

## به نام حق

راهنمای استفاده از سرویسهای فروش بلیت قطارهای مسافری

در حال حاضر داکيومنت برای فدک کامل است دراینده

رجا و سفیر نیز توضیح داده خواهد شد

(version : 1.9.3)

کلمات کلیدی:

1. Gender: هر کوپه به یک گروه جنسیتی (1: ویژه برادران، 2: ویژه خواهران و 3: خانواده (تخصیص یافته است که کاربر بر اساس جنسیت مسافری یکی از آنها را باید انتخاب کند).
2. Tariff: تعرفه (مقداری است طبق جدول ذیل که نوع مسافر و نوع محاسبه قیمت بلیت وی را تعیین می کند).

کد	عنوان
1	بزرگسال (قیمت کامل)
2	خردسال
3	خانواده شهدا
4	جانباز
5	کسر کوپه
6	طفل

3. تعرفه کسر کوپه: بطور مثال در صورتی که کاربر تقاضای 1 بلیت با تعرفه بزرگسال از یک کوپه 4 نفره را به همراه کوپه دربست داشته باشد، باید 4 بلیت خریداری کند که شامل 1 بلیت با تعرفه قیمتی بزرگسال و 3 بلیت با تعرفه قیمتی کسر کوپه می باشد. در واقع در این مثال هزینه دربست کردن کوپه از جمع مبالغ 3 بلیت با تعرفه کسر کوپه بدست می آید.

فرایند فروش بلیت:

1. ابتدا با استفاده از سرویس GetWagonAvailableSeatCount موجودی قطارها را دریافت کرده و به کاربر نمایش دهید تا وی قطار مورد نظر خود را انتخاب نماید. خروجی این سرویس آرایه ای از اطلاعات قطارها می باشد. در مورد این مرحله به موارد ذیل توجه نمایید:
  - a. با توجه به آنکه امکان سفر خردسالان و یا اطفال به تنهایی وجود ندارد، تعداد بزرگسال یا جانباز باید حداقل 1 باشد.
  - b. همزمان امکان دربست کردن کوپه و گرفتن بلیت برای اطفال وجود ندارد.

2. در این مرحله با استفاده از سرویس LockSeatBulk صندلی های مورد نظر کاربر را موقتا رزرو (Lock) نمایید .در مورد این مرحله به موارد ذیل توجه نمایید:

