



ASSESSORADU DE SOS TRASPORTOS
ASSESSORATO DEI TRASPORTI

Evoluzione e gestione del Sistema Account Based Ticketing per il Trasporto Pubblico Locale
in Regione Autonoma della Sardegna

CUI S80002870923202500621 - CUP E71C25000020002

Allegato 2

Documentazione integrazione ARST



1. LOGIN E INFORMAZIONI RIVENDITA

Di seguito saranno riportati nel dettaglio i servizi che riguardano la login e il recupero delle informazioni relativi al touchpoint di rivendita.

- **Login:** permette di ottenere un token utile poi a richiamare i servizi successivi;
- **Get Reseller Info:** recupera l'identificativo del touchpoint in base ai dati di login;
- **Get Touchpoint Configuration:** recupera le informazioni del touchpoint;

Di seguito per ogni servizio saranno descritti i dati di input e di output con il relativo mapping.

1.1 LOGIN

1.1.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/oauth2/token

1.1.2 DATI INPUT

Attributo Modello	Tipo	Note
grant_type	String	Fisso a "password"
username	String	Per lo sviluppo da usare una mail che sarà consegnata successivamente
scope	String	Fisso a "internal_login"
password	String	Verranno inviate separatamente
CliendID	String	Verranno inviate separatamente
ClientSecret	String	Verranno inviate separatamente

1.1.3 DATI OUTPUT

Esito 200:

Attributo Modello	Tipo	Note
access_token	String	Token utile a richiamare i servizi successivi
token_type	String	Tipo del token (Bearer)
scope	String	
expires_in	String	
refresh_token	String	

1.2 RESELLER INFO

1.2.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/es/interactive-ticket-manager/v1/reseller/info

1.2.2 DATI OUTPUT

L'output legge i parametri nel token e torna i seguenti parametri:

Esito 200:

Attributo Modello	Tipo	Note
touchpointId	String	
organizationSellerId	String	

1.3 TOUCHPOINT CONFIGURATION

1.3.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/mms/common/touchpoint/configuration/api/v2/touchpoints/{{touchpointId}}

1.3.2 DATI INPUT

Nome	Tipo	Note
touchpointId	String	Da inserire nella url. Campo preso dalla GET Reseller Info.

1.3.3 DATI OUTPUT

Per l'output consultare lo swagger allegato.

2. FLUSSO DI VENDITA URBANO IMPERSONALE

Di seguito saranno riportati nel dettaglio i servizi che riguardano il flusso di vendita urbano impersonale:

- **Ricerca Prodotti:** permette la ricerca del catalogo tariffario e il relativo prezzo per ogni prodotto esclusivamente per il profilo IMPERSONALE;
- **Validità Temporale:** permette di ottenere l'intervallo temporale del prodotto scelto (IMP: questo endpoint sarà richiamato solo nel caso di prodotti di tipo abbonamenti);
- **Calcolo del prezzo:** permette di ricalcolare il prezzo nel caso vengano scelti più prodotti da acquistare/vendere;
- **Vendita:** permette di finalizzare l'acquisto sulla piattaforma del ticket scelto;
- **Creazione QRCode:** permette di creare un'immagine criptata del QRCode.

Di seguito per ogni servizio saranno descritti i dati di input e di output con il relativo mapping.

2.1 RICERCA PRODOTTI URBANI

2.1.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/arst/productmanager/v1/Product/FindProducts

2.1.2 DATI INPUT

In questo documento saranno riportati solo alcuni parametri di input, descritti di seguito:

Nome	Tipo	Note
pageNumber	Number	
pageSize	Number	
where.profiles	String Array	Array di stringhe con il valore fisso "CP.BS"
where. profileOp	String	String fissa a "OR"
where.mediaTypes	String Array	Array con il valore fisso "VT"
where.zoneld	String	Sarà fisso in base alla zona di ricerca
touchpointId	String	Da inserire l'identificativo del touchpoint recuperabile al punto 1.2
touchpointType	String	Fisso a "POS"
areaType	String	Fisso a "URBAN"

Invece, gli altri parametri sono descritti nello swagger allegato.

2.1.3 DATI OUTPUT

Per l'output consultare lo swagger allegato.

Alcuni dei parametri sono i seguenti:

Attributo Modello	Tipo	Note
items.name	String	
items.unitaryPrice.amount	Number	È il prezzo del titolo di viaggio per quantità pari a 1
items.id	String	
items.version	Number	

In caso di abbonamento bisogna invocare l'endpoint descritto al 2.2

2.2 VALIDITA' TEMPORALE

Questo endpoint verrà richiamato solo nel caso in cui un titolo scelto dall'utente sarà di tipo `productType = 11` e `usageParameters.validityPeriods.trigger.usageTriggerType != "V"`.

