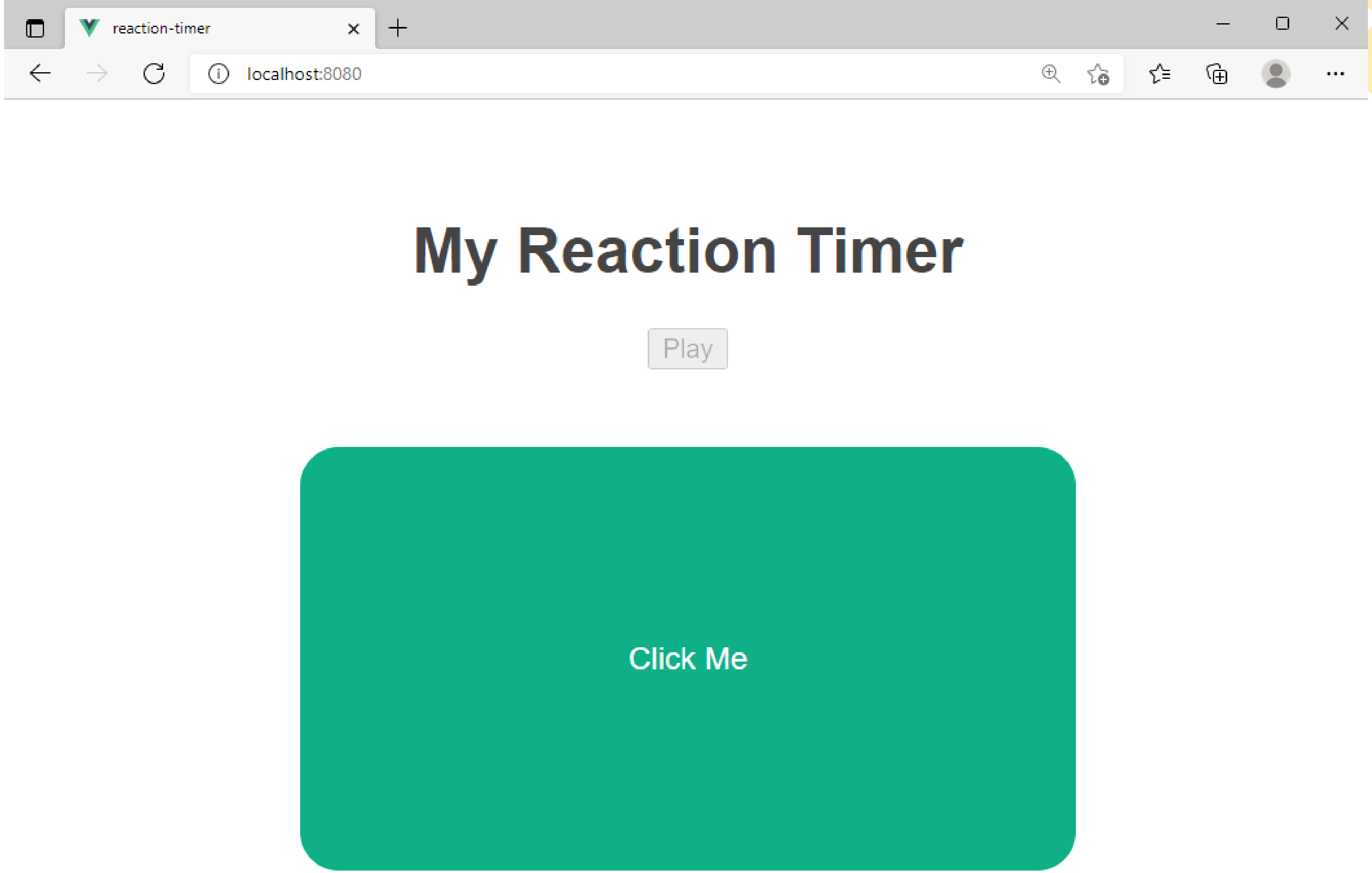
Edit file App.vue

App.vue



```
<template>
  <h1>My Reaction Timer</h1>

  <button @click="start" :disabled = "isPlaying" >Play</button>

  <Block v-if="isPlaying" :delay = "delay" />

</template>

<script>

import Block from './components/Block.vue'

export default {
  name: 'App',

  components: { Block },

  data() {
    return {
      isPlaying: false,
      delay: null,
    }
  },
  methods: {
    start() {
      this.delay = 2000 + Math.random() * 5000
      this.isPlaying = true
      console.log(this.delay)
    },
  }
}
</script>

<style>...</style>
```

Add
code



My Reaction Timer

Lifecycle Hooks

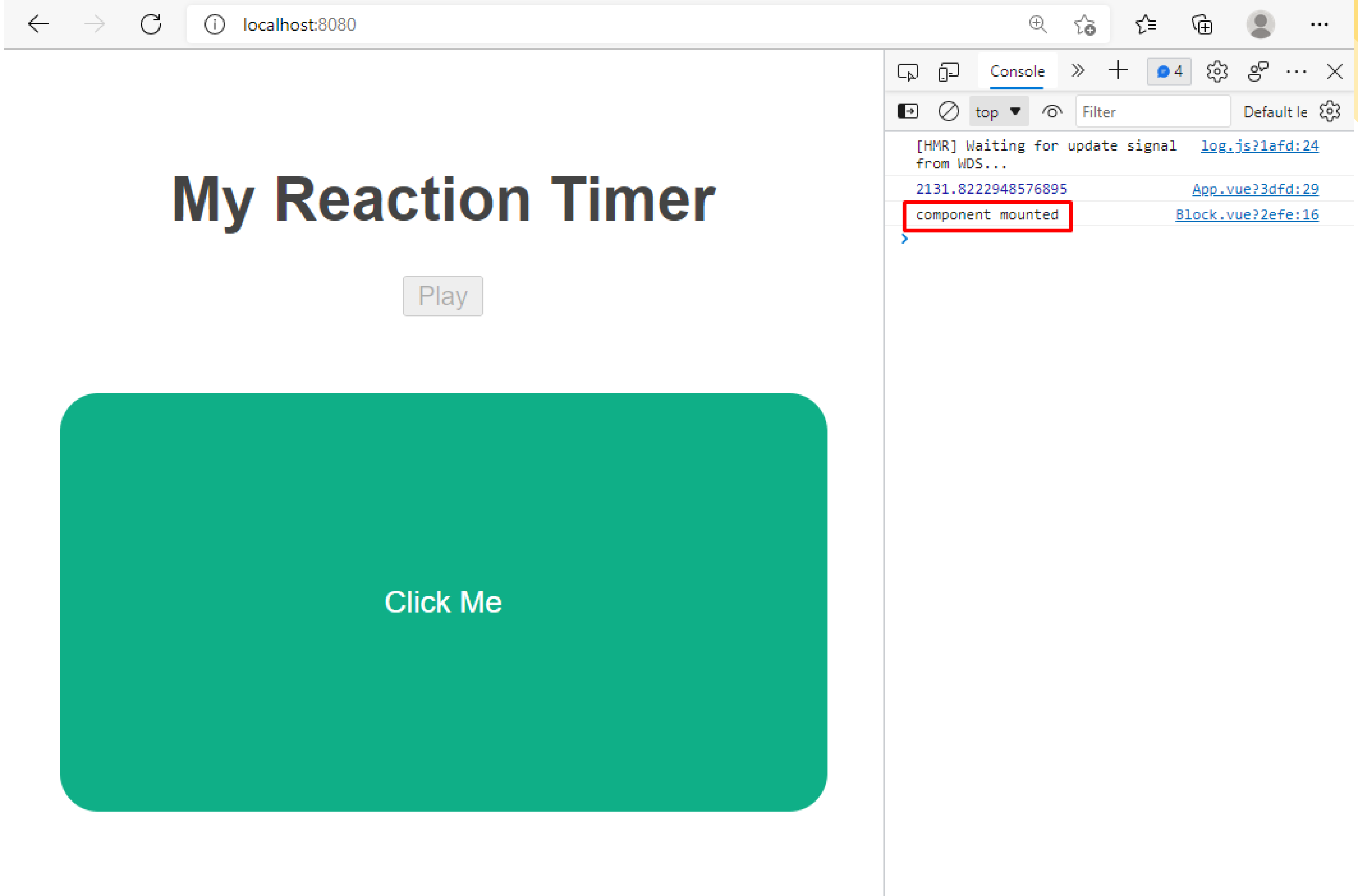
Lifecycle Diagram – more information

Edit file Block.vue

Block.vue

```
<template>
  <div class="block">
    Click Me
  </div>
</template>

<script>
export default {
  props: ['delay'],

  data(){
    return{
      showBlock: false
    }
  },
  mounted() {
    console.log('component mounted')
  },
  updated(){

  }

}
</script>

<style>
...
</style>
```

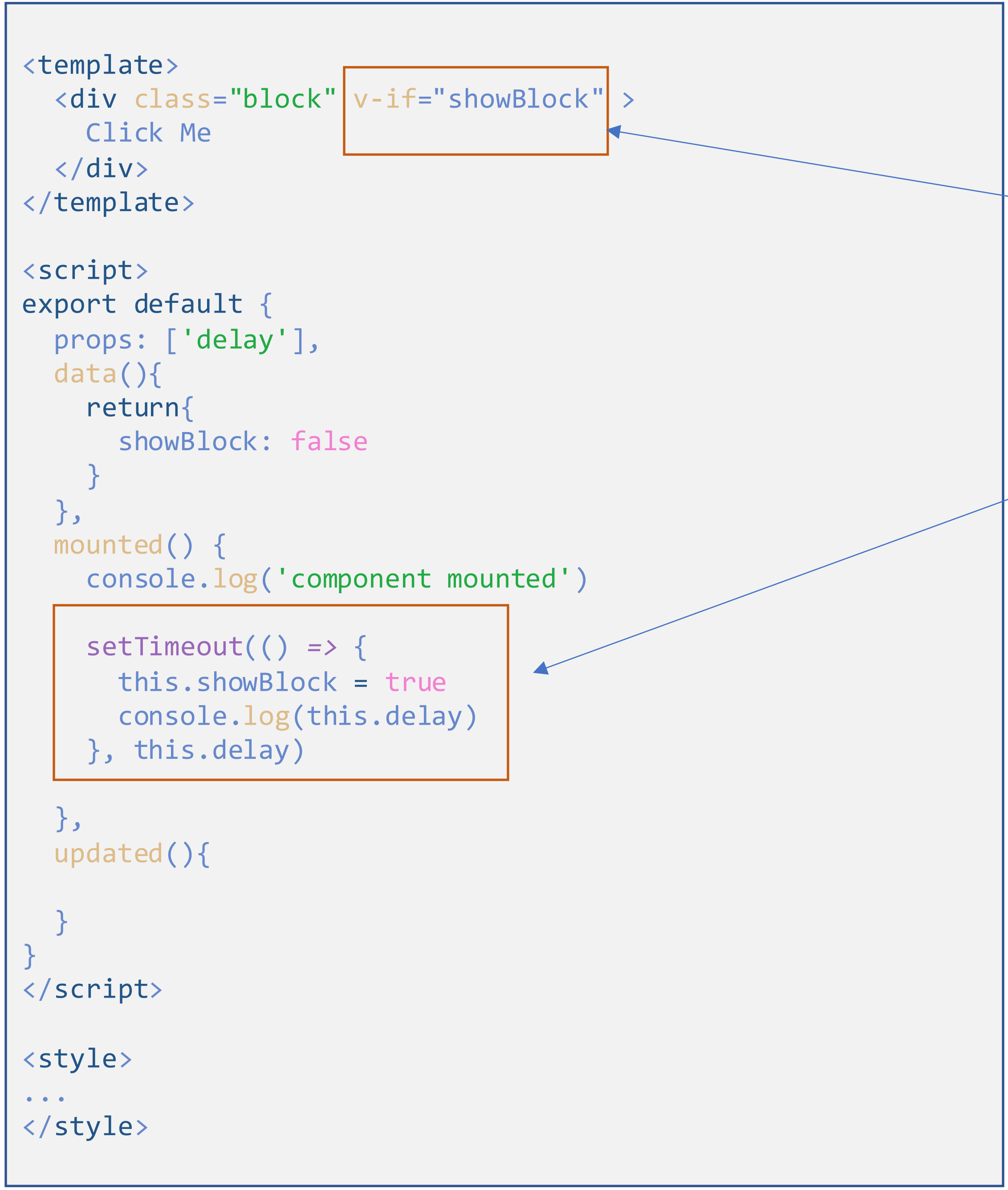
Add code



```
[HMR] Waiting for update signal    log.js?1afd:24
from WDS...
2131.8222948576895                 App.vue?3dfd:29
component mounted                Block.vue?2efe:16
>
```

