

Synesthesia Vision

Manual do Software

Sumário

Introdução	2
Instalação	3
Diretórios	4
Pages	5
Bluetooth Connection Verify Page	5
Horários Page	7
Modals	8
Áudio Provider 3	9
Synesthesia vision Page	10
Providers	11
Áudio Provider:	11
Bluetooth Provider:	11
Bus Integration Provider	12
Permission Provider	13
Text To Speech Provider	13
Weather Forecast	13

Introdução

Este documento foi criado com o intuito de fornecer as informações obtidas durante todo o desenvolvimento do projeto para auxiliar futuros “colaboradores” do código-fonte. Neste arquivo estão especificadas as funcionalidades de cada módulo implementado e as suas respectivas explicações com o propósito de fornecer um guia de construção de novas funcionalidades.

Instalação

- **Requisitos:**
 - Ionic 3;
 - Node.js;
- **Tutorial:**

Diretórios

- App
- Pages
- Pipes/date
- Providers

Pages

As páginas são responsáveis por interagir diretamente com o usuário.

São compostas por três arquivos:

- .html: Responsável pela interface com o usuário.
- .scss: Responsável por estilizar o conteúdo html.
- .ts: Responsável pela funcionalidade do sistema.

Bluetooth Connection Verify Page

Métodos:

- ***ionViewDidLoad()***: Executa o som inicial.
- ***ionViewWillEnter()***: Verifica se está o dispositivo está conectado com algum dispositivo bluetooth sempre que a página é carregada. Caso esteja, ele irá desconectar.
- ***startScanning()***: Verifica dispositivos pareados e não pareados. Se um dispositivo pareado tiver o nome "Synesthesia", o método procurará um endereço no localStorage.
- ***selectDevice()***: Exibe um alerta perguntando se o usuário deseja se conectar a um endereço que foi selecionado pelo usuário. Redireciona para a tela principal se o usuário desejar se conectar
 - **@param** address
 - **@param** device
- ***disconnect()***: Desconecta do bluetooth
- ***synesthesia()***: Irá carregar a página Synesthesia vision Page.
- ***showAlert()***: Cria um alert informando um erro, com a mensagem de erro do parâmetro.
 - **@param** message
- ***createLoading()***: Cria o spinner de carregamento.
- ***saveAddress(device)***: Salva o endereço do dispositivo pressionado caso haja uma conexão bem sucedida.
 - **@param** address
- ***checkAddress()***: Verifica se o local storage tem o endereço do dispositivo do synesthesia.

- ***autoConnect(address: string)***: Se conecta a um endereço bluetooth já conhecido e redireciona para a página principal
 - **@param** address
- ***checkEnabledBluetooth()***: Verifica se o bluetooth está ativado. Se não estiver, irá ativá-lo.

Horários Page

Métodos:

- ***ionViewDidLoad()***: requisita a primeira busca de paradas, caso não tenha sido ainda
- ***speakData()***: Realiza o tratamento de fala do tempo restante do ônibus selecionado
- ***getLinhas()***: Realiza a busca pelas paradas próximas
- ***getParadas()***: Exibe a lista de paradas, sendo ordenada pela distância, da mais perto para a mais longínqua
 - **@param** recarregando
- ***getHorários()***: Exibe a lista de ônibus da parada selecionada
 - **@param** labelParada
- ***transformTime()***: Converte o valor da hora recebido em horas e minutos
 - **@param** value
- ***openModalRenomear()***: Exibe o quadro de renomeio de parada
 - **@param** labelParada
 - **@param** nombreParada

Modals

Métodos:

- ***setRenomear()***: Define o nome digitado como um apelido para a parada selecionada
- ***dismiss()***: Fecha o quadro de renomeio de parada

Áudio Provider 3

Métodos:

- **setUpAudio():** Realiza a conversão para um arquivo compatível
 - @param bufferedContent
- **playAudioTrack():** Cria um som estéreo e o carrega no aplicativo
 - @param track
- **playSound():** Inicializa todos os serviços de sonorização, realizando as devidas checagens
- **zeraDistancia():** Zera os valores presentes no sensor
- **stopRunningSound():** Finaliza a repetição de execução do som

Synesthesia vision Page

Métodos:

