

# Avantik WebAPI Implementation Guide

---



Document Date: 13 Mar 2017

Author: Angkoon Jariyasin, Jaturasri Suansermssak

Version: 40

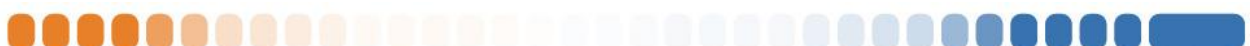
Document Name: WebAPI Implementation Guide

© All rights reserved by Bravo

Bravo Passenger Solutions (Thailand) Co.,Ltd

17<sup>th</sup> Fl. UBCII Building, 591 Sukhumvit Soi 33, Klong Tan Nua, Wattana, Bangkok 10110, Thailand

www.bravo.aero Tel: +66 2 260 8488 Fax: +66 2 260 8490



# 1 RECORD OF CHANGE

Change Request Number	Version No.	Date	A* M D	Title or Brief Description	By
	1	26 Jun 2009	A	Initial Creation	Angkoon
	2	14 Jul 2009	M	Modifications throughout document	Angkoon
		20 Jul 2009	M	Modifications throughout document	Jaturasri
	3	12 Dec 2009	M	Modified Save Booking section and added error code information	Angkoon Jaturasri
		13 Dec 2009	M	Converted to new tikaAERO documentation format	Thomas
	4	03 Feb 2010	A	Update Functionality	Angkoon
	5	22 Feb 2010	A	Update Functionality	Angkoon Jaturasri
	6	27 Apr 2010	A	Update Error Numbers	Angkoon
	7	29 Oct 2010	M	Update Session less Availability	Angkoon
	8	24 Nov 2010	M	Converted to new Avantik documentation format	Jaturasri
	9	20 Apr 2011	M	tikaAERO-2011-1000: Need to enhance API to save Gender	Angkoon
	10	30 Jun 2011	M	Changed date formats in XML examples	Chulanee
	11	02 Jul 2011	A	Adding error code.	Angkoon
	12	02 Dec 2011	A	Adding new field	Angkoon
	13	16 Jan 2013	A	Adding new function	Angkoon
	14	26 Aug 2013	A	Adding add remark function	Angkoon
	15	07 Oct 2013	A	Add Error code for agency registration.	Angkoon
	16	07 Nov 2013	A	Add error code from Payment save	Angkoon
	17	14 Nov 2013	A	Add new function for purchase voucher API	Angkoon
	18	24 Feb 2014	A	Add new function for get passenger manifest.	Angkoon



	19	25 Feb 2014	A	Add language parameter.	Angkoon
	20	27 Mar 2014	M	Add description for missing field.	Angkoon
	21	20 May 2014	M	Add remark function	Angkoon
	22	27 May 2014	M	Update company name and logo	Sawitree
	23	01 AUG 2014	A	Add function for manager client profile.	Angkoon



Change Request Number	Version No.	Date	A* M D	Title or Brief Description	By
	24	08 AUG 2014	A	Add new error code.  New parameter for infant validation.	Angkoon
	25	18 AUG 2014	A	Add new error code.	Angkoon
	26	28 AUG 2014	A	Add parameter to pass currency code to FlightAdd function. [Refer to tikaAERO-2014-1794: Booking API: Retrieve fares in currency based on point of Origin]	Angkoon
	27	27 OCT 2014	A	Add Flow chart of booking process and update XML sample of client profile.	Angkoon
	28	12 DEC 2014	M	Change description of Serviceinitialize function.	Angkoon
	29	23 JAN 2015	A	Add function GetQuoteSummary and adding new error code.	Angkoon
	30	24 FEB 2015	A	Add voucher payment on BookingSave and new error code	Pinyo
	31	09 APR 2015	A	Add Voucher Payment on BookingSave XML	Siriwarch
	32	07 JUL 2015	A	Update DOB and passport expiry date format in Booking XML [p.96] and Save Booking XML [p.139]	Siriwarch
	33	14 AUG 2015	A	Add MultipleFOP and GetInfantCapacity function	Pinyo
	34	08 DEC 2015	A	Add actual number on flight availability & update comment of multiple voucher concept	Pinyo
	35	12 MAY 2016	A	Add other passenger type (youth)	Asada
	36	06 OCT 2016	A	Update new response about server ID from ServiceInitialize	Ratchanee



	37	4 JAN 2017	A	Add FFP method	Ratchanee
	38	20 Feb 2017	A	Add GetPassengerManifest method	Siriwarch
	39	20 Feb 2017	A	Add GetCoporateSessionProfile method	Asada
	40	13 Mar 2017	A	GetSeatMapLayout	Asada