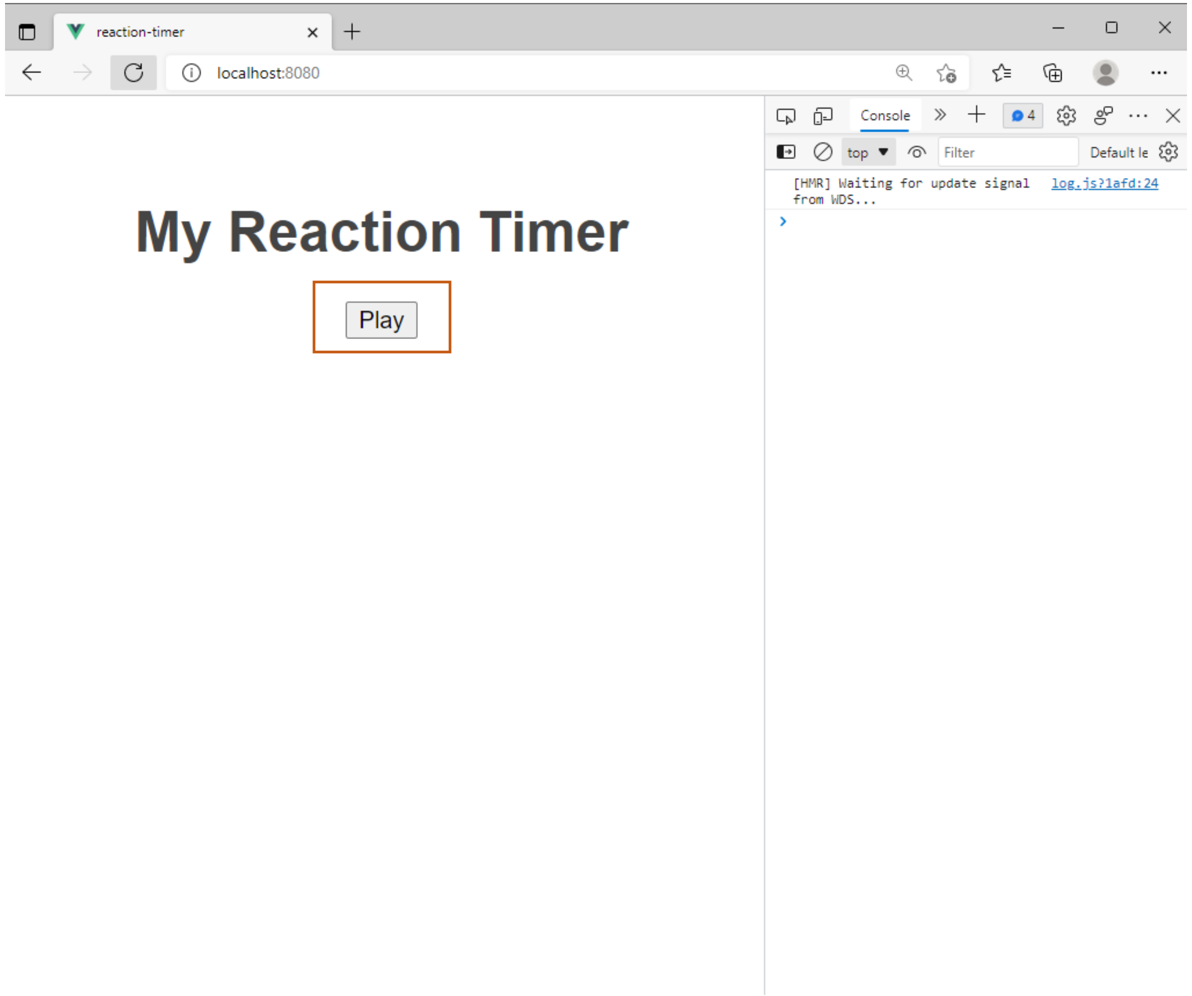
Edit file Block.vue

Block.vue



```
<template>
  <div class="block" v-if="showBlock" >
    Click Me
  </div>
</template>

<script>
export default {
  props: ['delay'],
  data(){
    return{
      showBlock: false
    }
  },
  mounted() {
    console.log('component mounted')

    setTimeout(() => {
      this.showBlock = true
      console.log(this.delay)
    }, this.delay)

  },
  updated(){

  }
}
</script>

<style>
...
</style>
```