2.2.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/ilm/arst/productmanager/v1/Product/Validity?productId={{productid}}&version={{version}}&maxOption={{maxOption}}&date={{date}}

2.2.2 DATI INPUT

Nome	Tipo	Note
productId	String	Inseriti all'interno della URL e recuperato dal prodotto scelto al punto 2.1
version	String	Inseriti all'interno della URL e recuperato dal prodotto scelto al punto 2.1
maxOption	Number	Inseriti all'interno della URL (Fisso a 1)
date	String	Inseriti all'interno della URL (Data di possibile acquisto in formato yyyy-mm-dd)

2.2.3 DATI OUTPUT

L'output è un array che in questo caso prevede solo un elemento:

Attributo Modello	Tipo	Note
from	String	Data di inizio validità del titolo
to	String	Data di fine validità del titolo

2.3 CALCOLO DEL PREZZO

Questo endpoint verrà invocato nel caso in cui si vogliano scegliere quantità del titolo superiori a 1.

2.3.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/es/interactive-ticket-manager/fareProductPrice

2.3.2 DATI INPUT

Nome	Tipo	Note
items	Array	
itemType	String	Fisso a "PRODUCT" se si tratta di un titolo di viaggio
itemId	String	Da recuperare dal campo di output del 2.1 items[i].id dove con i si intende l'i-esimo prodotto
itemVersion	Number	Da recuperare dal campo di output del 2.1 items[i].version dove con i si intende l'i-esimo prodotto
date	String	Data di acquisto in formato UTC (es. 2023-06-19T14:30:49.718Z)
fareTableCellId	String	Da recuperare dal campo di output del 2.1 items[i].fareTableCells[0].cellId dove con i si intende l'i-esimo prodotto
quantity	Number	Numero di quantità del prodotto scelto

2.3.3 DATI OUTPUT

In output avremo una lista dei seguenti parametri:

Attributo Modello	Tipo	Note
totalAmount	String	È il totale dei titoli di viaggio da acquistare
ccy	String	"E" indica l'euro
items	Array	
itemId	String	
itemName	String	
itemVersion	Integer	
fareTableCellId	String	
quantity	Integer	
ccy	String	



amount	String	
--------	--------	--

2.4 VENDITA PRODOTTI URBANI IMPERSONALI

2.4.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/arst/purchasemanager/v1/Purchase/purchase?channelType=RESELLER&codeMediaType=VT

2.4.2 DATI INPUT

Il body in input è di tipo PurchaseCart (da vedere nel file yaml allegato) e un esempio lo si può trovare nella collection allegata.

Nome	Tipo	Note
channelType	String	Fisso a "POS"
codeMediaType	String	Fisso a "VT"

2.4.3 DATI OUTPUT

In output avremo:

Attributo Modello	Tipo	Note
purchaseTransactionId	String	
ticketIdList	Array	Array di stringhe che identificano i ticketId

2.5 CREAZIONE QR CODE

2.5.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/mediamanager/v1/Barcode/image?TicketId={{ticketId}} &BarCodePrefix=ARST1&EncryptedOutput=true

2.5.2 DATI INPUT

Nome	Tipo	Note
TicketId	String	Dal servizio precedente in riferimento al ticket al quale si vuole creare il qrcode
BarCodePrefix	String	Fisso a "ARST1"
EncryptedOutput	Boolean	Fisso a true

2.5.3 DATI OUTPUT

In output si ottiene l'immagine di un qrcode.

3. FLUSSO DI VENDITA EXTRAURBANO IMPERSONALE

Di seguito saranno riportati nel dettaglio i servizi che riguardano il flusso di vendita extraurbano anonimo:

- **Ricerca Origine:** permette di ricercare la fermata di origine;
- **Ricerca Destinazione:** permette di ricercare la fermata di destinazione partendo da una di origine;
- **Ricerca Prodotti:** permette la ricerca del catalogo tariffario e il relativo prezzo per ogni prodotto esclusivamente per il profilo IMPERSONALE;
- **Validità Temporale:** permette di ottenere l'intervallo temporale del prodotto scelto (IMP: questo endpoint sarà richiamato solo nel caso di prodotti di tipo abbonamenti);
- **Calcolo del prezzo:** permette di ricalcolare il prezzo nel caso vengano scelti più prodotti da acquistare/vendere;
- **Vendita:** permette di finalizzare l'acquisto sulla piattaforma del ticket scelto;
- **Creazione QRCode:** permette di creare un'immagine criptata del qrcode.

Di seguito per ogni servizio saranno descritti i dati di input e di output con il relativo mapping.

3.1 RICERCA ORIGINE

3.1.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/es/interactive-ticket-manager/extendedDistanceMatrix/location?

3.1.2 DATI INPUT

Nome	Tipo	Note
pageNumber	Number	
pageSize	Number	
locationType	String array	Fisso a ORIGIN
locationName	String	Serve per filtrare le località inserendo le iniziali della località di origine. OPZIONALE

3.1.3 DATI OUTPUT

In caso di esito positivo bisognerà riprendere la locationRef per effettuare poi la chiamata successiva (GET DESTINATION).