- a. طبق قوانین شرکت رجا، این سرویس در هر فراخوانی حداکثر به اندازه 16 صندلی امکان رزرو موقت وجود دارد
- b. مقدار پارامتر SelectedWagon یکی از المان های آرایه دریافتی از سرویس GetWagonAvailableSeatCount است که بدون هر گونه تغییری باید ارسال گردد.
- c. هر خرید دارای یک شماره سریال یکتا با نام SaleId می باشد که با فراخوانی این متد توسط سرور ایجاد و در خروجی ارسال می شود.
- d. با توجه به آنکه خریدهای رفت و برگشت، 2 خرید مجزا می باشند، در مورد خرید های دو طرفه این سرویس باید 2 بار فراخوانی گردد که یکی برای رفت و دیگری برای برگشت می باشد.
- e. با توجه به موضوع بند فوق، جهت برقراری ارتباط سیستمی بین خرید های رفت و برگشت شناسه SellMasterId ایجاد گردیده است که هم در ورودی و هم در خروجی این سرویس وجود دارد .در مورد این شناسه به این موارد توجه نمائید:
  - i. در خرید رفت مقدار SellMasterId ورودی باید 1- ارسال شود .در این صورت سیستم یک SellMasterId جدید ایجاد کرده و در خروجی ارسال می نماید.
  - ii. در خرید های دو طرفه، SellMasterId دریافت شده از فراخوانی سرورس فوق برای خرید رفت را بعنوان ورودی به متد LockSeatBulk خرید برگشت ارسال نمایید تا سیستم بجای ایجاد یک SellMasterId جدید از آن استفاده نموده و این 2 خرید را به هم مرتبط نماید.
- f. جهت ورود اطلاعات مسافری و پرداخت وجه، صندلی های رزرو شده توسط این سرویس تا 20 دقیقه برای کاربر رزرو خواهند بود .صندلی های رزرو شده باید قبل از اتمام این زمان ثبت نهایی شوند و یا در صورت انصراف کاربر یا بروز خطا، بوسیله سرویس UnlockSeat سریعآ آزاد شوند .در غیر اینصورت بصورت اتوماتیک توسط سیستم آزاد خواهند شد .بطور مثال در صورتی که فراخوانی LockSeatBulk مربوط به خرید رفت موفق باشد اما فراخوانی LockSeat مربوط به خرید برگشت با خطا همراه باشد، خرید رفت باید بوسیله سرویس UnlockSeat آزاد شود.
- g. پارامتر SeatCount این پارامتر از جمع تعداد بلیت های درخواستی کاربر شامل بزرگسال، خردسال، طفل، جانباز و خانواده شهدا (است .توجه شود که در صورتی که کاربر درخواست کوپه دربست را دارد، مقدار این پارامتر باید با ظرفیت کوپه قطار انتخابی یکی باشد .بطور مثال در صورتی کاربر تقاضای 2 بلیت به همراه کوپه دربست از قطاری با ظرفیت کوپه 4 را دارد، مقدار این پارامتر باید 4 فرستاده شود .در خروجی سرویس GetWagonAvailableSeatCount هر قطار یک مشخصه با نام CompartmentCapacity دارد که ظرفیت هر کوپه را مشخص می نماید.
- h. در صورتی که بخواهید تمام صندلی ها در کوپه های دربست رزرو شوند پارمتر IsExclusiveCompartment را مقدار true قرار دهید . توجه نمایید تضمینی در کنار هم قرار گرفتن کوپه ها در یک سالن وجود ندارد اما تخصیص صندلی حتما داخل کوپه های دربست اتفاق می افتد . اگر این مقدار false باشد ، صندلی های رزرو شده ممکن است به صورت پراکنده در سالن ها و کوپه های مختلف قرار گیرد . با توجه به وضعیت پراکندگی مسافر در قطار ممکن است برای تقاضا با کوپه دربست امکان رزور وجود نداشته باشد اما به صورت پراکنده امکان تخصیص صندلی وجود داشته باشد.

3. سرویس SaveTicketsInfo اطلاعات مسافری را ثبت و قیمت نهایی را بر می گرداند.

- a. پارامتر Tel: شماره ای است که از کاربر جهت تماس های اضطراری دریافت می گردد .با توجه به اینکه اطلاعات خرید از طریق پیامک به این شماره اطلاع رسانی می شود باید شماره موبایل باشد.

4. سرویس RegisterTickets صندلی های رزرو (Lock) شده در مرحله قبل را ثبت نهایی و بلیت های آنها صادر می نماید. این سرویس با دریافت sellMasterId، هم بلیت های رفت و هم بلیت های برگشت (در صورت وجود (را همزمان با هم ثبت می نماید. بنابراین چه در خرید های یک طرفه و چه در خرید های دو طرفه (رفت و برگشت (این سرویس تنها 1 بار فراخوانی می شود. در مورد این مرحله به موارد ذیل توجه نمایید:

a. این سرویس جایگزین سرویس RegisterTicket شده است). توجه نمائید که سرویس جدید در انتهای نام خود حرف s را اضافه تر از سرویس قبلی دارد)

b. سرویس RegisterTicket بزودی حذف خواهد شد. بنابراین اگر قبلا از این سرویس استفاده کرده اید جهت جلوگیری از ایرادات احتمالی هر چه سریعتر آن را حذف کرده و بجای آن از سرویس RegisterTickets استفاده نمائید.

5. با استفاده از سرویس TicketReportA اطلاعات کامل بلیت را دریافت و چاپ نمائید. مقدمات اولیه جهت استفاده از سرویس ها:

1- جهت استفاده از سرویس ها ابتدا باید یک شناسه تشخیص هویت (Authentication Token) از سرویس Login دریافت گردد. سپس در http header تمامی فراخوانی ها به فرمت ذیل اضافه شود.

Authorization: bmhh شناسه تشخیص هویت

Example: Authorization: bmhh aeDfrfcvbljzkljie78hh2323nfwejhk

این شناسه تا زمانی که سرویس فراخوانی می گردد معتبر بوده و در صورت استفاده نشدن به مدت 20 دقیقه بطور خودکار اعتبار خود را از دست خواهد داد که در اینصورت باید سرویس Login مجددا فراخوانی شده و شناسه جدید دریافت گردد.

