



07

Validando propriedades

Transcrição

Terminamos mais um componente. Ponto para nós. No entanto, vimos que se o programador esquecer de usar o dois pontos na propriedade `confirmacao` do nosso `Botao` lá em `Home` o valor será passado como string fazendo com que a confirmação não funcione direito. A boa notícia é que podemos indicar o tipo aceito por uma `prop` dentro de um componente, inclusive se o seu valor é obrigatório. Nossa aplicação continua dando erro, mas a mensagem no console deixa claro que determinada regra foi ferida.

Vamos alterar `Botao`. Nela, todas as propriedades são tipo `String`, exceto `confirmacao` que é boolean. Além disso, `confirmacao` e `estilos` não são propriedades obrigatórias:

```
<!-- alurapic/src/components/shared/botao/Bota.vue -->
<template>
  <button :class="estiloDoBotao" :type="tipo"
estilo="perigo" @click="disparaAcao()">{{rotulo}}</button>
</template>
<script>
export default {

  props: {
    tipo: {
      type: String,
      required: true
```

```
    },

    rotulo: {
      type: String,
      required: true
    },

    confirmacao: Boolean,
    estilo: String
  },
  methods: {

    /* código omitido */
  },

  computed: {

    /* código omitido */
  }
}
</script>

<style scoped>
  /* código omitido */

</style>
```

[COPIAR CÓDIGO](#)

Agora, se voltarmos para `Home` e removermos `:` da propriedade `confirmacao` será exibida a seguinte mensagem no console do Chrome:

```
vue.common.js?e881:519[Vue warn]: Invalid prop: type check
failed for prop "confirmacao". Expected Boolean, got String.
(found in component <meu-botao> at Botao.vue)
```

[COPIAR CÓDIGO](#)

Muito mais fácil de detectar o problema, não? Só não se esqueça de voltar com os dois pontos lá na propriedade que modificamos por último.