Attributo Modello	Tipo	Note
locationRef	String	Identificativo che viene utilizzato come input per ottenere le destination.
locationName	String	

3.2 RICERCA DESTINAZIONE

3.2.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/es/interactive-ticket-manager/extendedDistanceMatrix/location?

3.2.2 DATI INPUT

Nome	Tipo	Note
pageNumber	Integer	
pageSize	Integer	
locationType	String	Fisso a DESTINATION
comboLocationRef	String	Inserire il campo locationRef ottenuto dall'output di GetOrigin.
locationName	String	Serve per filtrare le località inserendo le iniziali della località di destinazione. OPZIONALE

3.2.3 DATI OUTPUT

In caso di esito positivo bisognerà riprendere la locationRef per effettuare poi la chiamata successiva (FIND PRODUCTS).

Attributo Modello	Tipo	Note
locationRef	String	Identificativo che viene utilizzato come input per la Find Products.
locationName	String	

3.3 RICERCA PRODOTTI EXTRAURBANI

3.3.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/arst/productmanager/v1/Product/FindProducts

3.3.2 DATI INPUT

In questo documento saranno riportati solo alcuni parametri di input, descritti di seguito:

Nome	Tipo	Note
pageNumber	Number	
pageSize	Number	
where.profiles	String array	Array di stringhe con il valore fisso "CP.BS"
where.profileOp	String	String fissa a "OR"
where.mediaTypes	String	Array con il valore fisso "VT"
where.origin	String	LocationRef ricavato nell'output di GetOrigin
where.destination	String	LocationRef ricavato nell'output di GetDestination
where.areaType		Fisso a "SUBURBAN"
where.touchpointId	String	Da inserire l'identificativo del touchpoint recuperabile al punto 1.2
where.touchpointType	String	Fisso a "POS"

Invece, gli altri parametri sono descritti nello swagger allegato.

3.3.3 DATI OUTPUT

Per l'output consultare lo swagger allegato.

Alcuni dei parametri sono i seguenti:

Attributo Modello	Tipo	Note
items.name	String	
items.unitaryPrice.amount	Number	È il prezzo del titolo di viaggio per quantità pari a 1
items.id	String	
items.version	Number	

In caso di quantità superiori a uno è necessario invocare l'endpoint al punto 3.5.

3.4 VALIDITA' TEMPORALE

Questo endpoint verrà richiamato solo nel caso in cui un titolo scelto dall'utente sarà di tipo `productType = 11` e `usageParameters.validityPeriods.trigger.usageTriggerType != "V"`.

3.4.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/ilm/arst/productmanager/v1/Product/Validity?productId={{productid}}&version={{version}}&maxOption={{maxOption}}&date={{date}}

3.4.2 DATI INPUT

Nome	Tipo	Note
productId	String	Inseriti all'interno della URL e recuperato dal prodotto scelto al punto 3.3
version	String	Inseriti all'interno della URL e recuperato dal prodotto scelto al punto 3.3
maxOption	Number	Inseriti all'interno della URL (Fisso a 1)
date	String	Inseriti all'interno della URL (Data di possibile acquisto in formato yyyy-mm-dd)

3.4.3 DATI OUTPUT

L'output è un array che in questo caso prevede solo un elemento:

Attributo Modello	Tipo	Note
from	String	Data di inizio validità del titolo
to	String	Data di fine validità del titolo

3.5 CALCOLO DEL PREZZO

Questo endpoint verrà invocato nel caso in cui si vogliano scegliere quantità del titolo superiori a 1.

3.5.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/es/interactive-ticket-manager/fareProductPrice

3.5.2 DATI INPUT

Nome	Tipo	Note
items	Array	
itemType	String	Fisso a "PRODUCT" se si tratta di un titolo di viaggio
itemId	String	Da recuperare dal campo di output del 3.3 items[i].id dove con i si intende l'i-esimo prodotto
itemVersion	Number	Da recuperare dal campo di output del 3.3.3 items[i].version dove con i si intende l'i-esimo prodotto
date	String	Data di acquisto in formato UTC (es. 2023-06-19T14:30:49.718Z)
fareTableCellId	String	Da recuperare dal campo di output del 2.1 items[i].fareTableCells[0].cellId dove con i si intende l'i-esimo prodotto
quantity	Number	Numero di quantità del prodotto scelto

3.5.3 DATI OUTPUT

In output avremo una lista dei seguenti parametri:

Attributo Modello	Tipo	Note
totalAmount	String	È il totale dei titoli di viaggio da acquistare
ccy	String	"E" indica l'euro
items	Array	
itemId	String	
itemName	String	
itemVersion	Integer	
fareTableCellId	String	
quantity	Integer	
ccy	String	



amount	String	
--------	--------	--

3.6 VENDITA PRODOTTI EXTRAURBANI IMPERSONALI

3.6.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/arst/purchasemanager/v1/Purchase/purchase?channelType=RESELLER&codeMediaType=VT

3.6.2 DATI INPUT

Il body in input è di tipo PurchaseCart (da vedere nel file yaml allegato)

Nome	Tipo	Note
channelType	String	Fisso a "POS"
codeMediaType	String	Fisso a "VT"

Molto importante è inserire le informazioni nella parte della "geographicalIntervallLimit".

