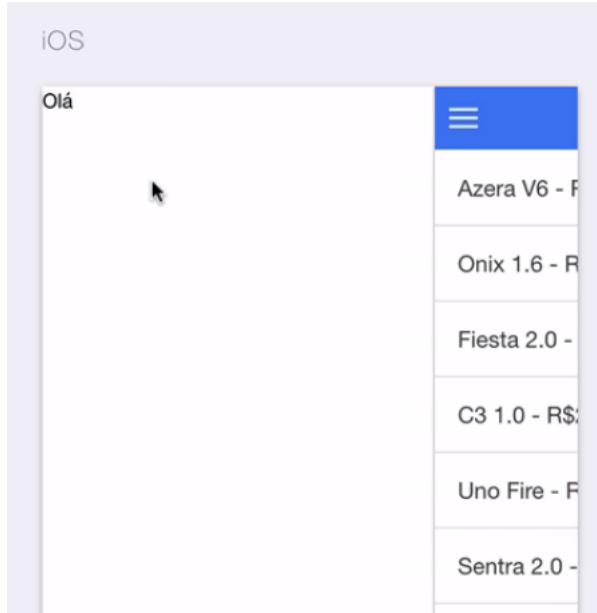


02

Incrementando nosso SideMenu com dados

Transcrição

O menu da nossa aplicação já está funcionando. Quando clicamos no ícone, ele abre com o conteúdo.

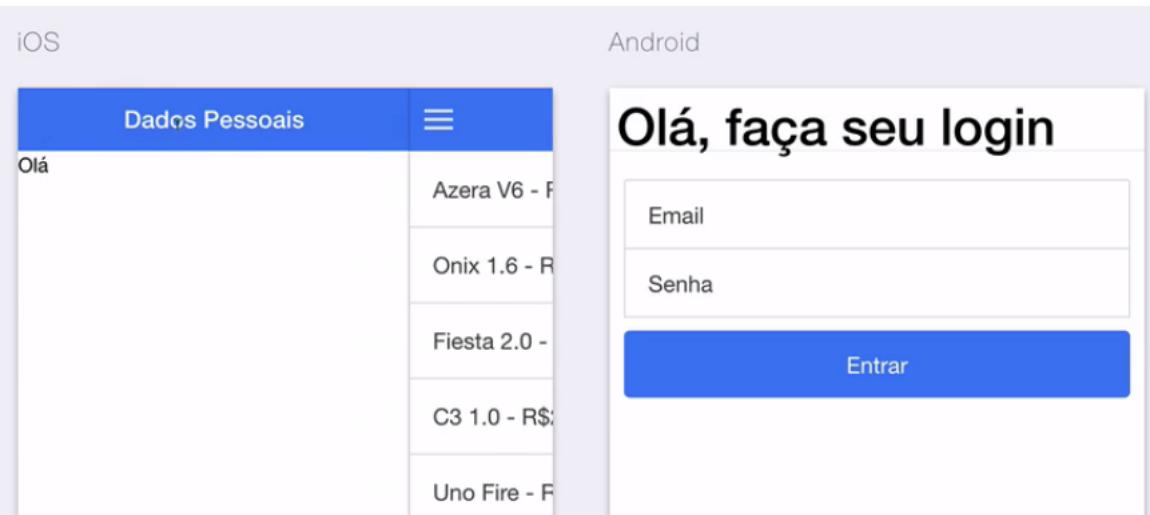


Agora, já podemos personalizar esse menu. Podemos colocar uma barra, um cartão, e outros tipos de componentes na tag `<ion-content>`. Vamos começar com uma barra, indicando que estamos no menu. Adicionaremos a barra.

No arquivo `menu.html`, usaremos a tag `<ion-header-bar>`. Também adicionaremos a classe `bar bar-positive` para deixar a barra azul.

```
<ion-side-menu side="left">
  <ion-header-bar class='bar bar-positive'>
    <h1 class="title">Dados Pessoais</h1>
  </ion-header-bar>
  <ion-content>
    Olá
  </ion-content>
</ion-side-menu>
```

Agora, já veremos a diferença no menu.

