Main page

개요 : itexi 메인 페이지 상단바에 공지 알림과 홈버튼이 있으며, 홈버튼을 통해 사이드바(루트 페이지?)를 열수 있다. Taxi, carpool방 페이지로 접근할 수 있으며, 하단에는 자신이 탈 예정인 택시/카풀을 표형식으로 나타나있으며, 이를 클릭하면 해당 방의 채팅방으로 페이지가 연결된다.

<ion-header> // 헤더

<ion-navbar color="primary"> // 상단바 색깔 primary

<ion-buttons right> // 버튼 위치 : right

<button ion-button menuToggle> // 버튼 형식 menuToggle

<ion-icon name="menu"></ion-icon> // menu icon 사용

</button>

</ion-buttons>

<ion-title>iTaxi</ion-title> // 타이틀은 iTaxi

</ion-navbar>

<div id="notice" ion-item> // 공지 알림을 위한 구역

<ion-icon name="ios-information-circle-outline" item-start></ion-icon> // ios-information-circle-outline icon 사용, 위치는 맨 왼쪽

{{notification}} // notification에 있는 string data를 표시 {{…}}는 변수안에 있는 내용을 string으로 표시함.

</div>

</ion-header>

<ion-content> // main.html content 부분(body)

<div class="menu"> // div class를 menu로 설정

<ion-row>

<ion-col col-6>

<div id="taxi" (click)="takeTaxi()"> // div id는 taxi 이 구역 클릭시 takeTaxi() 함수 실행

<img src="assets/img/main\_taxi.png" alt="taxi"> // alt : 이미지가

<span>택시</span>

<span>탑승</span>

</div>

</ion-col>

<ion-col col-6>

<div id="cpool" (click)="takeCarpool()"> // div id는 cpool 이 구역 클릭시 takeCarpool() 함수실행

<span>카풀</span>

<span>탑승</span>

<img src="assets/img/main\_carpool.png" alt="taxi"> // alt?

</div>

</ion-col>

</ion-row>

</div>

<div class="ticketLabel"> // div 클래스 tiketLabel

<ion-icon color="secondary" name="person"></ion-icon> // color는 secondary icon 이름이 person인 icon 사용

탑승예정

</div>

<!-- 카드 예제 -->

<div \*ngFor="let room of rooms | async | roomSort">

//\*ngFor : 앵굴러 For 문; let [실제 for문 내에서 사용될 변수] of [for문 내에서 사용할 변수를 가진 array]; rooms array에서 값들을 하나씩 호출 하위의 작업을 진행 후 반복;

async : firebase 명령어, 실시간으로 데이터 싱크를 맞춤;

roomSort : src/pipes/room-sort/room-sort.ts, 시간순으로 정령

pipe( | )를 이용하여 room 변수 -> async -> roomSort 순으로 작업수행

<div \*ngIf='room.departDate + room.departTime >= nowDate + nowTime'>

// room.departDate + room.departTime 이 nowDate + nowTime보다 크거나 같을경우 div 표시

<ion-card class="ticket" (click)="goChatroomPage(room)"> // class가 tiket인 ion-card, 이부분을 클릭시 input 값이 room인 goChatroomPage 함수 실행

<ion-row>

<ion-col col-1>

<img \*ngIf="room.transportType == 'taxi'" src="assets/img/main\_taxi\_mini.png" alt="taxi"> // room.transportType 이 taxi 일 경우, 이 이미지를 표시

<img \*ngIf="room.transportType == 'carpool'" src="assets/img/main\_carpool\_mini.png" alt="carpool"> // room.transportType 이 carpool 일 경우, 이 이미지를 표시

</ion-col>

<ion-col col-4 text-left> // 해당 ion-col의 text는 왼쪽정렬

<div class="text">

{{ room.departDate | date: 'M.dd ' }} {{ dateServices.getKToday(room.departDate) }}요일

// {{ room.departDate | date: 'M.dd ' }} : room.departDate 값을 M.dd 형식으로 string으로 표시; {{ dateServices.getKToday(room.departDate) }}요일 : dataServices에 있는 getKToday함수를 input 값 room.departDate로 준후 나오는 요일을 출력값으로하여 요일 표시

</div>

<div class="text"> // class는 text로 설정

{{ room.departTime }} // room.departTime을 string으로 표시

</div>

</ion-col>

<ion-col col-5>

<div class="text">

{{ room.depart }} // room.depart을 string으로 표시

</div>

<div class="text">

<span>&gt;</span>&nbsp;&nbsp;{{ room.arrive }} // ?