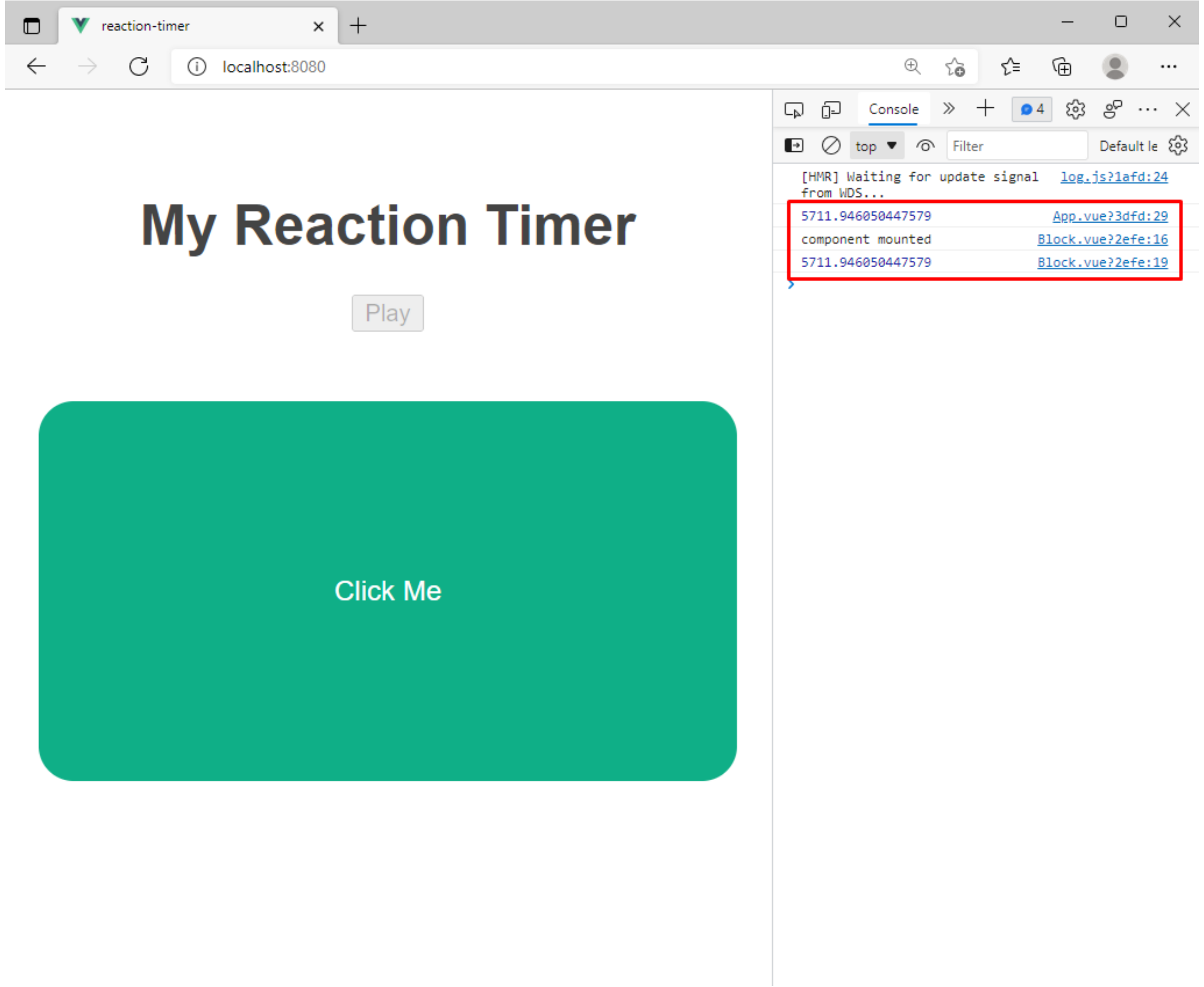
Add code



My Reaction Timer

Play

```
[HMR] Waiting for update signal    log.js?1afd:24
from WDS...
```



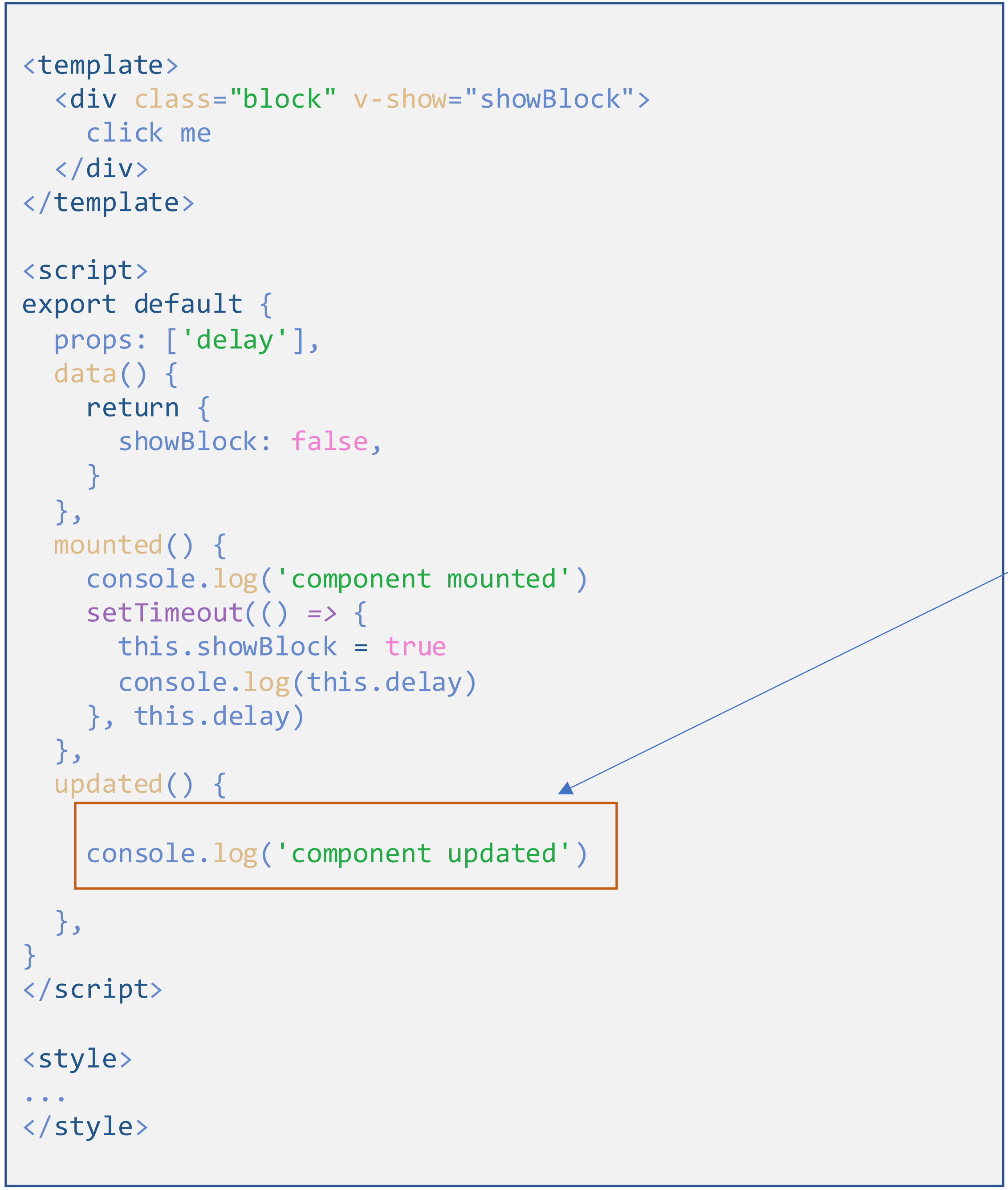
My Reaction Timer

Play

Click Me

```
[HMR] Waiting for update signal    log.js?1afd:24
from WDS...
5711.946050447579                  App.vue?3dfd:29
component mounted                Block.vue?2efe:16
5711.946050447579                Block.vue?2efe:19
```

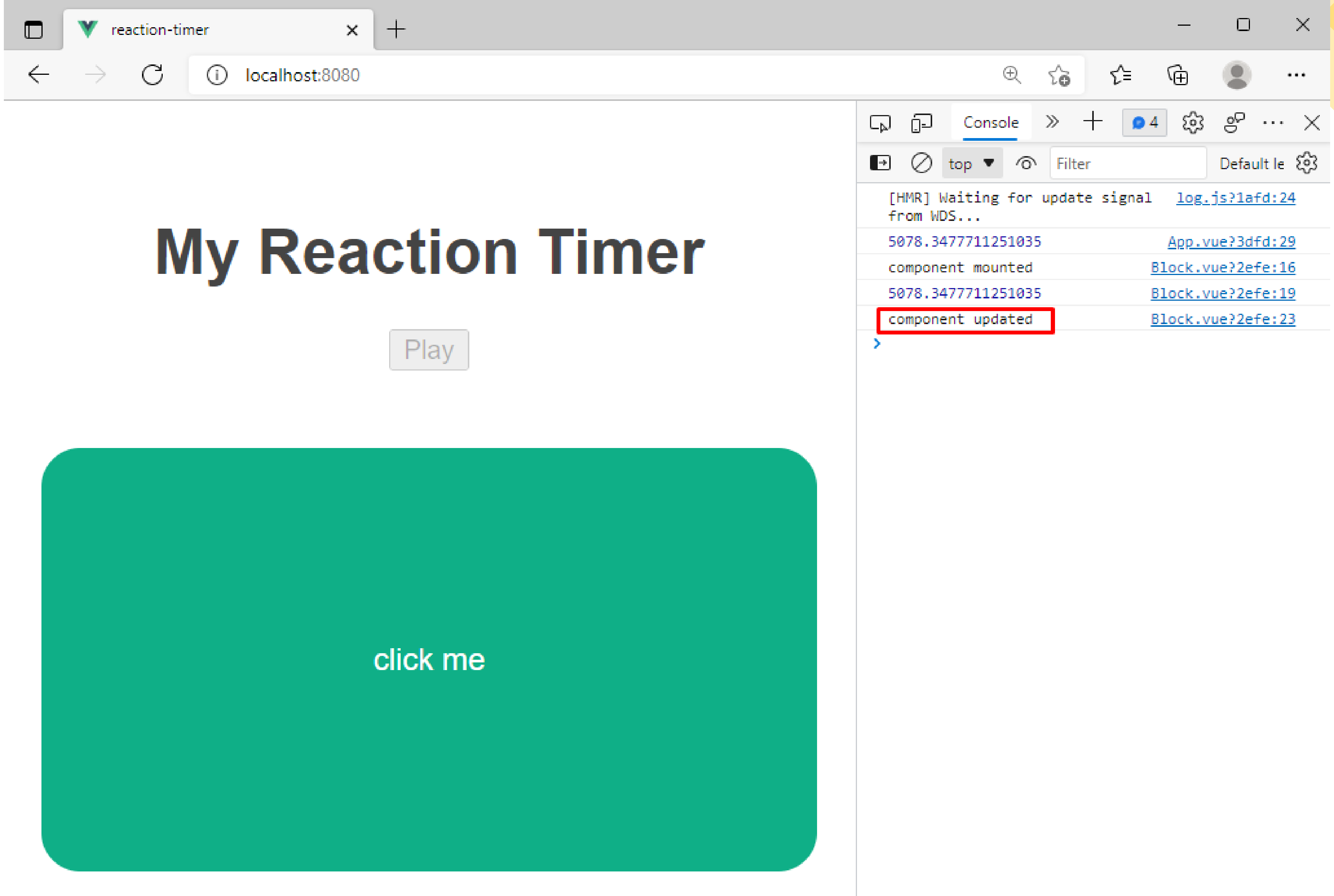
Edit file Block.vue



Block.vue

```
<template>
  <div class="block" v-show="showBlock">
    click me
  </div>
</template>

<script>
export default {
  props: ['delay'],
  data() {
    return {
      showBlock: false,
    }
  },
  mounted() {
    console.log('component mounted')
    setTimeout(() => {
      this.showBlock = true
      console.log(this.delay)
    }, this.delay)
  },
  updated() {

    console.log('component updated')

  },
}
</script>

<style>
...
</style>
```

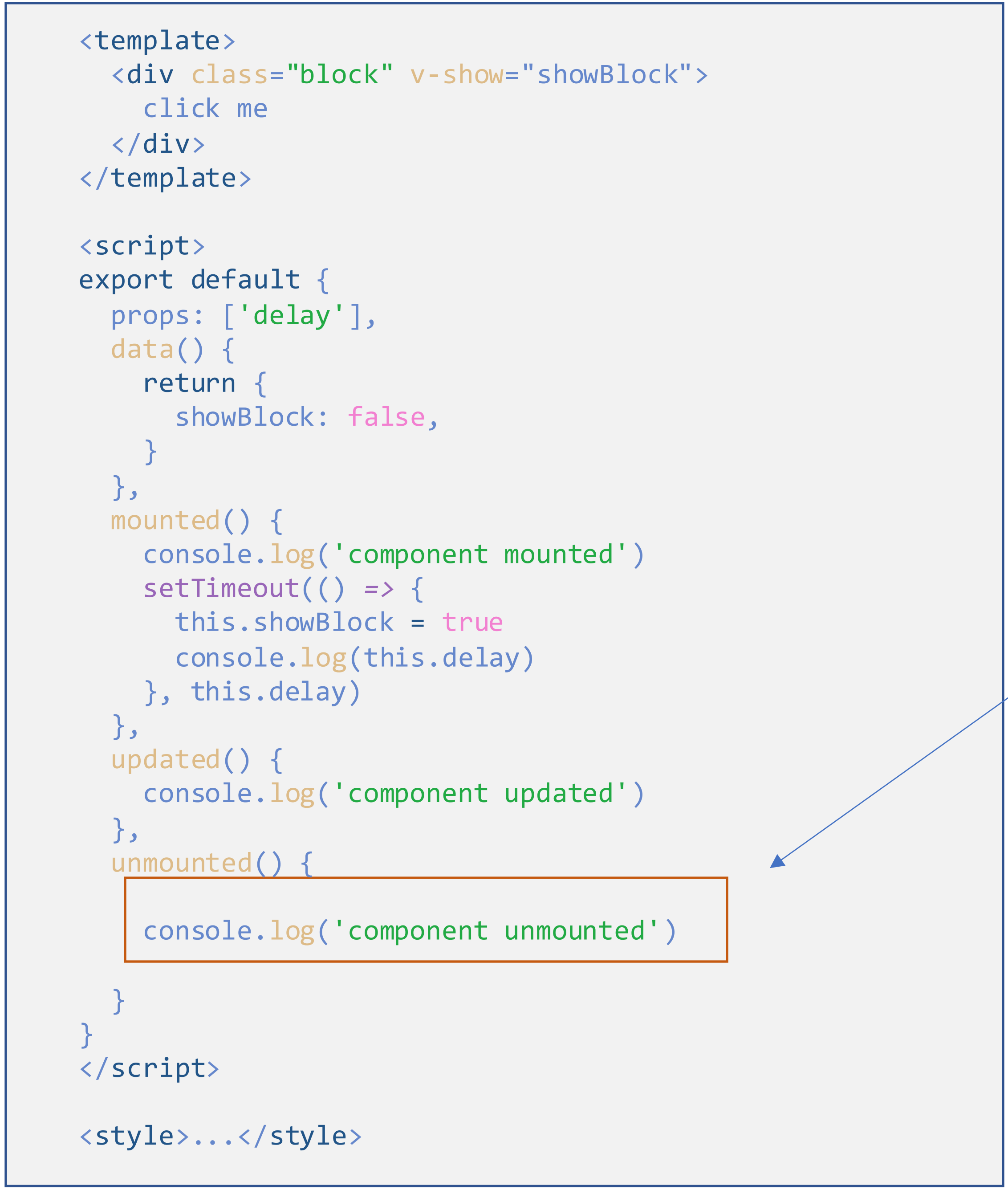
Add
code



```
[HMR] Waiting for update signal     log.js?1afd:24
from WDS...
5078.3477711251035                  App.vue?3dfd:29
component mounted                   Block.vue?2efe:16
5078.3477711251035                  Block.vue?2efe:19
component updated                   Block.vue?2efe:23
```