\*A: Added, M: Modified, D: Deleted



## 2 TABLE OF CONTENT

1	Record of Change.....	2
2	Table of Content .....	6
3	Introduction .....	10
3.1	Audience.....	10
3.2	Overview .....	10
4	Implementation and Code Samples.....	11
4.1	Overview .....	11
4.2	Booking API .....	11
4.2.1	ServiceInitialize() .....	18
4.2.2	GetAgency() .....	20
4.2.3	FlightAvailability() .....	21
4.2.4	FlightLowestFareAvailability() .....	24
4.2.5	FlightGroupFareAvailability() .....	27
4.2.6	FlightRedemptionAvailability". () .....	30
4.2.7	FlightAdd() .....	32
4.2.8	BookingInitialize() .....	41
4.2.9	BookingGetSession() .....	43
4.2.10	BookingSave() .....	45
4.2.11	BookingGetItinerary() .....	52
4.2.12	GetSpecialServiceFee() .....	54
4.2.13	AddSpecialService().....	56
4.2.14	RemoveSpecialService() .....	60
4.2.15	GetBaggageFee().....	62



4.2.16	AddBaggageFee()	64
4.2.17	RemoveBaggageFee()	66
4.2.18	ClientLogin()	68
4.2.19	ClientLogout	70
4.2.20	GetClientPassenger	71
4.2.21	GetPointHistory	72
4.2.22	GetActiveBooking	74
4.2.23	GetFlownBooking	76
4.2.24	GetSeatMap()	77
4.2.25	GetSeatMapLayout()	80
4.2.26	SelectSeat()	82
4.2.27	GetFeeDefinition()	84
4.2.28	AddFee()	85
4.2.29	RemoveFee()	90
4.2.30	AddRemark()	92
4.2.31	RemoveRemark()	94
4.2.32	GetPassenger()	97
4.2.33	GetBookings	101
4.2.34	GetQuoteSummary	104
4.2.35	GetInfantCapacity()	108
4.2.36	GetCorporateSessionProfile	109

#### 4.3 Purchase Voucher API .....110



4.3.1	VoucherTemplateList .....	112
4.3.2	SaveVoucher .....	115
4.3.3	ReadVoucher.....	123
4.3.4	VoidVoucher .....	124
4.4	Manage Booking API .....	127
4.4.1	AddRemarkToBooking .....	127
4.4.2	DeleteRemarkFromBooking.....	130
4.4.3	UpdateBookingRemark .....	132
4.4.4	GetActivities.....	135
4.4.5	CompleteRemark .....	137
4.4.6	GetQueCount .....	139
4.5	Manage client profile .....	141
4.5.1	CreateClientProfile .....	141
4.5.2	EditClientProfile .....	147
4.5.3	AddPassengerProfile .....	153
4.5.4	EditPassengerProfile .....	158
4.5.5	PassengerProfileRead .....	163
4.6	Retreiving Passenger Details.....	165
4.6.1	Get Passenger Manifest .....	165
5.	Appendix 1: Serviceinitialize xml.....	168
6.	Appendix 2: Availability XML .....	168
7.	Appendix 3: Booking XML .....	174
8.	Appendix 4: Itinerary XML .....	196
9.	Appendix 5: Save Booking XML.....	243





10.	Appendix 6: Agency XML .....	252
11.	Appendix 7: Error XML.....	253
12.	Appendix 8: special service fee xml .....	261
13.	Appendix 9: Baggage Fee xml .....	264
14.	Appendix 10: Passenger Profile read .....	269
15.	Appendix 11: Get Seatmap .....	283
16.	Appendix 12: GetFeeDefinition XML .....	288
17.	Appendix 13: Voucher template xml .....	290
18.	Appendix 14: Voucher xml.....	294
19.	Appendix 15: PassengersManifest XML.....	299
20.	Appendix 16: Activities .....	304
21.	Appendix 17: RemarkQueue.....	307
22.	Appendix 18: GetBookings.....	309
23.	Appendix 19: Active Booking XML .....	313
24.	Appendix 20: GetPointHistory .....	316
25.	Appendix 21: Flight Redemption Availability response .....	318
26.	APPENDIX 22: PAssenger manifest .....	321
27.	APPENDIX 23: GetCoporateSessionProfile.....	328



---

## 3 INTRODUCTION

The purpose of this manual is to provide the technical information necessary to implement a web services connection to the Avantik WebAPI. It also provides code samples that illustrate how to use the APIs to send data from a website to the Avantik WebAPI.

### 3.1 Audience

The target audience of this implementation guide is the integrator or programmer who will develop the interface between the Booking Interface and the Avantik WebAPI.

