

Resumo

Transcrição

Olá, pessoal! Tudo bem? Vamos fazer uma revisão do conteúdo que vimos nesta aula. Nós começamos com a criar o menu lateral e depois, inserimos conteúdo.

Nós criamos um template um pouco diferente no `menu.html`. Trata-se de um `<ion-side-menus>`, o menu lateral - enquanto nas outras Views, usavamos a tag `<ion-view>`. Dentro da estrutura do `<ion-side-menus>`, temos duas tags principais: `<ion-side-menu-content>`, que cria a barra do menu, e o `<ion-side-menu>`, em que encontraremos o cabeçalho e o conteúdo.

```
<ion-side-menus>
  <ion-side-menu-content>
    <ion-nav-bar>
      <ion-nav-buttons side="left">
        <button></button>

      </ion-nav-buttons>
    </ion-nav-bar>

    <ion-nav-view name="menuContent"> </ion-nav-view>

  </ion-side-menu-content>

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

</ion-side-menus>
```

Dentro do `<ion-side-menu-content>`, criamos uma `<ion-nav-view>` e demos o nome de `menuContent`. Lembrando que o menu é um template de rota abstrata, conforme declaramos o `routes.js`, a partir disso, vamos colocar as Views para rodar em cima da `nav-view`.

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

```
.state('app.listagem',{
  url : '/listagem',
  templateUrl : 'templates/listagem.html',
  controller: 'ListagemController'
})
```

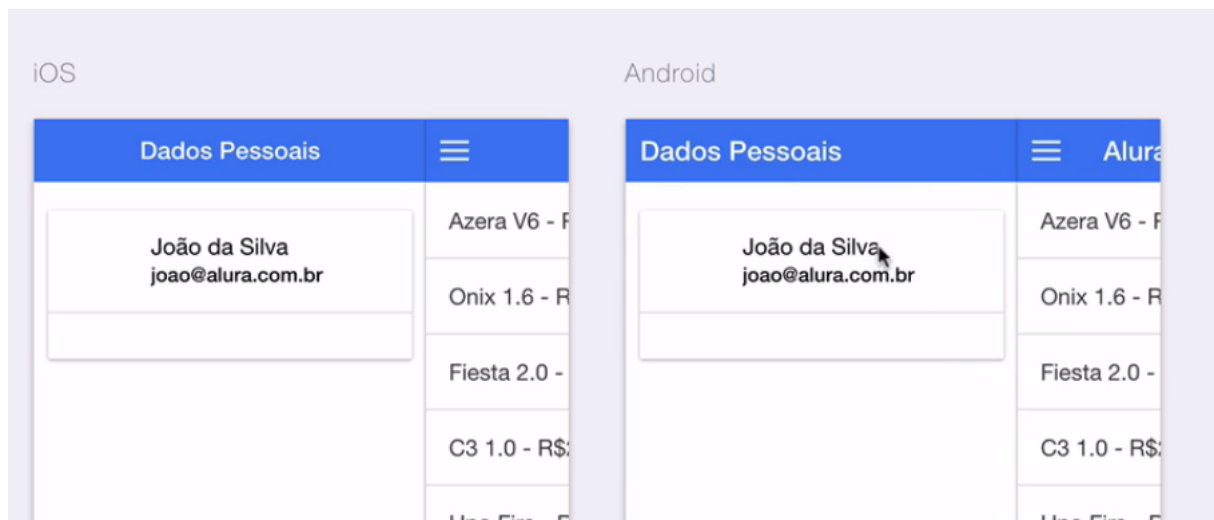
Também fizemos uma refatoração para pegar a View da listagem. Nós trocamos a navegação para atender a nova demanda que o nosso cliente nos pediu.

Também personalizamos o menu. Nós podemos criar componentes neste menu. Nós colocamos um card, que recebeu os dados do login.

No `controller.js`, adicionamos o `$rootScope`, para pegar o dado da `controller` e usarmos na `controller` do menu. Usamos também o `$scope` para receber o objeto para aparecer na tela.

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

E no fim, conseguimos que os dados fossem visualizados no menu.



Cada vez que o usuário entrar, vamos pegar o nome do login.