3.6.3 DATI OUTPUT

In output avremo:

Attributo Modello	Tipo	Note
purchaseTransactionId	String	
ticketIdList	Array	Array di stringhe che identificano i ticketId

3.7 CREAZIONE QR CODE

3.7.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/mediamanager/v1/Barcode/image?TicketId={{ticketId}} &BarCodePrefix=ARST1&EncryptedOutput=true

3.7.2 DATI INPUT

Nome	Tipo	Note
TicketId	String	Dal servizio precedente in riferimento al ticket al quale si vuole creare il qrcode
BarCodePrefix	String	Fisso a "ARST1"
EncryptedOutput	Boolean	Fisso a true

3.7.3 DATI OUTPUT

In output si ottiene l'immagine di un qrcode.

4. FLUSSO DI VENDITA URBANO CON CARD ARST

Di seguito saranno riportati nel dettaglio i servizi che riguardano il flusso di vendita urbano nominativo:

- **Ricerca Prodotti:** permette la ricerca del catalogo tariffario e il relativo prezzo per ogni prodotto esclusivamente per il profilo con CARD ARST;
- **Validità Temporale:** permette di ottenere l'intervallo temporale del prodotto scelto (IMP: questo endpoint sarà richiamato solo nel caso di prodotti di tipo abbonamenti);
- **Calcolo del prezzo:** permette di ricalcolare il prezzo nel caso vengano scelti più prodotti da acquistare/vendere;
- **Calcolo del prezzo con Card ARST:** permette di ricalcolare il prezzo nel caso venga inserito l'identificativo della card ARST;
- **Vendita:** permette di finalizzare l'acquisto sulla piattaforma del ticket scelto;
- **Creazione QRCode:** permette di creare un'immagine criptata del QRCode.

Di seguito per ogni servizio saranno descritti i dati di input e di output con il relativo mapping.

4.1 RICERCA PRODOTTI

4.1.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/arst/productmanager/v1/Product/FindProducts

4.1.2 DATI INPUT

In questo documento saranno riportati solo alcuni parametri di input, descritti di seguito:

Nome	Tipo	Note
pageNumber	Number	
pageSize	Number	
where.profiles	String Array	Array di stringhe con i valori fissi "CP.BS", "CP_00001"
where. profileOp	String	String fissa a "OR"
where.mediaTypes	String Array	Array con il valore fisso "VT"
where.zoneld	String	Sarà fisso in base alla zona di ricerca
touchpointId	String	Da inserire l'identificativo del touchpoint recuperabile al punto 1.2
touchpointType	String	Fisso a "POS"
areaType	String	Fisso a "URBAN"

Invece, gli altri parametri sono descritti nello swagger allegato.

4.1.3 DATI OUTPUT

Per l'output consultare lo swagger allegato.

Alcuni dei parametri sono i seguenti:

Attributo Modello	Tipo	Note
items.name	String	
items.unitaryPrice.amount	Number	È il prezzo del titolo di viaggio per quantità pari a 1
items.id	String	
items.version	Number	
items.cardArstProfileExist	Boolean	Questo parametro permette di capire se il prodotto è dedicato ai possessori di CARDARST (in caso il valore sia true)

4.2 VALIDITA' TEMPORALE

Questo endpoint verrà richiamato solo nel caso in cui un titolo scelto dall'utente sarà di tipo `productType = 11` e `usageParameters.validityPeriods.trigger.usageTriggerType != "V"`.

4.2.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/ilm/arst/productmanager/v1/Product/Validity?productId={{productid}}&version={{version}}&maxOption={{maxOption}}&date={{date}}

4.2.2 DATI INPUT

Nome	Tipo	Note
productId	String	Inseriti all'interno della URL e recuperato dal prodotto scelto al punto 4.1
version	String	Inseriti all'interno della URL e recuperato dal prodotto scelto al punto 4.1
maxOption	Number	Inseriti all'interno della URL (Fisso a 1)
date	String	Inseriti all'interno della URL (Data di possibile acquisto in formato yyyy-mm-dd)

4.2.3 DATI OUTPUT

L'output è un array che in questo caso prevede solo un elemento:

Attributo Modello	Tipo	Note
from	String	Data di inizio validità del titolo
to	String	Data di fine validità del titolo

4.3 CALCOLO DEL PREZZO

Questo endpoint verrà invocato nel caso in cui si vogliano scegliere quantità del titolo superiori a 1.

