

# ۰.۴ مدل‌های رزرو و پرداخت (Booking, Payment)

طی، 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

:API ورودی

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 ۰.۵ (مدل‌های خرید و مسافر) تعریف شده است

## (۲) پاسخ ثبت رزرو — ReserveTicketModel

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

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 فیلد های کلیدی

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

## Ticket — مدل بلیط (۳)

استفاده در:

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

```

interface Ticket {
    id: number;
    sellerId: number;
    userId?: number;
    tripId: number;
    ticketId: number;
    chairs: string;           // 1" مثلث, 2f"
    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;
}

```

## TicketStatus — وضعیت بلیط

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

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 )
}
```

## پاسخ موفق پرداخت — RequestPayment

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>payload</code> در <code>RequestPayment</code>	<code>POST /api/Payment/ SendRequestPayment</code>	پرداخت

## Comments