Curso: Spring Boot com Ionic - Estudo de Caso Completo

https://www.udemy.com/user/nelio-alves

Prof. Dr. Nelio Alves

Capítulo: Aplicação Ionic - Parte 2/2

Objetivo geral:

- Construir a segunda etapa do aplicativo de pedidos usando Ionic. Escopo:
  - Fluxo de finalização de pedido
    - Escolher endereço de entrega
    - Escolher forma de pagamento
    - Conferir pedido
    - Tela de confirmação
  - Janela de loading
  - Infinite scroll
  - Upload de foto de perfil com Cordova

Tela de escolha de endereço

Checklist:

- Em cart.html, acrescentar o botão de finalizar pedido
- Criar EnderecoDTO
- Criar página pick-address
  - Definir script básico com dados mockados
  - Definir HTML

Buscando endereços do banco de dados

Checklist:

- **IMPORTANTE! Refatoração**:
  - Em ClienteService, retirar a tipagem do método findByEmail
  - Em ProfilePage, corrigir a instanciação de ClienteDTO
- Em PickAddressPage, atualizar o carregamento dos endereços do cliente logado

Armazenamento dos dados do pedido

Checklist:

- Criar RefDTO, PagamentoDTO, ItemPedidoDTO e PedidoDTO
- Em PickAddressPage, instanciar um PedidoDTO a partir dos dados existentes
- Em pick-address.html, criar um binding em cada item para um método nextPage
- Em PickAddressPage, criar o método nextPage
  - Mostrar o pedido no console

Tela de escolha de pagamento

Checklist:

- Criar página Payment
  - Definir script para formulário (PagamentoDTO)
  - Definir HTML
- Em PickAddressPage, fazer a navegação para PaymentPage
  - Nota: enviar o objeto Pedido
- Em PaymentPage, no método nextPage, mostrar o pedido no console

Tela de confirmação de pedido

Checklist:

- Em ClienteService, criar método findById
- Criar página OrderConfirmation
  - Definir script (nota: método checkout por enquanto mostrando o pedido no console)
  - Definir HTML
- Em PaymentPage, no método nextPage, fazer a navegação e passar o pedido (use setRoot)

Salvando pedido

Checklist:

- Criar PedidoService com método insert
- Em order-confirmation.module.ts, registrar PedidoService nos providers
- Em OrderConfirmationPage, no método checkout, realizar a inserção do pedido
  - Testar se o location do novo recurso está sendo retornado
  - Lembre-se de esvaziar o carrinho
  - Note que o email de confirmação é enviado ao usuário

Informando pedido registrado ao usuário

Checklist:

- Em OrderConfirmationPage, incluir código para capturar o código do pedido
- Em order-confirmation.html, incluir HTML para mostrar o código do pedido
- Inclua botões de voltar na tela

Componente loading

Referências:

https://ionicframework.com/docs/components/#loading

Checklist:

- Injetar um LoadingController no controlador da página
- Incluir o método presentLoading() no controlador da página, com os devidos ajustes
  - Fazer o método retornar o objeto loader

- No local desejado, chamar o método presentLoading()
    - No local apropriado, chamar o método dismiss a partir do objeto loader

Refresher

Referências:

https://ionicframework.com/docs/api/components/refresher/Refresher/

Checklist:

- No HTML, incluir o elemento ion-refresher como primeiro filho de ion-content
- Incluir o método doRefresh no controlador da página, com os devidos ajustes

Infinite scroll

Referências:

https://ionicframework.com/docs/api/components/infinite-scroll/InfiniteScroll/

Checklist:

- Em ProdutoService, refatorar o método findByCategoria, de modo que receba também os parâmetros página e número de registros por página
- No HTML, incluir o elemento ion-infinite-scroll como último filho de ion-content
- Incluir o método doInfinite no controlador da página, com os devidos ajustes

Usando Cordova para tirar foto com a câmera

Referências:

https://ionicframework.com/docs/native/camera/
https://blog.ionicframework.com/10-minutes-with-ionic-2-using-the-camera-with-ionic-native/

Checklist:

- **ionic cordova platform add browser --save**
- **ionic cordova plugin add cordova-plugin-camera**
- **npm install --save @ionic-native/camera**
- Em profile.module.ts, incluir Camera nos providers
- Em ProfilePage, incluir o método getCameraPicture
    - Usar formato PNG
    - Desabilitar o botão enquanto a câmera estiver ligada
- Em profile.html, incluir HTML para mostrar a foto
- Para testar: **ionic cordova run browser**

Fazendo upload da foto

Referências:

https://gist.github.com/fupslot/5015897

Preparação:

- Se seu backend estiver em desenvolvimento/teste, inclua provisoriamente as credenciais AWS no seu arquivo properties do projeto STS (**LEMBRE-SE DE APAGAR DEPOIS**)
- Recomendável: teste primeiro no Postman se o envio de foto está funcionando

Checklist:

- Criar ImageUtilService com o método para converter base64 para arquivo (código abaixo)
  - Em app.module.ts, declarar ImageUtilService nos providers
- Em ClienteService, incluir um método uploadPicture para enviar a foto
- Em profile.ts, incluir um método sendPicture que chama o uploadPicture do ClienteService
- Em profile.html, incluir botão para enviar foto com binding para o método sendPicture

```
import { Injectable } from "@angular/core";

@Injectable()
export class ImageUtilService {

    dataUriToBlob(dataURI) {
        var byteString = atob(dataURI.split(',')[1]);
        var mimeString = dataURI.split(',')[0].split(':')[1].split(';')[0]
        var ab = new ArrayBuffer(byteString.length);
        var ia = new Uint8Array(ab);
        for (var i = 0; i < byteString.length; i++) {
            ia[i] = byteString.charCodeAt(i);
        }
        return new Blob([ab], {type: mimeString});
    }
}
```