4.3.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/es/interactive-ticket-manager/fareProductPrice

4.3.2 DATI INPUT

Nome	Tipo	Note
items	Array	
itemType	String	Fisso a "PRODUCT" se si tratta di un titolo di viaggio
itemId	String	Da recuperare dal campo di output del 4.1 items[i].id dove con i si intende l'i-esimo prodotto
itemVersion	Number	Da recuperare dal campo di output del 4.1 items[i].version dove con i si intende l'i-esimo prodotto
date	String	Data di acquisto in formato UTC (es. 2023-06-19T14:30:49.718Z)
fareTableCellId	String	Da recuperare dal campo di output del 4.1 items[i].fareTableCells[0].cellId dove con i si intende l'i-esimo prodotto
quantity	Number	Numero di quantità del prodotto scelto

4.3.3 DATI OUTPUT

In output avremo una lista dei seguenti parametri:

Attributo Modello	Tipo	Note
totalAmount	String	È il totale dei titoli di viaggio da acquistare
ccy	String	"E" indica l'euro
items	Array	
itemId	String	
itemName	String	
itemVersion	Integer	
fareTableCellId	String	
quantity	Integer	
ccy	String	



amount	String	
--------	--------	--

4.4 PREZZO PRODOTTO SELEZIONATO PER ID CARD ARST

4.4.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/arst/productmanager/v1/Product/FindProducts

4.4.2 DATI INPUT

In questo documento saranno riportati solo alcuni parametri di input, descritti di seguito:

Nome	Tipo	Note
pageNumber	Number	
pageSize	Number	
where.profiles	String Array	Array di stringhe con i valori fissi "CP.BS", "CP_00001"
where. profileOp	String	String fissa a "OR"
where.mediaTypes	String Array	Array con il valore fisso "VT"
where.zoneld	String	Sarà fisso in base alla zona di ricerca
touchpointId	String	Da inserire l'identificativo del touchpoint recuperabile al punto 1.2
touchpointType	String	Fisso a "POS"
areaType	String	Fisso a "URBAN"
cardId	String	
productId	String	

Invece, gli altri parametri sono descritti nello swagger allegato.

4.4.3 DATI OUTPUT

Per l'output consultare lo swagger allegato e l'esempio riportato nella collection Postman allegata. Alcuni dei parametri sono i seguenti:

Attributo Modello	Tipo	Note
items.name	String	
items.promotionsPrice.amount	Number	È il prezzo del titolo di viaggio per il prodotto scontato in base all'identificativo inserito
items.promotionsPrice.promotionValue	Number	Indica la promozione applicata in termini percentuali

4.5 VENDITA PRODOTTI URBANI CON CARD ARST

4.5.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/arst/purchasemanager/v1/Purchase/purchase?channelType=RESELLER&codeMediaType=VT

4.5.2 DATI INPUT

Il body in input è di tipo PurchaseCart (da vedere nel file yaml allegato) e un esempio lo si può trovare nella collection allegata.

Nome	Tipo	Note
channelType	String	Fisso a "POS"
codeMediaType	String	Fisso a "VT"
cardId	String	Identificativo della Card ARST

4.5.3 DATI OUTPUT

In output avremo:

Attributo Modello	Tipo	Note
purchaseTransactionId	String	
ticketIdList	Array	Array di stringhe che identificano i ticketId

4.6 CREAZIONE QR CODE

4.6.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/mediamanager/v1/Barcode/image?TicketId={{ticketId}} &BarCodePrefix=ARST1&EncryptedOutput=true

4.6.2 DATI INPUT

Nome	Tipo	Note
TicketId	String	Dal servizio precedente in riferimento al ticket al quale si vuole creare il qrcode
BarCodePrefix	String	Fisso a "ARST1"
EncryptedOutput	Boolean	Fisso a true

4.6.3 DATI OUTPUT

In output si ottiene l'immagine di un qrcode.

5. FLUSSO DI VENDITA EXTRAURBANO CON CARD ARST

Di seguito saranno riportati nel dettaglio i servizi che riguardano il flusso di vendita extraurbano nominativo:

- **Ricerca Origine:** permette di ricercare la fermata di origine;
- **Ricerca Destinazione:** permette di ricercare la fermata di destinazione partendo da una di origine;
- **Ricerca Prodotti:** permette la ricerca del catalogo tariffario e il relativo prezzo per ogni prodotto esclusivamente per il profilo con CARD ARST;
- **Validità Temporale:** permette di ottenere l'intervallo temporale del prodotto scelto (IMP: questo endpoint sarà richiamato solo nel caso di prodotti di tipo abbonamenti);
- **Calcolo del prezzo:** permette di ricalcolare il prezzo nel caso vengano scelti più prodotti da acquistare/vendere;
- **Calcolo del prezzo con Card ARST:** permette di ricalcolare il prezzo nel caso venga inserito l'identificativo della card ARST;
- **Vendita:** permette di finalizzare l'acquisto sulla piattaforma del ticket scelto;
- **Creazione QRCode:** permette di creare un'immagine criptata del QRCode.