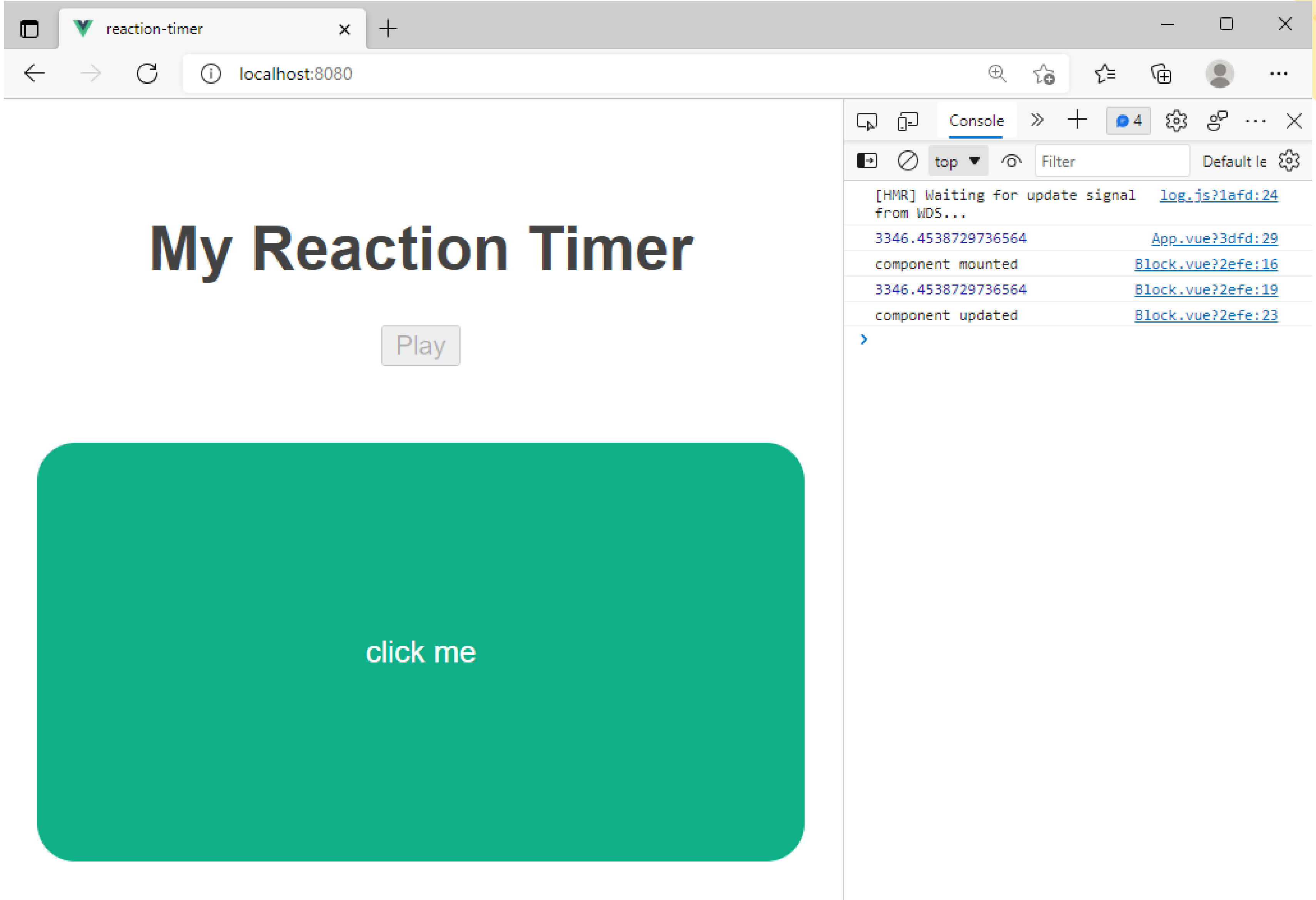
Edit file Block.vue

Block.vue

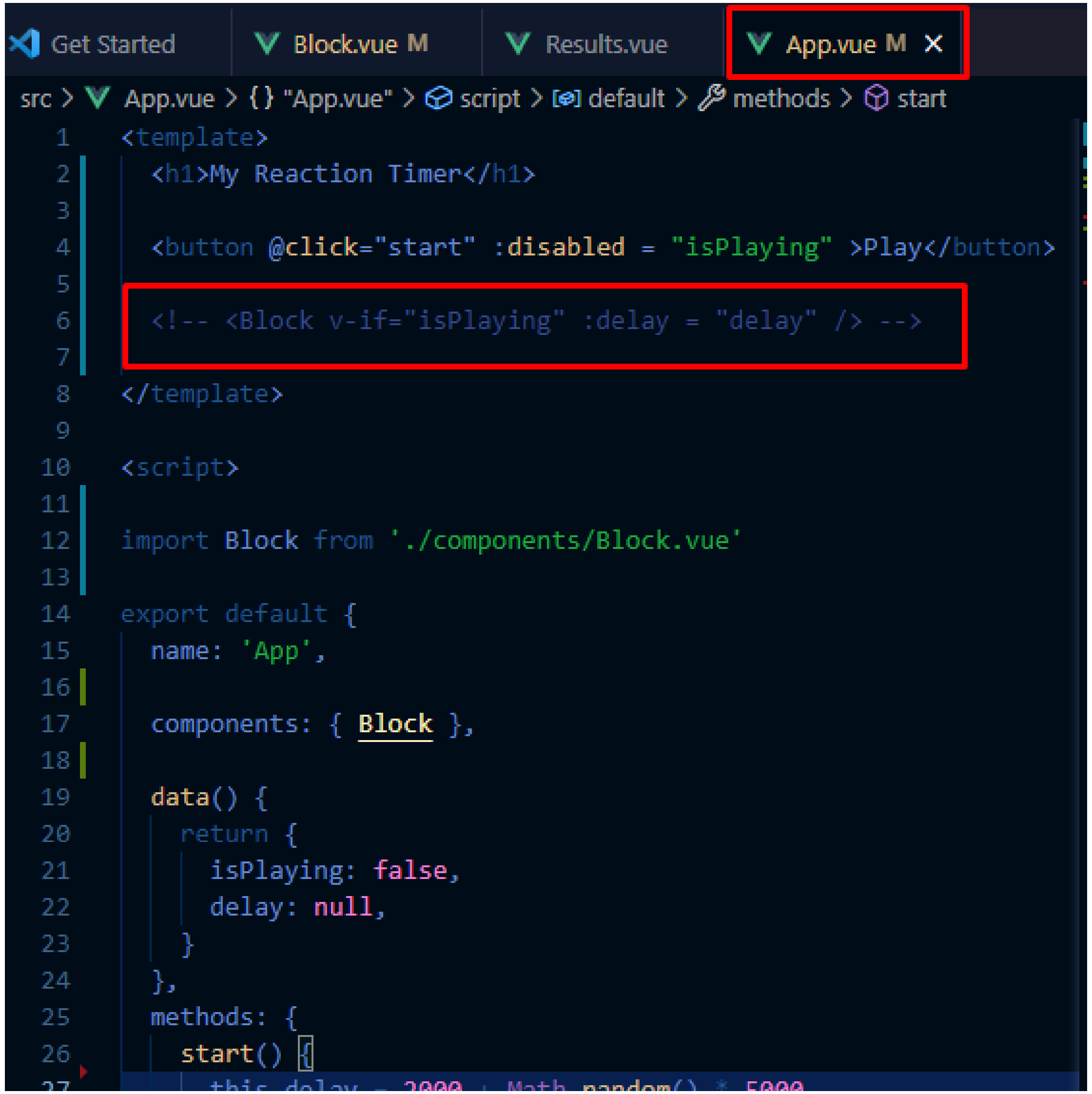


```
<template>
  <div class="block" v-show="showBlock">
    click me
  </div>
</template>

<script>
export default {
  props: ['delay'],
  data() {
    return {
      showBlock: false,
    }
  },
  mounted() {
    console.log('component mounted')
    setTimeout(() => {
      this.showBlock = true
      console.log(this.delay)
    }, this.delay)
  },
  updated() {
    console.log('component updated')
  },
  unmounted() {

    console.log('component unmounted')

  }
}
</script>

<style>...</style>
```

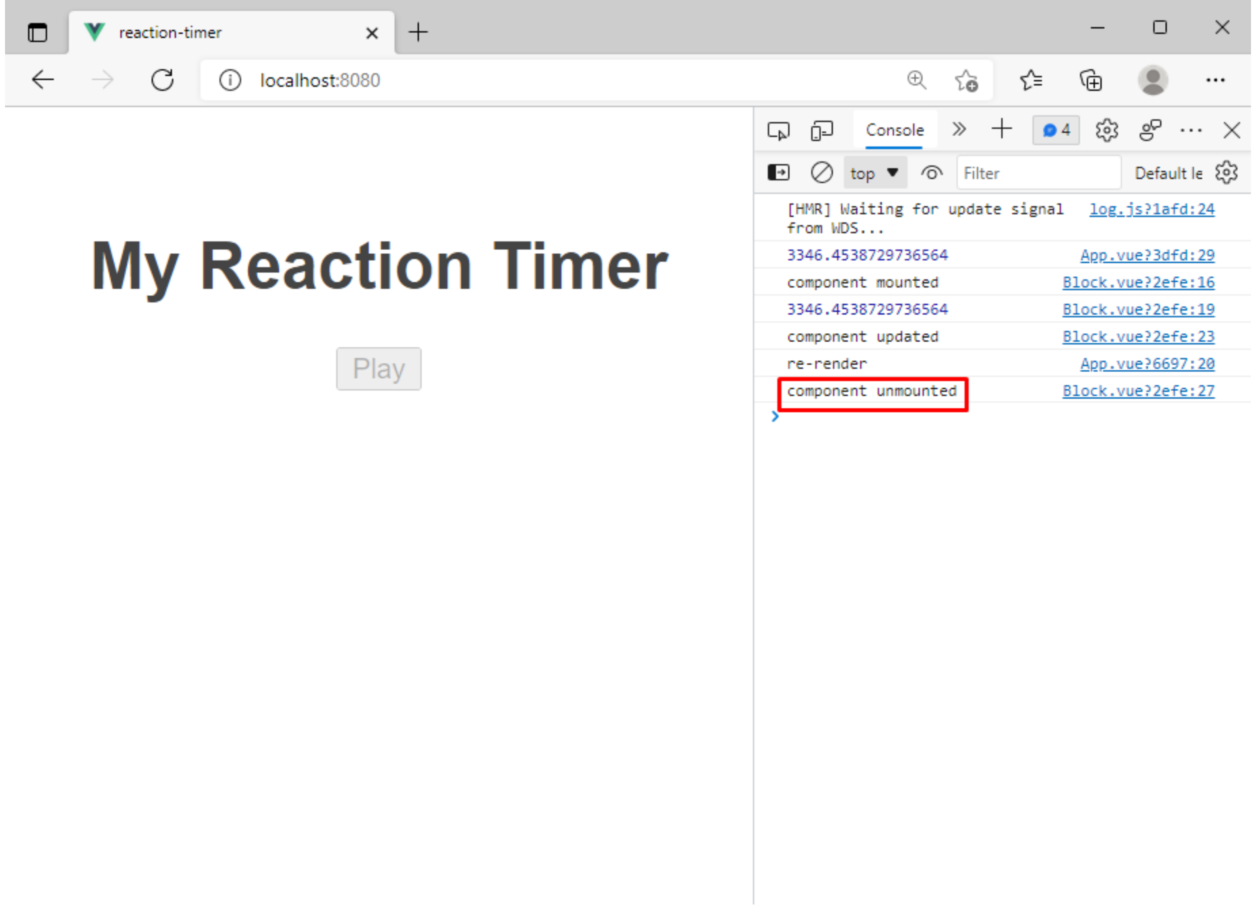
Add code



```
[HMR] Waiting for update signal   log.js?1afd:24
from WDS...
3346.4538729736564                App.vue?3dfd:29
component mounted                 Block.vue?2efe:16
3346.4538729736564                Block.vue?2efe:19
component updated                 Block.vue?2efe:23
```