2- خروجی سرویس ها از نوع کلاس ServiceResult می باشد. ساختار این کلاس به شرح زیر میباشد:

```
class ServiceResult<T>
{
    public bool Status { get; set; } = false;
    public bool Unauthorized { get; set; } = false;
    public string Message { get; set; } = "بدون خطا";
    public T Result { get; set; }
}
```

اگر خطایی رخ داده باشد، خطای مورد نظر در فیلدهای Message قرار خواهد گرفت. در غیر اینصورت Result دارای مقدار خواهد بود که نوع آن بر اساس سرویس فراخوانی شده متفاوت خواهد بود.

سرویس ها:

1- سرویس Login

ورودی: متغیری از نوع کلاس LoginModel می باشد

خروجی: از نوع ServiceResult<String> می باشد که حاوی شناسه تشخیص هویت (Authentication Token) می باشد

## 2- سرویس Station

: GetStations

ورودی : ندارد

خروجی <<Station>List>ServiceResult : می باشد که لیستی از ایستگاه ها را بر میگرداند . توجه نمایید که طبق توضیح شماره 2 ابتدای مستند،

: GetStationById

ورودی : یک عدد integer می باشد که کد ایستگاه مورد نظر می باشد

خروجی <Station>ServiceResult : می باشد که مشخصات ایستگاه مورد نظر را بر میگرداند.

## 3- سرویس GetLastVersion

ورودی : ندارد

خروجی : از نوع <string>ServiceResult می باشد که از این سرویس برای بررسی سرویس فروش بلیت شرکت رجا استفاده میشود به این ترتیب که اگر با فراخوانی این سرویس exception رخ داد یعنی سرورهای رجا با مشکل مواجه بوده و قادر به سرویس دهی نیستند در غیر اینصورت ادامه کار بدون مشکل خواهد بود.

## 4- سرویس GetWagonAvailableSeatCount

ورودی : متغیری از نوع کلاس GetWagonAvailableSeatCountParams می باشد.

مقادیر مربوط به Gender بصورت ذیل می باشد:

"1:ویژه برادران"

"2:ویژه خواهران"

"3:خانواده"

خروجی : از نوع <GetWagonAvailableSeatCountResult>ServiceResult می باشد و فراخوانی این سرویس، لیست قطارهای مورد نظر برای اطلاعات وارد

شده را برای مسیر رفت و برگشت بر میگرداند.

به عبارتی اجرای این سرویس یک object از نوع GetWagonAvailableSeatCountResult برمیگرداند که شامل دو لیست از کلاس

WagonAvailableSeatCount میباشد:

توضیحات فیلد ها:

SelectionHint توضیحاتی است که در زمان انتخاب واگن باید به کاربر نمایش داده شود

WagonName عنوان واگن

PathCode کد مسیر

WagonType کد نوع واگن

TrainNumber شماره قطار  
ExitTime ساعت حرکت  
MoveDate تاریخ حرکت  
IsCompartment قطار کوپه ای یا سالنی  
CompartmentCapacity ظرفیت هر کوپه  
Degree درجه  
Capacity ظرفیت خالی باقیمانده از سهمیه مورد انتخاب کاربر مانند خانواده، آقایان یا خانم ها  
Cost قیمت نهایی بلیت  
CountingAll کل ظرفیت باقیمانده قطار  
FullPrice قیمت بلیت قبل از تخفیف  
TimeOfArrival زمان رسیدن به مقصد  
DisabledReason علت اینکه این قطار قابل انتخاب توسط کاربر نیست

#### 5- سرویس LockSeat

برای خرید قطار با هتل حداقل تعداد میهمان باید 2 نفر انتخاب شود. این کنترل باید در UI صورت گیرد که تعداد مسافر حداقل 2 نفر باشد در غیر اینصورت امکان ورود به مرحله بعد وجود نداشته باشد. در صورت عدم رعایت این کنترل هزینه ی هتل نفر دوم بر عهده ی سایتی که API را پیاده سازی کرده می باشد.

ورودی : متغیری از نوع کلاس LockSeatParams می باشد.