Di seguito per ogni servizio saranno descritti i dati di input e di output con il relativo mapping.

5.1 RICERCA ORIGINE

5.1.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/es/interactive-ticket-manager/extendedDistanceMatrix/location?

5.1.2 DATI INPUT

Nome	Tipo	Note
pageNumber	Number	
pageSize	Number	
locationType	String array	Fisso a ORIGIN
locationName	String	Serve per filtrare le località inserendo le iniziali della località di origine. OPZIONALE

5.1.3 DATI OUTPUT

In caso di esito positivo bisognerà riprendere la locationRef per effettuare poi la chiamata successiva (GET DESTINATION).

Attributo Modello	Tipo	Note
locationRef	String	Identificativo che viene utilizzato come input per ottenere le destination.
locationName	String	

5.2 RICERCA DESTINAZIONE

5.2.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/es/interactive-ticket-manager/extendedDistanceMatrix/location?

5.2.2 DATI INPUT

Nome	Tipo	Note
pageNumber	Integer	
pageSize	Integer	
locationType	String	Fisso a DESTINATION
comboLocationRef	String	Inserire il campo locationRef ottenuto dall'output di GetOrigin.
locationName	String	Serve per filtrare le località inserendo le iniziali della località di destinazione. OPZIONALE

5.2.3 DATI OUTPUT

In caso di esito positivo bisognerà riprendere la locationRef per effettuare poi la chiamata successiva (FIND PRODUCTS).

Attributo Modello	Tipo	Note
locationRef	String	Identificativo che viene utilizzato come input per la Find Products.
locationName	String	

5.3 RICERCA PRODOTTI EXTRAURBANI

5.3.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/arst/productmanager/v1/Product/FindProducts

5.3.2 DATI INPUT

In questo documento saranno riportati solo alcuni parametri di input, descritti di seguito:

Nome	Tipo	Note
pageNumber	Number	
pageSize	Number	
where.profiles	String array	Array di stringhe con i valori fissi "CP.BS", "CP_00001"
where.mediaTypes	String	Array con il valore fisso VT
origin	String	LocationRef ricavato nell'output di GetOrigin
destination	String	LocationRef ricavato nell'output di GetDestination
areatype		Fisso a "SUBURBAN"
touchpointId	String	Da inserire l'identificativo del touchpoint recuperabile al punto 1.2
touchpointType	String	Fisso a "POS"

Invece, gli altri parametri sono descritti nello swagger allegato.

5.3.3 DATI OUTPUT

Per l'output consultare lo swagger e la collection Postman in allegato.

Alcuni dei parametri sono i seguenti:

Attributo Modello	Tipo	Note
items.name	String	
items.unitaryPrice.amount	Number	È il prezzo del titolo di viaggio per quantità pari a 1
items.id	String	
items.version	Number	
items.cardArstProfileExist	Boolean	Questo parametro permette di capire se il prodotto è dedicato ai possessori di CARDARST (in caso il valore sia true)

Attributo Modello	Tipo	Note
-------------------	------	------



items.moovaProduct.productName	String	
items.unitaryPrice.amount	Number	
Items.cardArstProfileExist	Boolean	In caso di valore a true il prodotto è vendibile con la scontistica associata all'identificativo della CARDARST

In caso di quantità superiori a uno è necessario invocare l'endpoint al punto 5.5.

5.4 VALIDITA' TEMPORALE

Questo endpoint verrà richiamato solo nel caso in cui un titolo scelto dall'utente sarà di tipo `productType = 11` e `usageParameters.validityPeriods.trigger.usageTriggerType != "V"`.

5.4.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/ilm/arst/productmanager/v1/Product/Validity?productId={{productid}}&version={{version}}&maxOption={{maxOption}}&date={{date}}

5.4.2 DATI INPUT

Nome	Tipo	Note
productId	String	Inseriti all'interno della URL e recuperato dal prodotto scelto al punto 5.1
version	String	Inseriti all'interno della URL e recuperato dal prodotto scelto al punto 5.1
maxOption	Number	Inseriti all'interno della URL (Fisso a 1)
date	String	Inseriti all'interno della URL (Data di possibile acquisto in formato yyyy-mm-dd)

5.4.3 DATI OUTPUT

L'output è un array che in questo caso prevede solo un elemento:

Attributo Modello	Tipo	Note
from	String	Data di inizio validità del titolo
to	String	Data di fine validità del titolo

5.5 CALCOLO DEL PREZZO

Questo endpoint verrà invocato nel caso in cui si vogliano scegliere quantità del titolo superiori a 1.