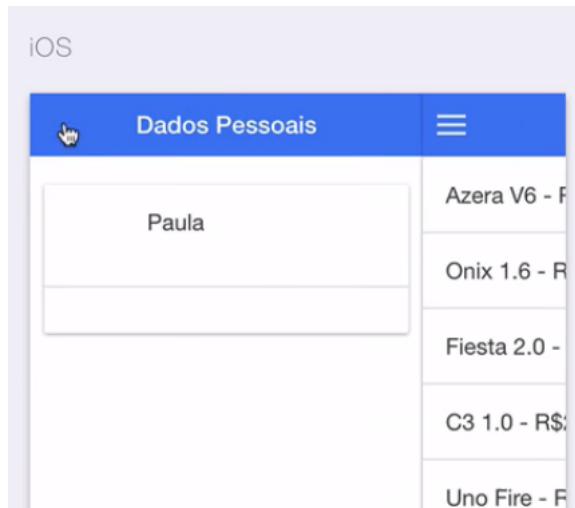


No conteúdo, inicialmente, quero que seja possível visualizar os dados que foram preenchidos no login e cadastrados no Back-End. Nós precisaremos apresentar, usando os componentes que vimos na documentação. Um deles é o [Card showcase](http://ionicframework.com/docs/components/#card-showcase) (<http://ionicframework.com/docs/components/#card-showcase>).

Este componente tem uma `div` pai e outras duas filhas. Iremos adicioná-las dentro do `<ion-content>`. Usaremos também a classe `list card`, `item item-avatar` e `item item-body` do CSS.

```
<ion-content>
  <div class="list card">
    <div class="item item-avatar">
      Paula </div>
    <div class="item item-body"><></div>
  </div>
</ion-content>
```

Os dados foram da controller de login para a controller de menu. Adicionaremos o nome `Paula` para verificar se ele já poderá ser visualizado.



Lembrando que temos um escopo, mas cada controller terá um escopo.

Vamos fazer algumas alterações no `controller.js` e inserir o `rootScope` para termos acesso a ele em qualquer controller que eu queira. Começaremos injetando o `$rootScope` dentro do `function` na `LoginController`.

```
angular.module('starter')
```

```
.controller('LoginController', function($scope, CarroService $ionicPopup, $state, $rootScope){
    //codigo omitido
})
```

Depois, o `$rootScope` irá receber os `dados.usuario`, que se refere ao email e o nome que vêm do Back-End:

```
CarroService.realizarLogin(dadosDoLogin).then(function(dados){
    $rootScope.usuario = dados.usuario;
    $state.go('app.listagem');
    //...
})
```

Agora, podemos passar o `$rootScope` para a controller do menu, que criaremos a seguir no `routes.js`. Ela receberá o nome de `MenuController`.

```
.state('app', {
    url : 'app',
    templateUrl : 'templates/menu.html',
    abstract: true,
    controller: 'MenuController'
})
```

No fim do arquivo, criaremos a controller:

```
angular.module('starter')
.controller('MenuController', function($rootScope, $scope){
    $scope.usuarioLogado = $rootScope.usuario;
})
```

Observe que injetamos o `$rootScope` e o `$scope` no `function` - uma das vantagens do Angular é poder fazer essas injeções à medida que for necessário, sem gerar uma ordem. Usamos o `usuarioLogado` para criar uma diferenciação.

Poderemos usar a variável `usuarioLogado` na View dessa controller. No `menu.html`, vamos substituir o nome `Paula` por um dinâmico. Para deixá-lo em destaque, vou adicionar a tag `<h2>` e `<h3>`.

```
<ion-content>
    <div class="list card">
        <div class="item item-avatar">
            <h2> {{usuarioLogado.nome}}</h2>
            <h3> {{usuarioLogado.email}}</h3>
        </div>
        <div class="item item-body"></div>
    </div>
</ion-content>
```

Após salvarmos as alterações, veremos como ficou o menu.



Conseguimos visualizar o nome do usuário que temos cadastrados, da mesma forma, poderemos vê-los no Android.

Nós conseguimos personalizar o nosso menu e podemos agora personalizar e incluir mais informações. Usamos o `$rootScope`, em que pegamos o dado da `controller` e usamos na controller do menu.

Até a próxima!