

۴.۶ تایپ‌ها و اینترفیس‌های اشتراکی (TypeScript Types & Interfaces)

• فایل مرجع: docs/api-types.ts

◦ شامل export تایپ‌ها و اینترفیس‌ها برای استفاده در پروژه TypeScript / Next.js

۴.۶.۱ تایپ‌ها و اینترفیس‌ها

پاسخ عمومی

```
export type ResultStatus = 'Success' | 'Error' | 'Warning' | 'NotFound';

export interface ResultBase<T> {
  status?: ResultStatus;
  code?: string;
  message?: string;
  data?: T;
  payload?: T;
  listPayload?: T[];
  isSuccess?: boolean;
}
```

احراز هویت (Auth)

```
export interface SendOtpRequest { mobile: string; }
export interface VerifyOtpRequest { mobile: string; otpCode: string; }
export interface VerifyUserModel {
  token: string;
  userName: string;
  userId: number;
  isAdmin: boolean;
  roleCode: number;
}
```

شهر و ایستگاه (City & Terminal)

```
export interface CityDto {
  id: number;
  title: string;
  code: string;
  provinceId?: number;
  enabled: boolean;
  sellerCityId?: number;
  coords?: string;
  etitle?: string;
}

export interface Station {
  city: string;
  terminal: string;
}
```

اتوبوس و صندلی (Bus & Seat)

```
export interface Seat {
  number: number;
  column: number;
  row: number;
  status: "Available" | "BookedForMale" | "BookedForFemale";
}

export interface Financial {
  price: number;
  maxApplicableDiscountPercentage: number;
  availablePaymentMethods: string[];
}
```

```

    commissionPercentage: number;
}

export interface Bus {
  id: string;
  company: string;
  companyCode: string;
  companyToken: string;
  boardingPoint: Station;
  droppingPoints: Station[];
  departureDate: string;
  price: number;
  fullPrice: number;
  type: string;
  availableSeats: number;
  commissionPercentage: number;
  description?: string;
  seates?: Seat[];
  financial?: Financial;
}

export interface ChairLayout {
  floors: number;
  columns: number;
  rows: number;
  numChairs: number;
  columnSpace: number;
  layoutChairs: string;
}

```

سفر و شرکت (Trip & Company)

```

export enum APIType {
  None = 0, Payaneha = 1, Safar724 = 2, Payaneh = 3, PayanehaWS = 4, Payanehacom = 5
}

export interface TripServiceModel {
  id: string;
  title: string;
  numChairs: number;
  date: string;
  departureTime: string;
  srcCity: string;
  destCity: string;
  finalDestCity?: string;
  srcCityCode: string;
  destCityCode: string;
  finalDestCode?: string;
  pathTitle?: string;
  sold: number;
  price: number;
  numFree: number;
  hasAgency?: boolean;
  discountPercent?: number;
  description?: string;
  seller?: Company;
  companyID?: number;
  companyTitle?: string;
  companyCode?: string;
  companyDiscountPercent?: number;
  commissionPercent?: number;
  companyBrand?: string;
}

export interface Company {
  id: number;
}

```

```
title: string;
code: string;
discountPercent: number;
}
```

خرید و مسافر (Buy & Passenger)

```
export interface ResponseBuyTicket {
  bussEntity: Bus;
  sellInfo?: Record<string, unknown>;
  chairLayout: ChairLayout;
  trip?: TripServiceModel;
}

export interface PersonInfo {
  name: string;
  lastName: string;
  nationalCode: string;
  birthDay: string;
  foreign: string;
  genders: string;
  address: string;
  isParent: boolean;
}

export interface SaleTicketRequest {
  selectChaires: string;
  busId: string;
  serviceId: string;
  tokenCompany: string;
  sellerId?: number;
  tripId: string;
  sellerTicketId?: number;
  userId?: number;
  srcCode: string;
  destCode: string;
  useCreditOnSafar724: boolean;
  mobileNumber: string;
  listPerson: PersonInfo[];
}

export interface PassengerInfoDto {
  name: string;
  lastName: string;
  nationalCode: string;
  birthDay: string | number; // ISO date or timestamp
  foreign: boolean;
  genders: number;
  address: string;
  isParent: boolean;
}

export interface ReserveTicketModel {
  userModel?: VerifyUserModel;
  ticketModel?: Record<string, unknown>;
  creditUser?: number;
  discountPercent?: number;
  useCredit?: boolean;
  doConfirm?: boolean;
  amountPay?: string;
  discountAmount?: string;
}
```

```

export interface BusSummary {
  id: string;
  company: string;
  companyCode: string;
  companyToken: string;
  boardingPoint: Station;
  droppingPoints: Station[];
  departureDate: string;
  arrivalDate: string;
  price: number;
  fullPrice: number;
  type: string;
  availableSeats: number;
  description?: string;
  destCityName?: string;
  apiType?: APIType;
  time?: string;
  financial?: Financial;
}

export interface GetListBusSummariesQuery {
  srcCode: string;
  destCode: string;
  dt: string;
  sortIndex: number;
  timeFilter: number;
  minPrice: number;
  maxPrice: number;
  titleCooperative?: string;
}

```

ثابت‌ها (Constants)

```

export const API_ERROR_CODES: Record<string, string> = {
  '0': 'عملیات با موفقیت انجام شد',
  '200': 'عملیات با موفقیت انجام شد',
  '1': 'API خطا در فراخوانی',
  '401': 'توکن احراز هویت ارسال نشده یا نامعتبر است',
  '404': 'منبع مورد نظر یافت نشد',
  '500': 'خطای داخلی سرور',
  '-1': 'خطا در عملیات',
};

```

Comments