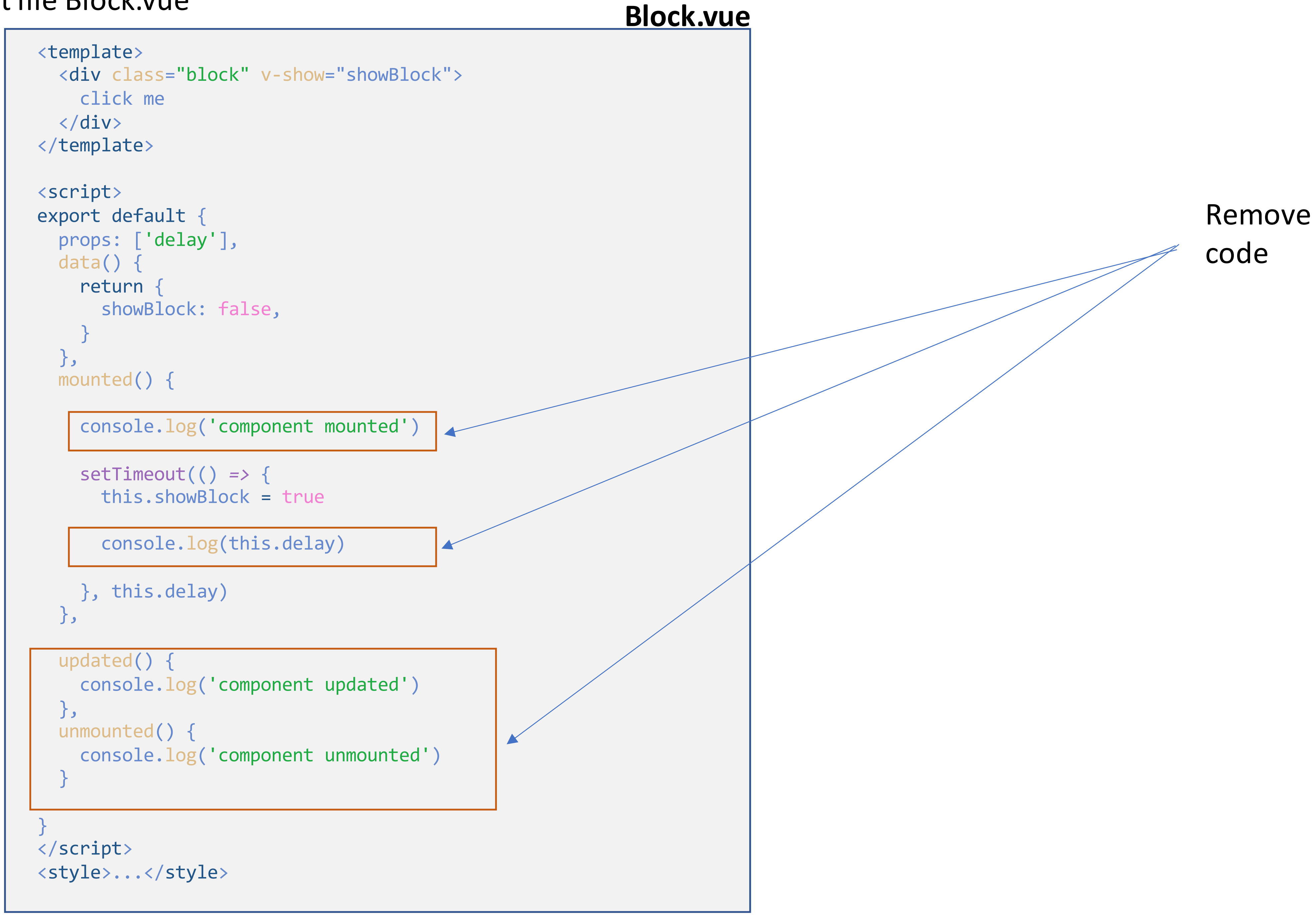

Get Started
Block.vue M
Results.vue
App.vue M
src > App.vue > {} "App.vue" > script > default > methods > start

```
<template>
  <h1>My Reaction Timer</h1>

  <button @click="start" :disabled = "isPlaying" >Play</button>

  <!-- <Block v-if="isPlaying" :delay = "delay" /> -->

</template>

<script>

import Block from './components/Block.vue'

export default {
  name: 'App',

  components: { Block },

  data() {
    return {
      isPlaying: false,
      delay: null,
    }
  },
  methods: {
    start() {
      this.delay = 2000 + Math.random() * 5000
```



```
[HMR] Waiting for update signal    log.js?1afd:24
from WDS...
3346.4538729736564                App.vue?3dfd:29
component mounted               Block.vue?2efe:16
3346.4538729736564              Block.vue?2efe:19
component updated               Block.vue?2efe:23
re-render                         App.vue?6697:20
component unmounted             Block.vue?2efe:27
```

Edit file Block.vue



Block.vue

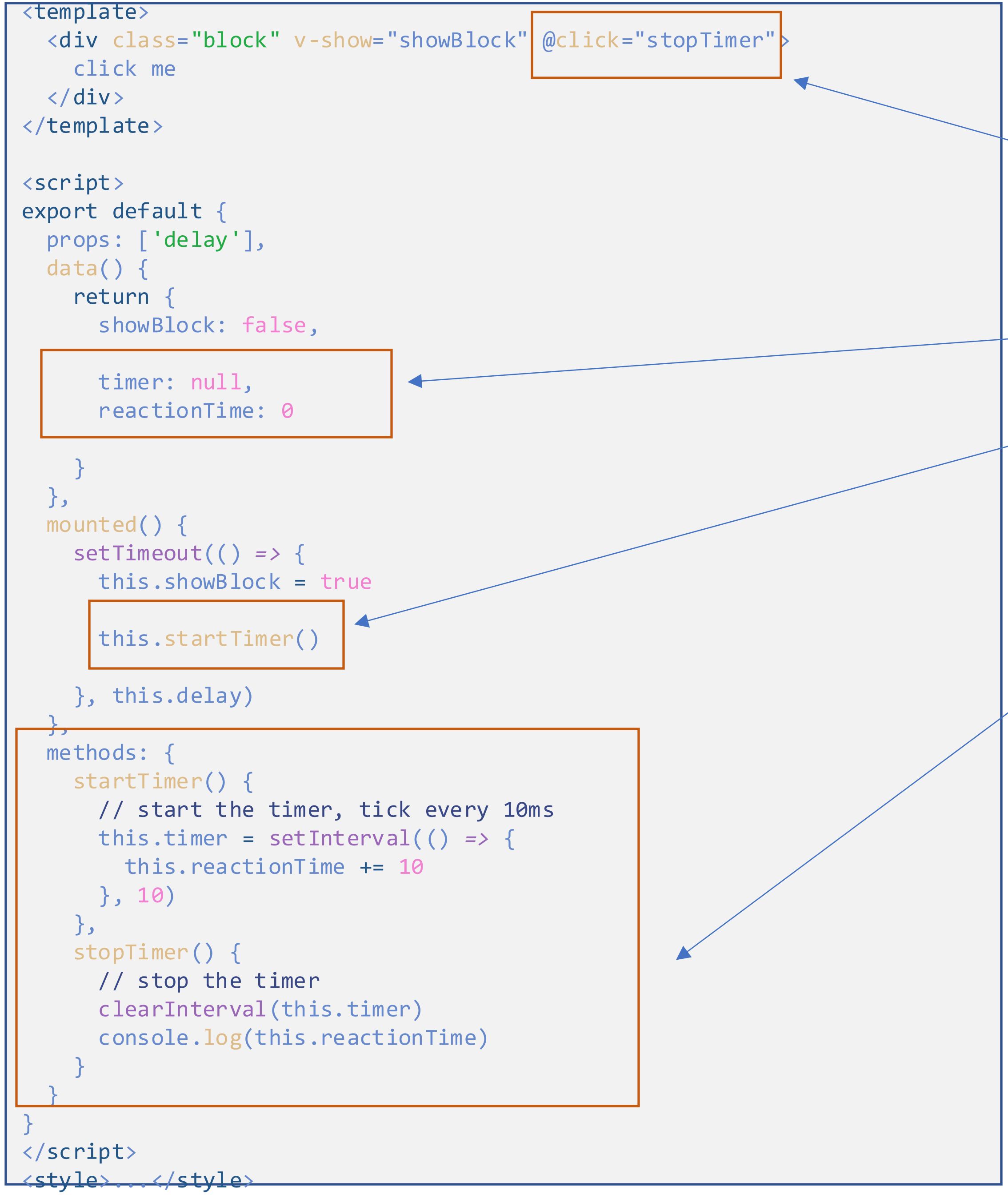
```
<template>
  <div class="block" v-show="showBlock">
    click me
  </div>
</template>

<script>
export default {
  props: ['delay'],
  data() {
    return {
      showBlock: false,
    }
  },
  mounted() {

    console.log('component mounted')

    setTimeout(() => {
      this.showBlock = true

      console.log(this.delay)

    }, this.delay)
  },

  updated() {
    console.log('component updated')
  },
  unmounted() {
    console.log('component unmounted')
  }

}
</script>
<style>...</style>
```