5.5.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/es/interactive-ticket-manager/fareProductPrice

5.5.2 DATI INPUT

Nome	Tipo	Note
items	Array	
itemType	String	Fisso a "PRODUCT" se si tratta di un titolo di viaggio
itemId	String	Da recuperare dal campo di output del 5.1 items[i].id dove con i si intende l'i-esimo prodotto
itemVersion	Number	Da recuperare dal campo di output del 5.1 items[i].version dove con i si intende l'i-esimo prodotto
date	String	Data di acquisto in formato UTC (es. 2023-06-19T14:30:49.718Z)
fareTableCellId	String	Da recuperare dal campo di output del 5.1 items[i].fareTableCells[0].cellId dove con i si intende l'i-esimo prodotto
quantity	Number	Numero di quantità del prodotto scelto

5.5.3 DATI OUTPUT

In output avremo una lista dei seguenti parametri:

Attributo Modello	Tipo	Note
totalAmount	String	È il totale dei titoli di viaggio da acquistare
ccy	String	"E" indica l'euro
items	Array	
itemId	String	
itemName	String	
itemVersion	Integer	
fareTableCellId	String	
quantity	Integer	
ccy	String	



amount	String	
--------	--------	--

5.6 PREZZO PRODOTTO SELEZIONATO PER ID CARD ARST

5.6.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/arst/productmanager/v1/Product/FindProducts

5.6.2 DATI INPUT

In questo documento saranno riportati solo alcuni parametri di input, descritti di seguito:

Nome	Tipo	Note
pageNumber	Number	
pageSize	Number	
where.profiles	String Array	Array di stringhe con i valori fissi "CP.BS", "CP_00001"
where. profileOp	String	String fissa a "OR"
where.mediaTypes	String Array	Array con il valore fisso "VT"
where.zoneld	String	Sarà fisso in base alla zona di ricerca
touchpointId	String	Da inserire l'identificativo del touchpoint recuperabile al punto 1.2
touchpointType	String	Fisso a "POS"
areaType	String	Fisso a "URBAN"
cardId	String	
productId	String	

Invece, gli altri parametri sono descritti nello swagger allegato.

5.6.3 DATI OUTPUT

Per l'output consultare lo swagger allegato e l'esempio riportato nella collection Postman allegata. Alcuni dei parametri sono i seguenti:

Attributo Modello	Tipo	Note
items.name	String	
items.promotionsPrice.amount	Number	È il prezzo del titolo di viaggio per il prodotto scontato in base all'identificativo inserito
items.promotionsPrice.promotionValue	Number	Indica la promozione applicata in termini percentuali

5.7 VENDITA PRODOTTI EXTRAURBANI CON CARD ARST

5.7.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/arst/purchasemanager/v1/Purchase/purchase?channelType=RESELLER&codeMediaType=VT

5.7.2 DATI INPUT

Il body in input è di tipo PurchaseCart (da vedere nel file yaml allegato) e un esempio lo si può trovare nella collection allegata.

Nome	Tipo	Note
channelType	String	Fisso a "POS"
codeMediaType	String	Fisso a "VT"
cardId	String	Identificativo della Card ARST

Molto importante è inserire le informazioni nella parte della "geographicalIntervalLimit".

5.7.3 DATI OUTPUT

In output avremo:

Attributo Modello	Tipo	Note
purchaseTransactionId	String	
ticketIdList	Array	Array di stringhe che identificano i ticketId

5.8 CREAZIONE QR CODE

5.8.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/mediamanager/v1/Barcode/image?TicketId={{ticketId}} &BarCodePrefix=ARST1&EncryptedOutput=true

5.8.2 DATI INPUT

Nome	Tipo	Note
TicketId	String	Dal servizio precedente in riferimento al ticket al quale si vuole creare il qrcode
BarCodePrefix	String	Fisso a "ARST1"
EncryptedOutput	Boolean	Fisso a true

5.8.3 DATI OUTPUT

In output si ottiene l'immagine di un qrcode.

6. CONTROLLO UNIVOCITA' TITOLO

Il seguente endpoint dovrà eseguire il controllo dell'univocità del titolo e bloccare un possibile acquisto già effettuato in precedenza dallo stesso utente.

- **Controllo Univocità Titolo:** permette di controllare se all'interno del wallet dell'utente finale (riconoscibile per identificativo della CARD ARST) non esista quel titolo che si sta per acquistare per intervallo di date, origine e destinazione.

6.1 CONTROLLO UNIVOCITA' TITOLO

6.1.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/ilm/Whitelist/v1/Whitelist/Check

6.1.2 DATI INPUT

Di seguito un esempio di JSON da inviare:

```
{
  "campaigncode": "ARST_CHECKTDV",
  "filters": [
    {
      "name": "$.exCode",
      "value": "XXXXXXXXXXXXXXXXXX"
    },
    {
      "name": "$.cardId",
      "value": "XXXXXXXXXXXXXXXXXX"
    },
    {
      "name": "$.datefrom",
      "value": "XXXXXXXXXXXXXXXXXX"
    },
    {
      "name": "$.dateto",
      "value": "XXXXXXXXXXXXXXXXXX"
    },
    {
      "name": "$.OD[*]",
      "value": "XXXXXXXXXXXXXXXXXX"
    },
    {
      "name": "$.OD[*]",
      "value": "XXXXXXXXXXXXXXXXXX"
    }
  ]
}
```

I valori nella proprietà **value** saranno tutti da inserire in base all'acquisto che si vuole acquistare.



