



Lifecycle Hooks: em que momento acessaremos nossa API?

Transcrição

Vamos voltar para `alurapic/src/App.vue`. Com nosso módulo `VueResource` baixado e registrado já temos condições de realizar requisições para nosso API, mas quando faremos isso? Quando o usuário clicar em um botão? Com certeza não.

Queremos que a requisição seja feita assim que nosso componente `App` tiver sido criado. Para nossa alegria, todo componente do Vue possui ganchos em seu ciclo de vida (Lifecycle Hooks) que permite executar um código em determinada fase do seu ciclo. No caso, a fase que nos interessa é a `created`. Vamos aproveitar e remover a lista de fotos:

```
<template>
```

```
  <div>
```

```
    <h1>{{ titulo }}</h1>
```

```
    
```

```
  </div>
```

```
</template>
```

```
<script>
```

```
export default {  
  
  data () {  
    return {  
      titulo: "Alurapic",  
      fotos: [  
        {  
          url: 'https://encrypted-tbn2.gstatic.com/images?  
q=tbn:ANd9GcTwV4kVzT5McBdGSgqlVeRzubrNH_mOrrkKseDOGFURq20Hmsre1Eh  
  
          titulo: 'cachorro'  
        },  
        {  
          url: 'https://encrypted-tbn2.gstatic.com/images?  
q=tbn:ANd9GcTwV4kVzT5McBdGSgqlVeRzubrNH_mOrrkKseDOGFURq20Hmsre1Eh  
  
          titulo: 'cachorro2'  
        }  
      ]  
    }  
  },  
  created() {  
  
    // buscaremos nossas fotos aqui  
  }  
}  
</script>  
<style>  
</style>
```

[COPIAR CÓDIGO](#)

Você pode acessar a própria documentação do Vue para ver todos os ganchos suportados no ciclo de vida de um componente neste endereço

<https://vuejs.org/v2/api/#Options-Lifecycle-Hooks> .

Aliás, não faz mais nenhum sentido termos fixa a lista de fotos. Vamos ter apenas um array vazio que receberá os dados vindos da nossa API:

```
<template>

  <div>
    <h1>{{ titulo }}</h1>

    

  </div>

</template>

<script>

export default {

  data () {
    return {
      titulo: "Alurapic",
      fotos: []

    }
  },
  created() {

    // buscaremos nossas fotos aqui
  }
}

</script>
```

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

[COPIAR CÓDIGO](#)

Para termos certeza que o código em `create` é chamado assim que nosso componente é criado, vamos colocar temporariamente um alerta. Em seguida, vamos subir novamente nossa aplicação através do CLI para que seja possível acessá-la no navegador:

```
<template>  
  
  <div>  
    <h1>{{ titulo }}</h1>  
  
      
  
  </div>  
  
</template>  
  
<script>  
  
export default {  
  
  data () {  
    return {  
      titulo: "Alurapic",  
      fotos: []  
    }  
  },  
  created() {
```

```
    alert('Funcionou!');  
  }  
}  
</script>  
<style>  
</style>
```

[COPIAR CÓDIGO](#)

Excelente, o alerta é exibido. Chegou a hora de substituí-lo pelo código que busca nossas fotos da nossa API.