خروجی : از نوع `ServiceResult<LockSeatResult>` می باشد و اجرای موفقیت آمیز این سرویس منجر به قفل شدن تعداد صندلی مورد درخواست از قطار مورد نظر خواهد شد.

#### 6- سرویس LockSeatBulk

ورودی : متغیری از نوع کلاس LockSeatParams می باشد.

خروجی : از نوع `ServiceResult<LockSeatBulkResult>` می باشد و اجرای موفقیت آمیز این سرویس منجر به قفل شدن تعداد صندلی مورد درخواست از قطار مورد نظر خواهد شد.

#### 7- سرویس UnlockSeat

ورودی : متغیری از نوع کلاس UnlockSeatParams می باشد

خروجی : از نوع `ServiceResult<EmptyResult>` می باشد و اجرای این سرویس در حالت بدون خطا منجر به لغو صندلی های قفل شده مورد نظر برای اطلاعات وارد شده خواهد گردید.

#### 8- سرویس SaveTicketsInfo

این سرویس ضمن ذخیره اطلاعات نهایی بلیت ها، قیمت کل را محاسبه و برمی گرداند.  
ورودی : متغیری از نوع SaveTicketsInfoParams می باشد.

خروجی : از نوع `ServiceResult<int>` است.

توجه :در خصوص اتباع خارجی شماره گذرنامه بجای کد ملی باید ارسال شود .در اینصورت به ابتدای شماره گذرنامه مسافر، کاراکتر % را اضافه نموده و در فیلد nationalCode ارسال نمائید.

#### 9- سرویس RegisterTickets

ورودی : متغیری از نوع کلاس RegisterTicketParams می باشد

خروجی : از نوع `ServiceResult<EmptyResult>` می باشد و اجرای این سرویس در حالت بدون خطا منجر به درج بلیط به تعداد درخواستی برای اطلاعات وارد شده خواهد گردید و یک پیغام از طریق موبایل و همچنین یک ایمیل به خریدار ارسال خواهد شد.

#### 10- سرویس TicketReportA

ورودی : متغیری از نوع `TicketReportAParams` می باشد

خروجی : از نوع `ServiceResult<List<TicketReportAResult>>` می باشد و فراخوانی این سرویس، لیستی برمیگرداند که هر عضو آن از نوع کلاس TicketReportAResult خواهد بود.

#### 11- سرویس UserSales

این سرویس لیست خریدهای شما را بر میگرداند.

ورودی : ندارد.

خروجی : از نوع `ServiceResult<List<UserSale>>` می باشد.

#### 12- سرویس AgentCredit

این سرویس اعتبار باقیمانده آژانس کاربر را می گرداند.

ورودی : ندارد.

خروجی : از نوع `ServiceResult<long>` است.

#### 13- سرویس ServerDateTime

این سرویس زمان سرور را بر می گرداند.

### استرداد بلیت:

#### 1- سرویس RefundTicketInfo

این سرویس جهت دریافت اطلاعات استرداد قبل از انجام آن، و صرفا جهت نمایش استفاده می شود.

ورودی:

خروج <RefundTicketInfoResult>ServiceResult

مبلغ بلیت در TotalAmount ، درصد جریمه در PenaltyPercentage و مبلغ جریمه در PenaltyAmount برگردانده می شود.  
کسر مبلغ جریمه از مبلغ بلیت ، مبلغ استرداد است.  
در صورتی که RetStatus مقداری منفی یا برابر با صفر داشته باشد بلیت غیر قابل استرداد است . ممکن است بلیت وجود نداشته باشد ، قبلا استرداد شده باشد و یا قطار حرکت کرده باشد

2- سرویس RefundTicket

این سرویس استرداد بلیت را انجام می دهد.

ورودی:

RefundTicketParams

خروجی <int>ServiceResult :

عدد 1 در صورت استرداد موفق

ملاحظات:

استرداد کلیه بلیت ها ی خرید به صورت یک جا وجود ندارد و باید بلیت های خرید جداگانه استرداد گردند . از TicketSeries که از سرویس های قبلی مقدار آن در اختیار است جهت تفکیک بلیت ها استفاده می شود .

لیست زمان های قطع سرویس

از ساعت	تا ساعت
00:00	00:15
06:30	06:45
13:30	13:45
19:30	19:45