Remove code

Creating a Timer

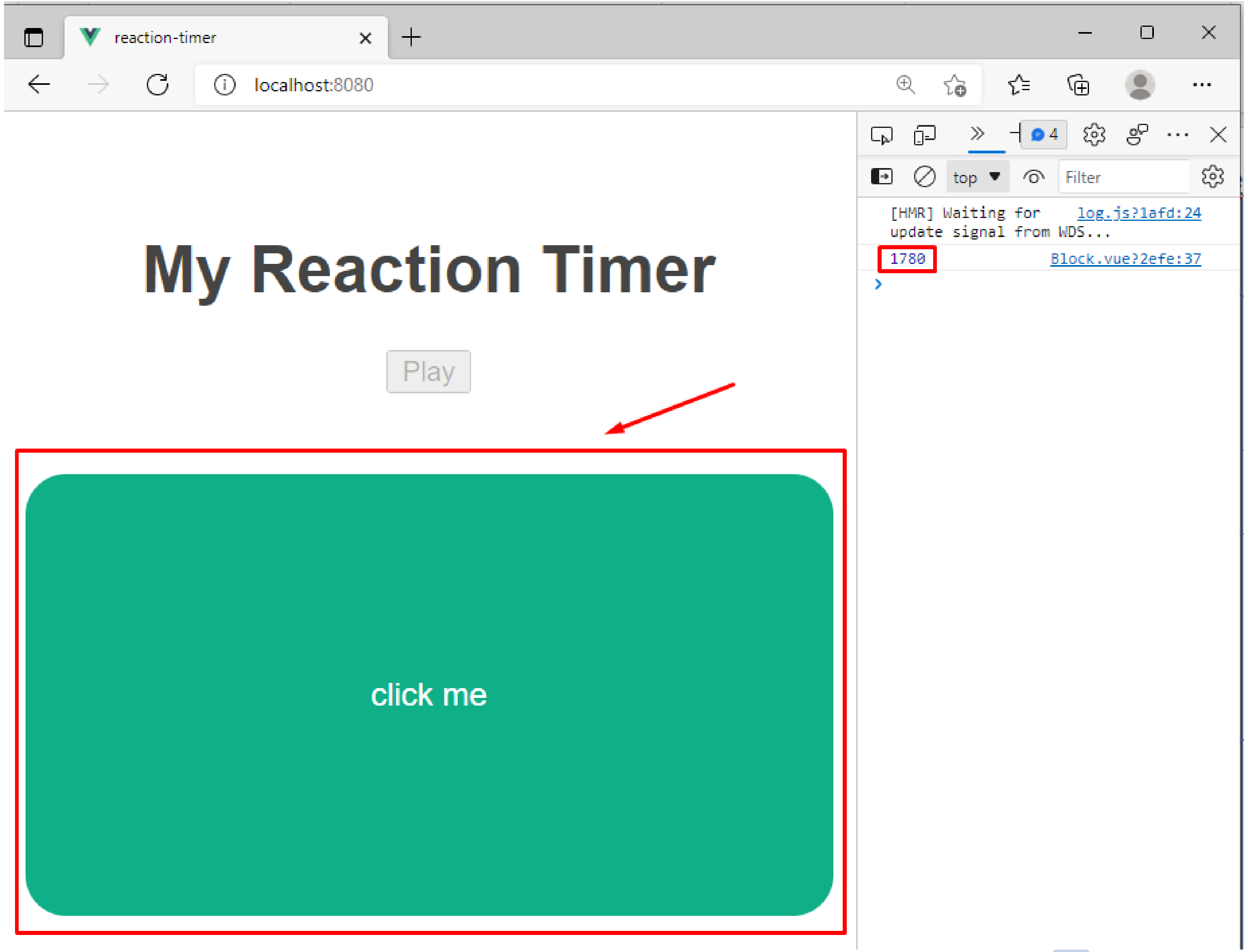
Edit file Block.vue

Block.vue



```
<template>
  <div class="block" v-show="showBlock" @click="stopTimer">
    click me
  </div>
</template>

<script>
export default {
  props: ['delay'],
  data() {
    return {
      showBlock: false,

      timer: null,
      reactionTime: 0

    }
  },
  mounted() {
    setTimeout(() => {
      this.showBlock = true

      this.startTimer()

    }, this.delay)
  },
  methods: {
    startTimer() {
      // start the timer, tick every 10ms
      this.timer = setInterval(() => {
        this.reactionTime += 10
      }, 10)
    },
    stopTimer() {
      // stop the timer
      clearInterval(this.timer)
      console.log(this.reactionTime)
    }
  }
}
</script>
<style>...</style>
```

Add code

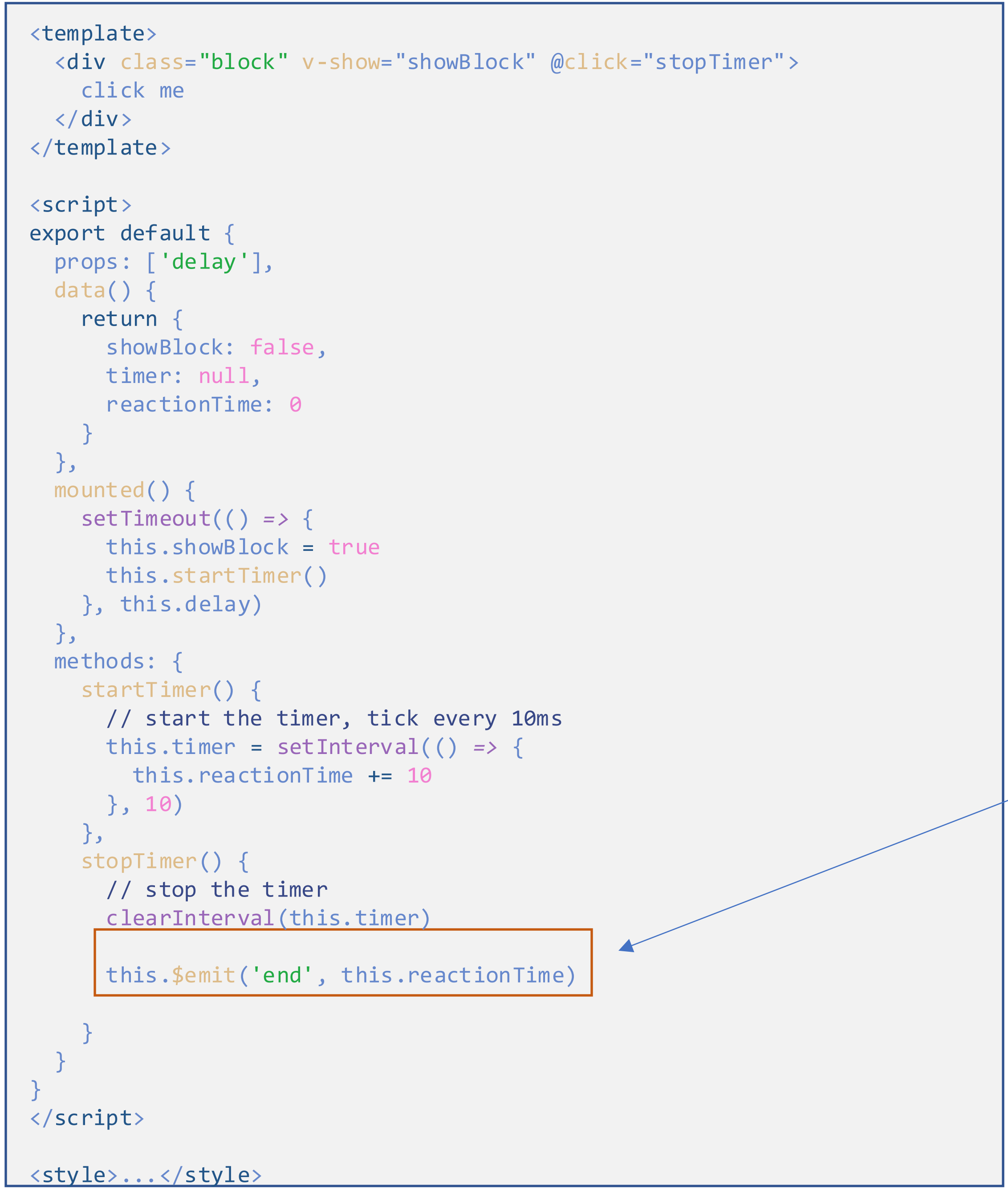


```
[HMR] Waiting for      log.js?1afd:24
update signal from WDS...
1780               Block.vue?2efe:37
```

Custom Events with Data

Edit file Block.vue

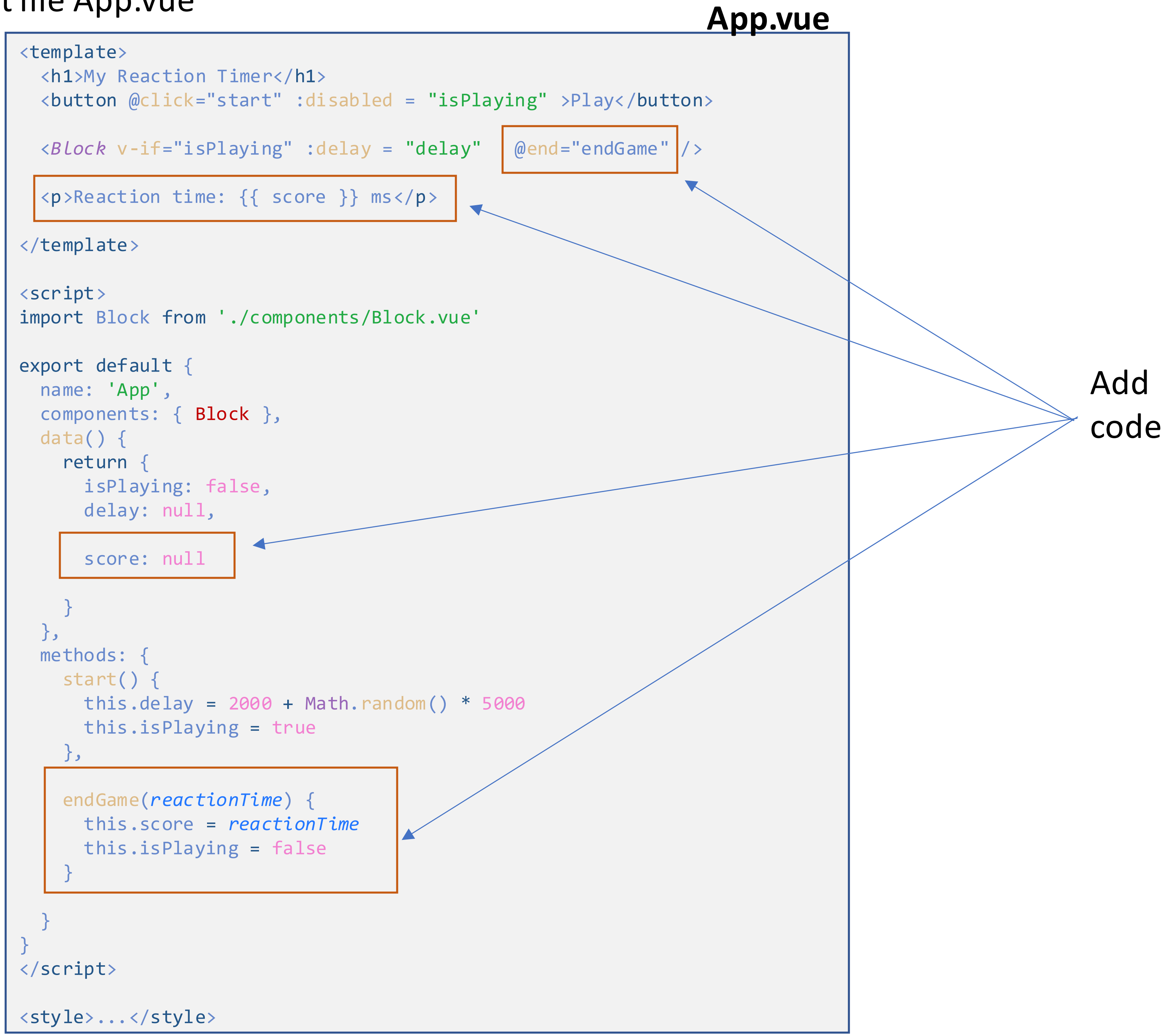
Block.vue



```
<template>
  <div class="block" v-show="showBlock" @click="stopTimer">
    click me
  </div>
</template>

<script>
export default {
  props: ['delay'],
  data() {
    return {
      showBlock: false,
      timer: null,
      reactionTime: 0
    }
  },
  mounted() {
    setTimeout(() => {
      this.showBlock = true
      this.startTimer()
    }, this.delay)
  },
  methods: {
    startTimer() {
      // start the timer, tick every 10ms
      this.timer = setInterval(() => {
        this.reactionTime += 10
      }, 10)
    },
    stopTimer() {
      // stop the timer
      clearInterval(this.timer)

      this.$emit('end', this.reactionTime)

    }
  }
}
</script>

<style>...</style>
```