- **ionViewDidLoad():** Realiza a checagem de permissões
- **ionViewWillLeave():** Realiza a desconexão do módulo bluetooth
- **ionViewWillEnter():** Obtém os dados do bluetooth
- **toggleStatusButton():** Realiza a troca do texto do botão ao inicializar a sonorização do aplicativo
- **playSound():** Inicializa a vibração do telefone e a sonorização
- **stopSound():** Encerra a vibração do telefone e a sonorização
- **checkWeather():** Realiza a checagem de clima
- **increaseFrequency():** Aumenta a frequência de beeps sonorizados
- **decreaseFrequency():** Diminui a frequência de beeps sonorizados
- **getBluetoothData():** Realiza o processamento dos dados recebidos pelos botões
- **getFunction():** Chama a função de acordo com o parâmetro recebido
 - **@param** bluetoothData
- **getParadaProxima():** Exibe uma lista com as paradas próximas
- **speakLuminosity():** Realiza o tratamento e informa se o ambiente está claro ou escuro
 - **@param** dataBuffer
- **voltarConexaoBluetooth():** Retorna à página de conexão.

Providers

Responsável por fazer a conexão com as API's, gerenciar armazenamento, autenticação e etc.

Áudio Provider:

Função: Responsável por fazer a geração de som.

Importações:

- *Web Audio API:* Processa e sintetiza áudio em aplicativos Web.

Referências: <https://www.w3.org/TR/webaudio/>

Bluetooth Provider:

Função: Este provider habilita a comunicação serial via Bluetooth. É utilizado para estabelecer conexões entre Android ou iOS e um Arduino, receber e tratar os dados que os sensores do óculos enviam para o dispositivo através do bluetooth.

Importações:

- *Native Storage:* Armazena os dados no Local Storage do dispositivo.

Referências: <https://ionicframework.com/docs/native/native-storage/>

- *Bluetooth Serial:* habilita a comunicação serial via Bluetooth.

Referências: <https://ionicframework.com/docs/native/bluetooth-serial/>

Bus Integration Provider

Função: Responsável por realizar a busca por paradas que estão localizadas próximas ao usuário.

Importações:

- *Http:* Cordova / Phonegap plugin para comunicação com servidores HTTP.

Referências: <https://ionicframework.com/docs/native/http/>

- *Geolocation:* Fornece informações sobre a localização do dispositivo, como latitude e longitude.

Referências: <https://ionicframework.com/docs/native/geolocation/>

- *Alert Controller:* Faz o controle e personalização de alerts, ou seja, diálogos onde são apresentados aos usuários informações ou coleta informações do usuário usando inputs.

Referências:

<https://ionicframework.com/docs/api/components/alert/AlertController/>

- *Loading Controller:* Faz o controle e personalização de um overlay ou loading spinner que pode ser usada para indicar atividade enquanto bloqueia a interação do usuário.

Referências:

<https://ionicframework.com/docs/api/components/loading/LoadingController/>

Permission Provider

Função: Realiza a verificação de permissões do Android.

Importações:

- *Android Permissions:* Este plugin é projetado para suportar o novo mecanismo de verificação de permissões do Android.

Referências: <https://ionicframework.com/docs/native/android-permissions/>

Text To Speech Provider

Função: Responsável por transmitir informações em áudio para o usuário.

Importações:

- *Text To Speech:* Plugin Text to Speech.

Referências: <https://ionicframework.com/docs/native/text-to-speech/>

Weather Forecast

Função: Verifica e informa ao usuário as condições climáticas atuais.

Importações:

- *HTTP:* Responsável por fazer a comunicação com servidores http.

Referências: <https://ionicframework.com/docs/native/http/>

- *Alert Controller:* Faz o controle e personalização de alerts, ou seja, diálogos onde são apresentados aos usuários informações ou coleta informações do usuário usando inputs.

Referências:

<https://ionicframework.com/docs/api/components/alert/AlertController/>

- *Text To Speech Provider:* Responsável por transmitir informações em áudio para o usuário.

- *Geolocation:* Fornece informações sobre a localização do dispositivo, como latitude e longitude.

Referências: <https://ionicframework.com/docs/native/geolocation/>

- *Location Accuracy:* Solicita a ativação/alteração dos Serviços de Localização ao acionar um diálogo nativo (alert) a partir do aplicativo, evitando a necessidade de o usuário deixar seu aplicativo para alterar as configurações de local manualmente.

Referências: <https://ionicframework.com/docs/native/location-accuracy/>

- *Mobile Accessibility*: Expõe o status de vários recursos de acessibilidade de sistemas operacionais móveis. Usado também para que o aplicativo envie uma string para ser falada pelo leitor de tela ou um comando para impedir que o leitor de tela fale.

Referências: <https://ionicframework.com/docs/native/mobile-accessibility/>