6.1.3 DATI OUTPUT

Il cliente non potrà procedere all'acquisto se il codice di response è uguale a 409.

6.2 SEARCH SALES PACKAGES

I carnet sono dei salespackages per l'ambiente Moova.

6.2.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/es/interactive-ticket-manager/salespackages

6.2.2 DATI INPUT

Nome	Tipo	Note
pageNumber	Integer	
pageSize	Integer	
spkTp	String	Fisso a 'SP3'
tpTyp	String	Fisso a 'POS'
medTp	String	Fisso a 'VT'
cpId	String	Fisso a 'CP.BS'
cpIdOp	String	Fisso a 'OR'

6.2.3 DATI OUTPUT

Esito 200:

In output avremo una lista dei sales packages da poter acquistare.

6.3 SEARCH SALES PACKAGES EXTRA URBANO

Una volta ricavata l'origine e la destinazione bisogna prendere il chilometraggio e passarlo nella search

6.3.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/es/interactive-ticket-manager/salespackages

Nome	Tipo	Note
pageNumber	Integer	1
pageSize	Integer	100
serMd	String	07.04 Extra urbano 01.00 Ferroviario
geoEn	String	
geoSt	string	
spkTp	String	Fisso a 'SP3'
tpTyp	String	Fisso a 'POS'
medTp	String	Fisso a 'VT'
cpId	String	Fisso a 'CP.BS'
cpIdOp	String	Fisso a 'OR'

6.4 GET TEMPORAL VALIDITY SALES PACKAGES

Una volta effettuata la ricerca dei pacchetti tariffari, viene richiamato questo servizio solo nel caso di usageTriggerType diverso da "V" della validityPeriods nella search sales packages (campo usageParameters.validityPeriods[i].trigger.usageTriggerType nell'oggetto in output alla chiamata precedente)

6.4.1 ENDPOINT

Endpoint	
Http Method	GET
Uri	/es/interactive-ticket-manager/salespackages/salespackages/{salespackageId}/{version}/validity

6.4.2 DATI INPUT

Nome	Tipo	Note
salespackageId (pathParam)	String	Recuperabile dalla response della search sales packages (campo salespackageId)
version (pathParam)	integer	Recuperabile dalla response della search search sales packages (campo salesPackageVersion)
maxOption	integer	Fisso a 1 (dato che siamo in fase vendita anonima)
date	String	Sysdate (data dal canale) in formato yyyy-MM-dd

6.4.3 DATI OUTPUT

Esito 200:

In output avremo un array di:

Attributo Modello	Tipo	Note
from	String	
to	String	

Avendo inserito in input maxOption = 1 avremo solo un element nell'array.

6.5 CALCULATE PRICE SALES PACKAGES

La richiesta del calcolo del prezzo avviene invocando il seguente endpoint:

6.5.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/es/interactive-ticket-manager/fareProductPrice

6.5.2 DATI INPUT

Di seguito un esempio di request body:

```
{
  "items": [
    {
      "itemType": "SALESPACKAGE",
      "itemId": "b158370c-cfdb-11ee-995d-693aa062aae6",
      "itemVersion": 1,
      "date": "2024-01-24T11:08:49.718Z",
      "quantity": 1
    }
  ]
}
```

Nome	Tipo	Note
itemType	String	Fisso a "SALESPACKAGE"
itemId	String	Recuperabile dalla response della search sales packages (campo salesPackageId)
itemVersion	String	Recuperabile dalla response della search sales packages (campo salesPackageVersion)
quantity	String	Quantità selezionata
date	String	Sysdate in formato ISO (es.2024-01-24T11:08:49.718Z)

6.5.3 DATI OUTPUT

Esito 200:

In output avremo un json di questo tipo:

```
{
  "totalAmount": 13.0,
  "ccy": "E",

```



```
"items": [  
  {  
    "itemId": "692f1537-d0c2-11ee-bb83-b31e97cb5627",  
    "itemName": "Carnet 12 Corse",  
    "itemVersion": 1,  
    "fareTableCellId": null,  
    "quantity": 1,  
    "ccy": "E",  
    "amount": 13.0  
  }  
]  
}
```


6.6 PURCHASE SALES PACKAGES

L'acquisto rimane identico a quello attuale con l'unica differenza che bisogna passare nel campo in input itemType il valore "SALESPACKAGES" ("itemType": "SALESPACKAGE").

6.6.1 ENDPOINT

Endpoint	
Http Method	POST
Uri	/es/interactive-ticket-manager/purchases