

03

Recarregando a tela

Transcrição

Estou com a ferramenta de desenvolvedor do browser aberto, em que podemos visualizar a tabela da nossa aplicação. Apenas o agendamento do Alexandre (`id=6`) tem na última coluna o `false` . Vamos pedir para reenviar o pedido.



O reenvio será feito com sucesso e a tabela será atualizada na coluna "confirma". No entanto, o item continua sinalizado em vermelho e o botão de "Reenviar" ainda está lá.



O ideal seria que após termos reenviado o pedido de agendamento, a tela fosse recarregada, pegasse o dado atualizado e o item não fosse marcado em vermelho. Faremos isto no arquivo `agendamento.controller.js`. Atualmente, o trecho referente ao pop up está assim:

```
CarroService.salvarPedido(agendamentoFinalizado).then(function(dados){

  DatabaseValues.setup();
  DatabaseValues.bancoDeDados.transaction(function(transacao){
    transacao.executeSql("UPDATE agendamentos SET confirmado = 'true' WHERE id = ? ", [agendamento.i
  })

  $ionicPopup.alert({
    title : 'Parabens',
    template : 'Seu agendamento foi confirmado com sucesso'
  })

  }, function(erro){

    $ionicPopup.alert({
      title : 'Ops!',
      template : 'O servidor continua com erro. Tente mais tarde'
    })
  })
})
```

Depois que aperto `alert`, podemos chamar a função `then()`. Dentro, teremos uma `function()`:

```
$ionicPopup.alert({
  title : 'Parabens',
  template : 'Seu agendamento foi confirmado com sucesso'
}).then(function(){

})
```

Para mudarmos de rota, já vimos que precisamos do `$state`, mas ainda não o temos injetado na `controller`. Faremos isto a seguir:

```
angular.module('starter')
.controller('AgendamentosController', function($scope, DatabaseValues, CarroService, $ionicPopup, $s
//...
```

Agora, podemos usar o `$state` no `alert`. Como nós não queremos mudar de tela, usaremos o `$state.current`. E sim, queremos que a tela seja recarregada.

```
$ionicPopup.alert({
  title : 'Parabens',
  template : 'Seu agendamento foi confirmado com sucesso'
}).then(function(){
```

```
$state.go($state.current, {}). {reload: true})
})
```

Com `{reload: true}` , ele recarregará a tela. Vamos testar para ver como ficou.

No browser, não teremos mais nenhum agendamento que esteja marcado como `false` . A vantagem de trabalharmos com WebSQL é que podemos fazer *queries*. Ele nos fornece um prompt para digitarmos o SQL.

The screenshot shows the Chrome DevTools Application tab. On the left, there's a tree view of storage: Manifest, Service Workers, Clear storage, Storage (Local Storage, Session Storage, IndexedDB), Web SQL (selected), and Cookies. Under Web SQL, there's a database named 'aluraCar' with two tables: 'agendamentos' and 'sqlite_sequence'. The 'agendamentos' table is expanded, showing two rows of data.

id	id	nome	endereco	email	dataAgendamento	mo...	pr...	confir...
1	1	Lazaro ...	Rua Tobi...	lazaro@alura.co...	Sun Oct 16 2016 01:00:00 GMT-02...	Azer...	8...	true
2	2	Paula C...	Rua Tobi...	paula@alura.com...	Thu Oct 20 2016 00:00:00 GMT-02...	Oni...	3...	true

Nós digitaremos a seguinte query:

```
UPDATE agendamentos SET confirmado = 'false' WHERE id = 3
```

Com isso, ele alterará a coluna "confirmado" da tabela `agendamentos` , onde o id for igual a `3` . Veremos que a tabela foi alterada.

The screenshot shows the Chrome DevTools Application tab after the update. The 'agendamentos' table now has three rows. The row with id 3 has 'confirmado' set to false.

id	id	nome	endereco	email	dataAgendamento	mo...	pr...	confir...
1	1	Lazaro ...	Rua Tobi...	lazaro@alura.co...	Sun Oct 16 2016 01:00:00 GMT-02...	Azer...	8...	true
2	2	Paula C...	Rua Tobi...	paula@alura.com...	Thu Oct 20 2016 00:00:00 GMT-02...	Oni...	3...	false
3	3	Paula C...	Rua Tobi...	paula@alura.com...	Thu Oct 20 2016 00:00:00 GMT-02...	Oni...	3...	false

Repetiremos o mesmo processo com o id `2` e `6` no Prompt, lembrando que ele aproveitará a última SQL que foi feita. Agora, teremos três id com `false` e três `true` . Precisamos que ele pegue esses dados e atualize a tela.



Clicaremos no botão "Reenviar" no id 2 e o agendamento será realizado com sucesso. Clicaremos no "Ok" no pop up com a mensagem de sucesso, mas o recarregamento não será realizado. Quando trabalhamos com o Ionic, ele faz um quest das nossas Views e por isso, não nos deixa carregar. Teremos que desabilitar o quest desta View, para poder recarregá-la. Ele receberá um atributo: `cache-view="false"`. Nessa View, não queremos cache.