Add code

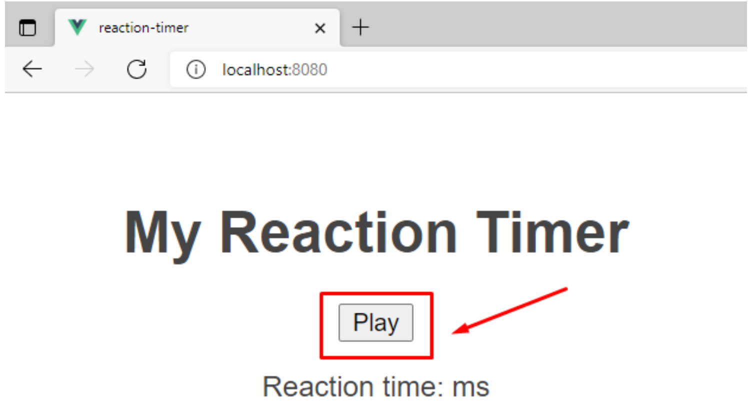
Edit file App.vue



App.vue

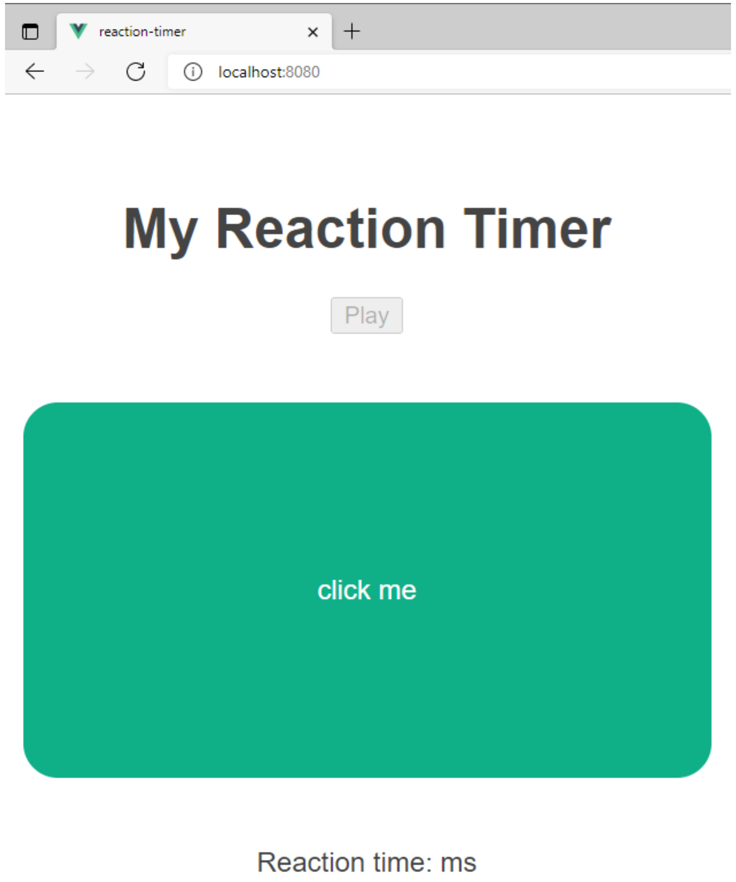
```
<template>
  <h1>My Reaction Timer</h1>
  <button @click="start" :disabled = "isPlaying" >Play</button>

  <Block v-if="isPlaying" :delay = "delay"  @end="endGame" />

  <p>Reaction time: {{ score }} ms</p>

</template>

<script>
import Block from './components/Block.vue'

export default {
  name: 'App',
  components: { Block },
  data() {
    return {
      isPlaying: false,
      delay: null,

      score: null

    }
  },
  methods: {
    start() {
      this.delay = 2000 + Math.random() * 5000
      this.isPlaying = true
    },

    endGame(reactionTime) {
      this.score = reactionTime
      this.isPlaying = false
    }

  }
}
</script>

<style>...</style>
```

Add code



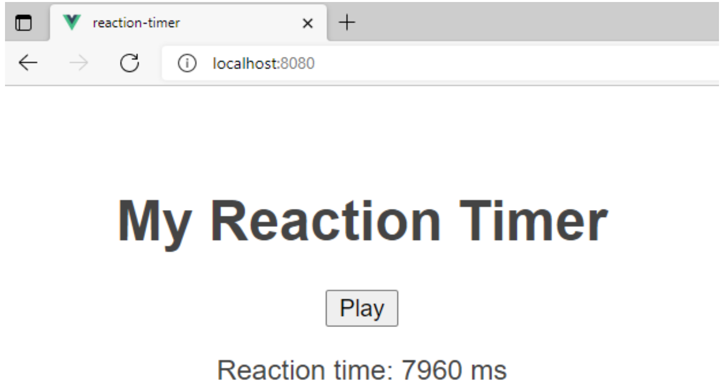
My Reaction Timer

Reaction time: ms



My Reaction Timer

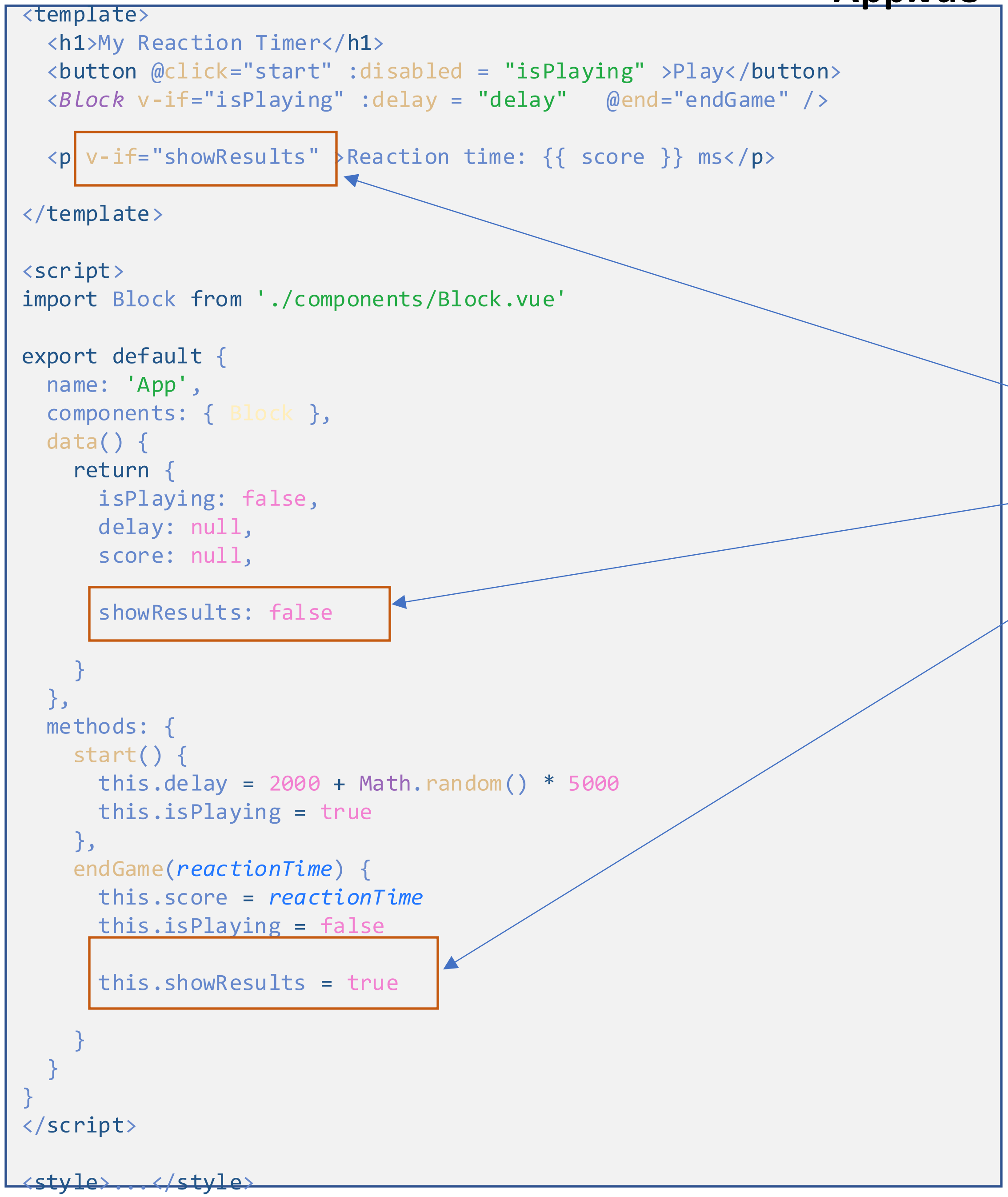
Reaction time: ms



My Reaction Timer

Reaction time: 7960 ms

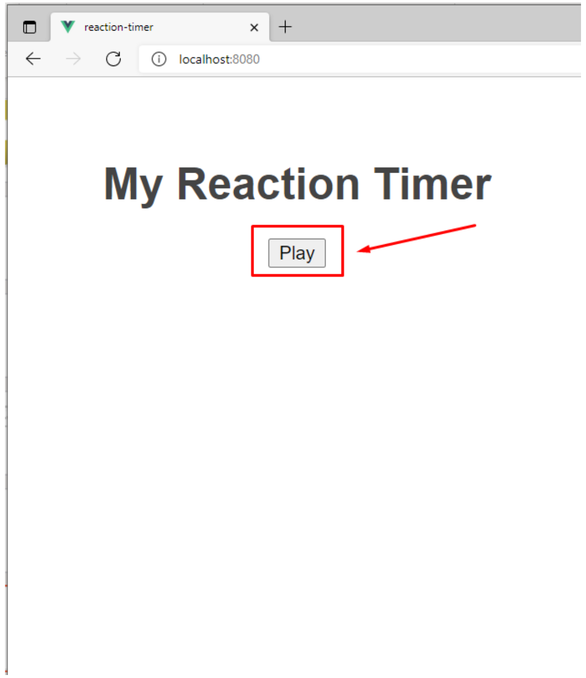
Edit file App.vue



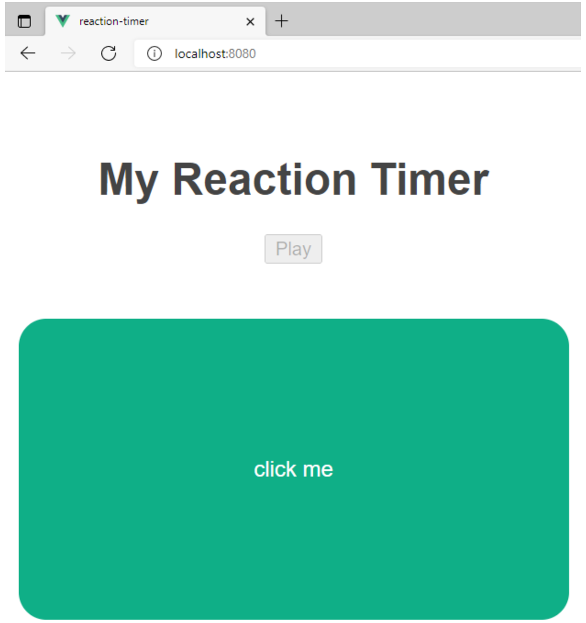
App.vue

```
<template>
  <h1>My Reaction Timer</h1>
  <button @click="start" :disabled = "isPlaying" >Play</button>
  <Block v-if="isPlaying" :delay = "delay"   @end="endGame" />

  <p v-if="showResults" >Reaction time: {{ score }} ms</p>

</template>

<script>
import Block from './components/Block.vue'

export default {
  name: 'App',
  components: { Block },
  data() {
    return {
      isPlaying: false,
      delay: null,
      score: null,

      showResults: false

    }
  },
  methods: {
    start() {
      this.delay = 2000 + Math.random() * 5000
      this.isPlaying = true
    },
    endGame(reactionTime) {
      this.score = reactionTime
      this.isPlaying = false

      this.showResults = true

    }
  }
}
</script>

<style>...</style>
```

