

# Documentação do arquivo viewVehicles.js

## Introdução

Este arquivo contém a implementação de um componente Lightning Web Component (LWC) chamado ViewVehicles. Ele é responsável por exibir uma lista de veículos recuperados de um Apex Controller no Salesforce e permitir que o usuário realize a compra de um veículo.

## Descrição

O componente ViewVehicles utiliza a funcionalidade de @wire para buscar dados de veículos a partir de um método Apex chamado getVehicles. Além disso, ele implementa uma funcionalidade para persistir a compra de um veículo utilizando o método Apex persistPurchase. O componente também exibe notificações (toasts) para informar o usuário sobre o sucesso ou falha das operações realizadas.

## Estrutura

O arquivo é estruturado da seguinte forma:

1. Importação de módulos e dependências.
2. Declaração da classe ViewVehicles que estende LightningElement.
3. Uso do decorador @wire para buscar dados de veículos.
4. Implementação de métodos para manipular eventos e exibir notificações.

## Dependências

O arquivo depende dos seguintes módulos e métodos:

- LightningElement e wire do módulo lwc.
- Métodos Apex getVehicles e persistPurchase do Apex Controller Vehicles.
- ShowToastEvent do módulo lightning/platformShowToastEvent.

## Imports

```
import { LightningElement, wire } from 'lwc';
import getVehicles from '@salesforce/apex/Vehicles.getVehicles';
import persistPurchase from '@salesforce/apex/Vehicles.persistPurchase';
import { ShowToastEvent } from "lightning/platformShowToastEvent";
```

### Descrição dos Imports:

- **LightningElement:** Classe base para criar componentes LWC.
- **wire:** Decorador usado para conectar métodos Apex ou outros serviços ao componente.
- **getVehicles:** Método Apex para buscar a lista de veículos.
- **persistPurchase:** Método Apex para persistir a compra de um veículo.
- **ShowToastEvent:** Classe para criar e exibir notificações (toasts) no Salesforce.

## Variáveis

- **vehicles:** Armazena a lista de veículos recuperados do método Apex getVehicles.

## Métodos

```
wiredVehicles({data, error})
```

- **Descrição:** Método decorado com @wire que busca a lista de veículos do método Apex getVehicles. Ele atualiza a variável vehicles com os dados recebidos ou exibe um erro no console caso a operação falhe.
- **Parâmetros:**
  - data: Dados retornados pelo método Apex.
  - error: Erro retornado pelo método Apex, caso ocorra.
- **Funcionamento:**
  - Se data estiver disponível, a lista de veículos é armazenada na variável vehicles.
  - Se ocorrer um erro, ele é exibido no console.

## purchase(event)

- **Descrição:** Método chamado ao clicar no botão de compra de um veículo. Ele persiste a compra utilizando o método Apex persistPurchase.
- **Parâmetros:**
  - event: Evento disparado pelo clique no botão, contendo o ID do veículo no atributo dataset.id.
- **Funcionamento:**
  - Recupera o ID do veículo a partir do evento.
  - Chama o método Apex persistPurchase com o ID do veículo.
  - Exibe uma notificação de sucesso ou erro dependendo do resultado da operação.

## showToast(title, message, variant)

- **Descrição:** Método para exibir notificações (toasts) no Salesforce.
- **Parâmetros:**
  - title: Título da notificação.
  - message: Mensagem da notificação.
  - variant: Tipo da notificação (SUCCESS, ERROR, etc.).
- **Funcionamento:**
  - Cria um evento ShowToastEvent com os parâmetros fornecidos.
  - Dispara o evento para exibir a notificação.

## Exemplo

Abaixo está um exemplo de como o componente pode ser utilizado em um arquivo HTML:

```
<template>
  <template if:true={vehicles}>
    <ul>
      <template for:each={vehicles} for:item="vehicle">
        <li key={vehicle.Id}>
          {vehicle.Name}
          <button data-id={vehicle.Id} onclick={purchase}>Comprar</button>
        </li>
      </template>
    </ul>
  </template>
  <template if:false={vehicles}>
    <p>Carregando veículos...</p>
  </template>
</template>
```

## Diagrama de Dependências

O diagrama abaixo ilustra as dependências do componente ViewVehicles:

```
classDiagram
    class ViewVehicles {
        - vehicles
        + wiredVehicles({data, error})
        + purchase(event)
        + showToast(title, message, variant)
    }
    ViewVehicles --> LightningElement
    ViewVehicles --> getVehicles
```

```
ViewVehicles --> persistPurchase  
ViewVehicles --> ShowToastEvent
```

## Notas

- O método `location.reload()` está comentado no código. Ele pode ser utilizado para recarregar a página após uma compra bem-sucedida, mas isso pode não ser necessário dependendo do comportamento desejado.
- Certifique-se de que os métodos Apex `getVehicles` e `persistPurchase` estejam devidamente implementados e acessíveis no Salesforce.

## Vulnerabilidades

- **Erro ao persistir compra:** Caso o método Apex `persistPurchase` falhe, o erro é apenas exibido no console. Considere implementar um tratamento de erro mais robusto.
- **Validação de ID:** O ID do veículo é recuperado diretamente do evento. Certifique-se de que ele seja validado antes de ser enviado ao método Apex para evitar problemas de segurança.