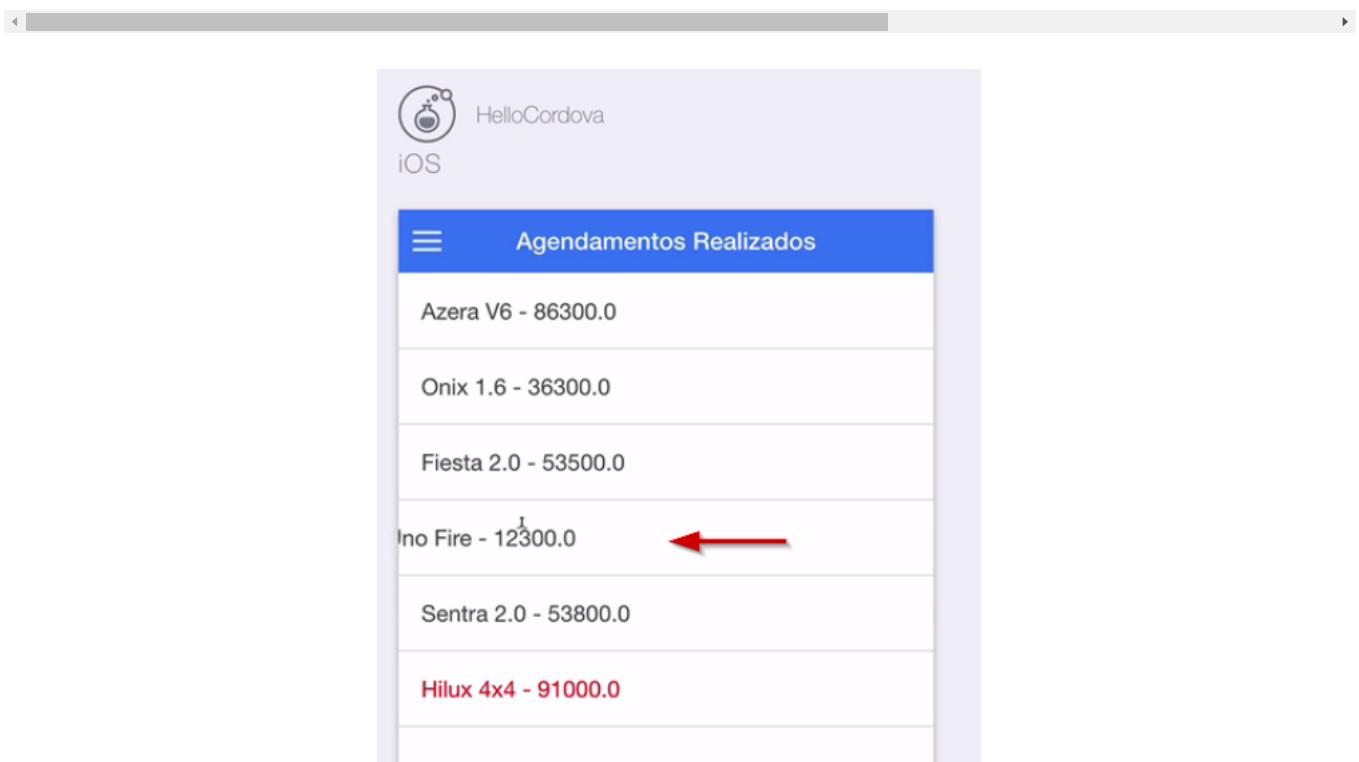
```
<ion-view cache-view="false" view-title='Agendamentos Realizados'>
  <ion-content>
    <ion-list>
      <ion item ng-class="{'falha' : agendamento.confirmado == 'false'}" ng-repeat="agendamento in agenda">
        <ion-option-button ng-click="reenviar(agendamento)" class="button-positive">Reenviar</ion-option-button>
        {{agendamento.modelo}} - {{agendamento.preco}}
      </ion-item>
    </ion-list>
  </ion-content>
</ion-view>
```

Agora, vamos testar a aplicação e reenviar o pedido de agendamento para os outros item marcados em vermelho. Após clicarmos na mensagem de "Parabéns", veremos que a tela foi carregada novamente.



Veja que o botão foi fechado também. Nós também podemos ocultar o botão dos itens que já forma confirmados. Faremos isso adicionando `ng-hide` dentro do `<ion-item>`.

```
<ion item ng-class="{'falha' : agendamento.confirmado == 'false'}" ng-repeat="agendamento in agendamentos" style="border: 1px solid #ccc; border-bottom: none;">
  <ion-option-button ng-hide="agendamento.confirmado == 'true'" ng-click="reenviar(agendamento)" class="button-assertive">
    {{agendamento.modelo}} - {{agendamento.preco}}
  </ion-option-button>
</ion-item>
```



Enquanto no "Hilux", em que aparece `false` na tabela, ainda é possível ver o botão "Reenviar".

```
<ul class="list">
  <li class="item item-toggle" ng-repeat="acessorio in listaDeAcessorios" >
    {{acessorio.nome}} - R{{acessorio.preco | currency }}
    <label class="toggle toggle-assertive">
```

```
<input type="checkbox" ng-model="isMarcado" ng-click="mudou(aceessorio, isMarcado)" >
<div class="track">
  <div class="handle"></div>
</div>
</label>
</li>
```