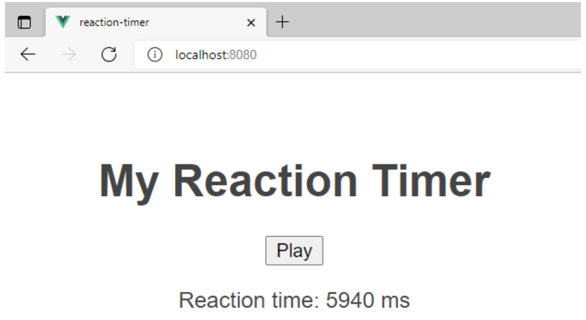
Add
code



My Reaction Timer



My Reaction Timer



My Reaction Timer

Reaction time: 5940 ms

Challenge

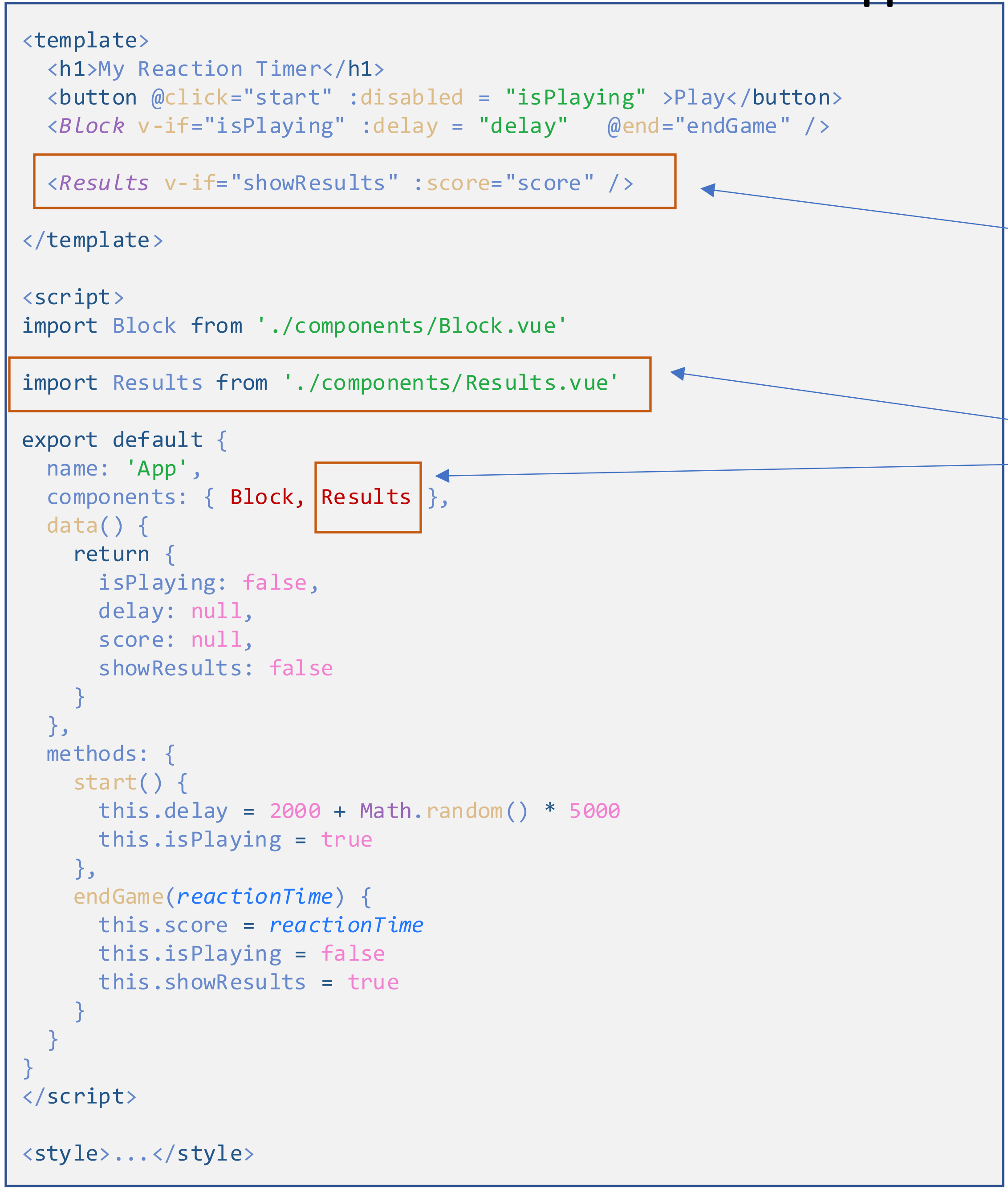
Showing a Results Component

Challenge

- when the game ends, show the results component
- output the score inside the results component

Edit file App.vue

App.vue



```
<template>
  <h1>My Reaction Timer</h1>
  <button @click="start" :disabled = "isPlaying" >Play</button>
  <Block v-if="isPlaying" :delay = "delay"   @end="endGame" />

  <Results v-if="showResults" :score="score" />

</template>

<script>
import Block from './components/Block.vue'

import Results from './components/Results.vue'

export default {
  name: 'App',
  components: { Block, Results },
  data() {
    return {
      isPlaying: false,
      delay: null,
      score: null,
      showResults: false
    }
  },
  methods: {
    start() {
      this.delay = 2000 + Math.random() * 5000
      this.isPlaying = true
    },
    endGame(reactionTime) {
      this.score = reactionTime
      this.isPlaying = false
      this.showResults = true
    }
  }
}
</script>

<style>...</style>
```

Modify code

Add code

Edit file Results.vue

Results.vue

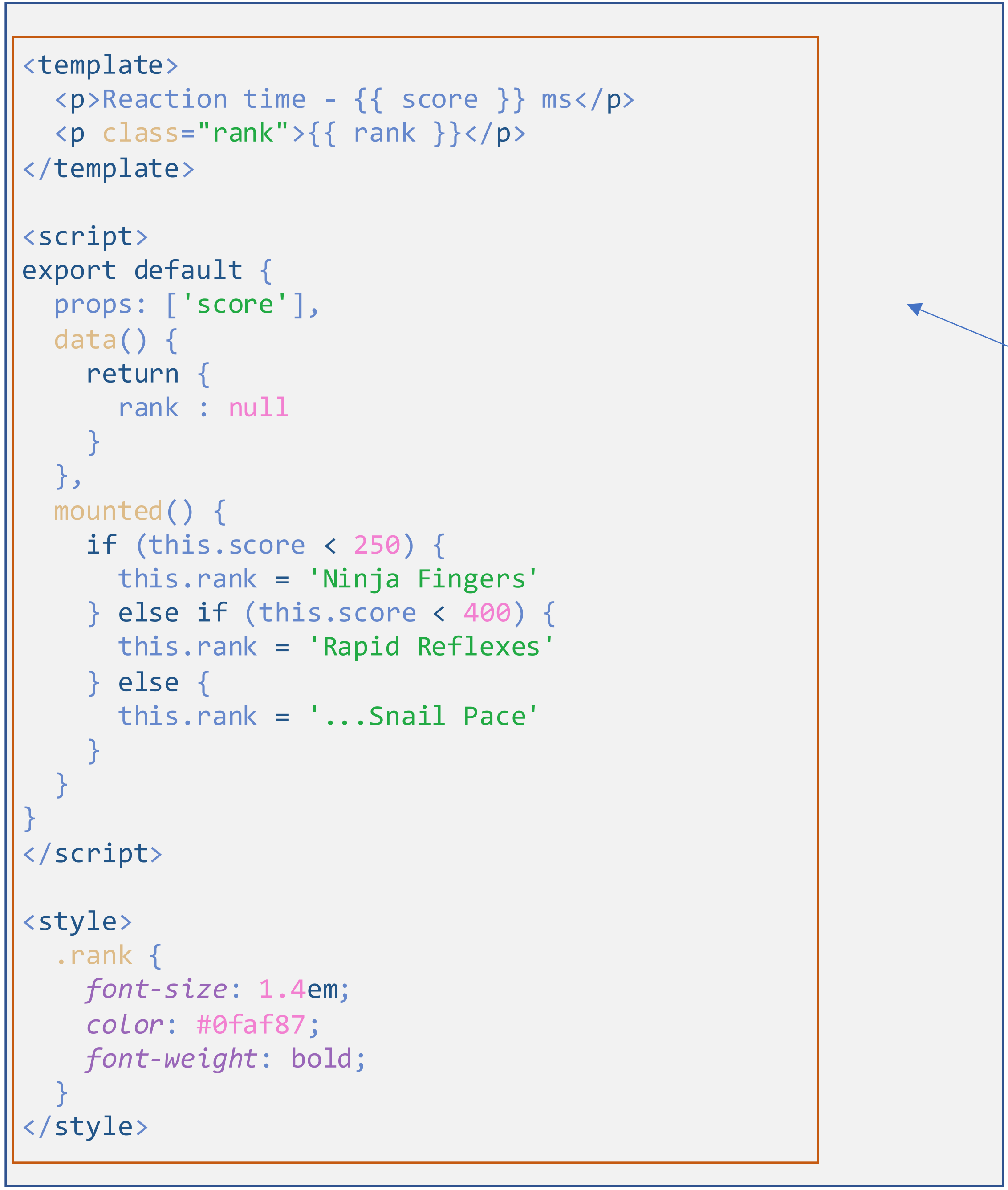
```
<template>
  <p>Reaction time - {{ score }} ms</p>
</template>

<script>
export default {
  props: ['score']
}
</script>
```

Add code

Finishing Touches

Edit file Results.vue

Results.vue



```
<template>
  <p>Reaction time - {{ score }} ms</p>
  <p class="rank">{{ rank }}</p>
</template>

<script>
export default {
  props: ['score'],
  data() {
    return {
      rank : null
    }
  },
  mounted() {
    if (this.score < 250) {
      this.rank = 'Ninja Fingers'
    } else if (this.score < 400) {
      this.rank = 'Rapid Reflexes'
    } else {
      this.rank = '...Snail Pace'
    }
  }
}
</script>

<style>
  .rank {
    font-size: 1.4em;
    color: #0faf87;
    font-weight: bold;
  }
</style>
```

Modify code