

۵.۴ (Booking, Payment) مدل‌های رزرو و پرداخت

طراحی، پاسخ رزرو، مدل بلیط و وضعیت آن و همچنین درخواست و پاسخ پرداخت درگاه است

Endpointهای مرتبط

- POST /api/Buy/SaleTicketsNew
- GET /api/Profile/GetUserTickets
- GET /api/Profile/GetTicketInfo
- POST /api/Payment/SendRequestPayment

۱) درخواست ثبت رزرو — SaleTicketRequest

ورودی API:

POST /api/Buy/SaleTicketsNew

```
interface SaleTicketRequest {
    selectChaires: string; // شماره صندلی‌ها با کاما، مثلاً "1,2,3"
    busId: string;
    serviceId: string;
    tokenCompany: string;
    sellerId?: number;
    tripId: string;
    sellerTicketId?: number;
    userId?: number;
    srcCode: string;
    destCode: string;
    useCreditOnSafar724: boolean;
    mobileNumber: string;
    listPerson: PersonInfo[]; // تعداد با تعداد صندلی برابر
}
```

قواعد مهم

فیلد	نوع	توضیح
selectChaires	string	شماره صندلی‌های انتخاب شده، جدا شده با کاما
busId / serviceId + tokenCompany / sellerId + tripId	—	دقیقاً یکی از این سه گروه باید ارسال شود
listPerson	[]PersonInfo	تعداد عناصر باید با تعداد صندلی‌ها برابر باشد. ساختار هر عنصر در بخش ۵.۵ (مدل‌های خرید و مسافر) تعریف شده است

۲) پاسخ ثبت رزرو — ReserveTicketModel

خروجی POST /api/Buy/SaleTicketsNew در payload زمانی که:

code: "0"

```
interface ReserveTicketModel {
    userModel?: VerifyUserModel;
    ticketModel?: Ticket;
    creditUser?: number;
    disCountPercent?: number;
    useCredit?: boolean;
    doConfirm?: boolean;
}
```

```

    amountPay?: string;
    discountAmount?: string;
}

interface VerifyUserModel {
    token: string;
    userName: string;
    userId: number;
    isAdmin: boolean;
    roleCode: number;
}

```

توضیح فیلدهای کلیدی

- `ticketModel` → اطلاعات بلیط رزرو شده
- `amountPay` → مبلغ نهایی قابل پرداخت
- `creditUser` → اعتبار فعلی کاربر
- `useCredit` → آیا از اعتبار استفاده شده است یا خیر

۳) مدل بلیط — Ticket

استفاده در:

- `GET /api/Profile/GetTicketInfo`
- `ReserveTicketModel.ticketModel`

```

interface Ticket {
    id: number;
    sellerId: number;
    userId?: number;
    tripId: number;
    ticketId: number;
    chairs: string;           // "1m, 2f" مثل 1
    price: number;
    totalPrice: number;
    serviceTitle?: string;
    tripDate: string;         // ISO DateTime
    payDate: string;          // ISO DateTime
    status: number;           // TicketStatus
    trackCode?: string;
    mobile?: string;
    srcCity?: string;
    destCity?: string;
}

```

وضعیت بلیط — TicketStatus

مقدار	نام	توضیح
1-	None	بدون وضعیت
0	Temp	رزرو موقت
1	Approved	تأیید شده / پرداخت نشده
2	Declined	انصراف
3	Error	خطا در پرداخت
4	Canceled	لغو شده
5	SentToMerchant	ارسال به درگاه
6	PaidBack	برگشت وجه

مقدار	نام	توضیح
9	Approving	در حال تأیید
10	Virtual	بلیط مجازی

۴) مدل خلاصه بلیط — TicketInfo

خروجی:

GET /api/Profile/GetUserTickets


```
interface TicketInfo {
  id: number;
  sellerId: number;
  userId: number;
  tripId: number;
  ticketId: number;
  chairs: string;
  price: number;
  totalPrice: number;
  serviceTitle: string;
  tripDate: string;
  payDate: string;
  status: TicketStatus;
  trackCode: string;
  mobile: string;
  srcCity: string;
  destCity: string;
  seller: string;
  payType: PayType | number;
}
```

نوع پرداخت — PayType

```
enum PayType {
  None = -1,
  Saman = 0,
  Credit = 1,
  Free = 2,
  Melat = 3,
  Saderat = 4,
  CreditAgency = 5,
  Pasargad = 6,
  Other = 10,
}
```

۵) درخواست پرداخت درگاه — SendRequestPaymentRequest

POST /api/Payment/SendRequestPayment

نیازمند  Authorization: Bearer {token}

```
interface SendRequestPaymentRequest {
  price: number; // مبلغ پرداخت (عدد صحیح)
}
```

در `payload` بازگردانده می‌شود:

```
interface RequestPayment {
    token: string; // توکن امن درگاه
    url: string; // آدرس درگاه
    getMethod: string;
    method: string; // معمولاً "post"
    formName: string; // نام فرم جهت submit
}
```

نمونه پاسخ موفق

```
{
  "code": "0",
  "message": "",
  "payload": {
    "token": "...",
    "url": "https://...",
    "getMethod": "...",
    "method": "post",
    "formName": "form1"
  }
}
```

جمع‌بندی کاربرد مدل‌ها در API

مدل خروجی	Endpoint	عملیات
<code>ReserveTicketModel</code>	<code>POST /api/Buy/SaleTicketsNew</code>	ثبت رزرو
<code>[]TicketInfo</code>	<code>GET /api/Profile/GetUserTickets</code>	لیست بلیط‌ها
<code>Ticket</code>	<code>GET /api/Profile/GetTicketInfo?TicketId={id}</code>	جزئیات بلیط
<code>RequestPayment</code> در <code>payload</code>	<code>POST /api/Payment/SendRequestPayment</code>	پرداخت

Comments