</div>

</ion-col>

<ion-col col-2 text-center>

<div class="text" \*ngIf="room.transportType == 'taxi'"> // class ㅣ text, \*nfIf : room.transportType가 taxi일 경우 이 div 표시

택시

</div>

<div class="text" \*ngIf="room.transportType == 'carpool'"> // class ㅣ text, \*nfIf : room.transportType가 carpool일 경우 이 div 표시

카풀

</div>

<div class="text">

{{ room.currentPeople }}/{{ room.capacity }} // room.currentPeople 과 room.capacity를 string으로 출력

</div>

</ion-col>

</ion-row>

</ion-card>

</div>

</div>

<!-- <button ion-button secondary menuToggle>Toggle Menu</button> -->

</ion-content>

main.ts

import { Component, ViewChild } from '@angular/core';

import { IonicPage, NavController, NavParams, MenuController, LoadingController, Content, Platform } from 'ionic-angular';

import { AngularFireDatabase, FirebaseListObservable, FirebaseObjectObservable } from 'angularfire2/database';

import { ChatRoomPage } from '../../pages/chatroom/chatroom';

import { ListPage } from '../../pages/list/list';

import { UsersProvider } from '../../providers/users/users';

import { DateProvider } from '../../providers/date/date';

import { RoomsProvider } from '../../providers/rooms/rooms'

@IonicPage() // ?

@Component({

selector: 'page-main', // 무슨 페이지와 연결시키질 선언하는

templateUrl: 'main.html', // 무슨 html

}) // 무슨

export class MainPage {

@ViewChild(Content) content: Content; // resize 관련

rooms: FirebaseListObservable<Object[]>; // rooms : 택시, 카풀 방들;

nowDate: string; // 현재 날짜(년-월-일)

nowTime: string; // 현재 시간

notification: string; // 공지내용

public counter = 0;

// constructor(…) : constructor input으로 사용할 라이브러리들을 일종의 변수로 초기화 할 때 사용 c에서의 일종의 #define

constructor(

public navCtrl: NavController,

public navParams: NavParams,

public af: AngularFireDatabase,

public menu: MenuController,

public userServices: UsersProvider,

private dateServices: DateProvider,

public roomServices: RoomsProvider,

public loadingCtrl:LoadingController,

public platform: Platform) {

//여기서부터는 로그인 및 회원가입 페이지를 넘어서 사이드 메뉴를 볼 수 있도록 만들기.

console.log(new Date().getMonth())

}

// ionViewWillEnter() : 해당 페이지(여기서는 main)가 작동할 때 자동으로 실행되는 함수(초기화 용?); this.a : 현재 ts파일에 선언 되어있는 변수 a 일종의 전역변수

ionViewWillEnter() {

this.dateServices.setNow(); // dateServices(== DateProvider)의 setNoew() 함수 실행

this.nowDate = this.dateServices.nowDate; // nowDate를 dateServices.nowDate로 초기화

this.nowTime = this.dateServices.nowTime; // nowTime을 dateServices.nowTime으로 초기화

this.rooms = this.af.list('/rideHistory/' + this.userServices.userInfo['studentID'], {

query:{

startAt: this.dateServices.getYearMonthDayWithDash(), // getYearMonthDayWithDash() : 현재 날짜 반환

orderByChild : 'departDate'

}

}) // rooms를 af(== AngularFireDatabase)을 이용하여 firebase에 있는 list로 초기화한다. 이 때 가저올 list의 경로는 '/rideHistory/' + this.userServices.userInfo['studentID'], firebase의 rideHistory 폴더에 userInfo[‘studentID’]로 반환된 해당 학번으로 된 폴더에 있다. 그리고 해당 경로에서 query 명령어를 통해 해당되는 data를 들고 오는데 기준은 startAt(== 현재시간) 이후부터 정렬기준(== orderByCild)은 departDate가 빠른순으로 정렬함.

this.af.object('/notification/').subscribe( data => this.notification = data.content); // subscribe : push후 응답을 받음 응답(data)를 받을경우 => 이후의 명령어들을 실행함.

this.content.resize(); // 실시간으로 content 갱신하는 함수?

this.menu.enable(true,'myMenu'); // 사이드 사용여부를 결정하는 함수, 여기서는 사용하게끔 되여 있고 해당 사이드바의 이름은 myMenu

}

// ionViewDidLoad(){ console.log("ionViewDidLoad at main.ts"); }

// 아래에 있는 함수들은 클릭시 다른페이지를 호출하는데 사용하는 함수

takeTaxi() { this.navCtrl.push(ListPage, {transportType: 'taxi'}); }

takeCarpool() { this.navCtrl.push(ListPage, {transportType: 'carpool'}); }

goChatroomPage(room){ this.navCtrl.push(ChatRoomPage, {room: room}); }

}

main.moudule.ts

import { NgModule } from '@angular/core';

import { IonicPageModule } from 'ionic-angular';

import { MainPage } from './main';

import { PipesModule } from '../../pipes/pipes.module';

@NgModule({

declarations: [

MainPage,

],

imports: [

IonicPageModule.forChild(MainPage),

PipesModule

],

})

export class MainPageModule {}