### 3.2 Overview

The Avantik WebAPI Engine enables you to integrate a new or existing airline booking engine to the Avantik WebAPI. This document also shows a code sample in .Net C# which can be used as a guide for your application integration.

Please contact the airline sponsoring your Avantik WebAPI integration to get agency code, user name, password and connectivity details.



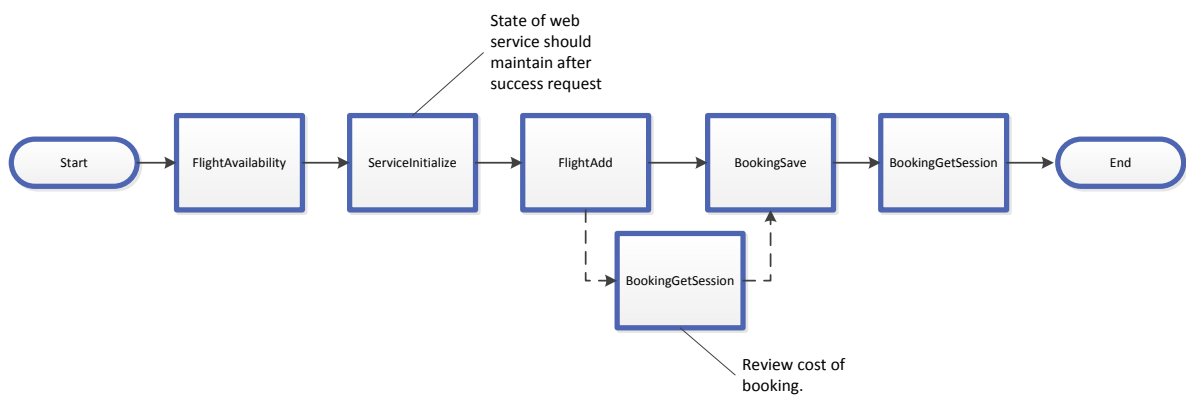
## 4 IMPLEMENTATION AND CODE SAMPLES

### 4.1 Overview

This chapter will provide you guidance on how to use the functions in the Avantik WebAPI. It also includes code samples in .Net C# that illustrate how to use each function to send data to the Avantik core booking engine.

### 4.2 Booking API

Below is the step of how each function of Booking API is call to book passenger selected flight without any ancillary revenue.

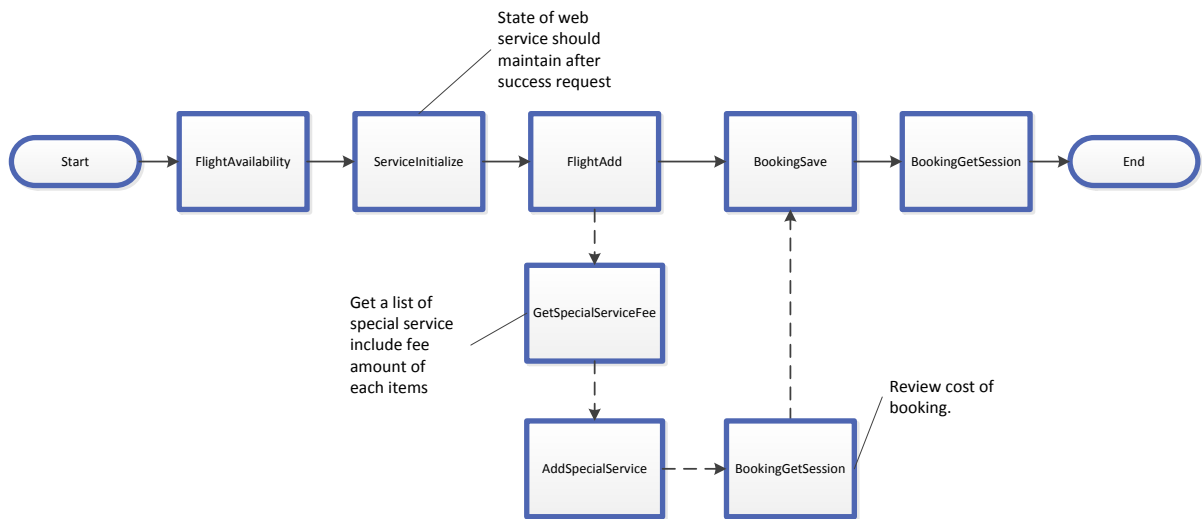


- |                       |               |
|-----------------------|---------------|
| 1. FlightAvailability | Section 4.2.3 |
| 2. ServiceInitialize  | Section 4.2.1 |
| 3. FlightAdd          | Section 4.2.6 |
| 4. BookingGetSession  | Section 4.2.8 |
| 5. BookingSave        | Section 4.2.9 |
| 6. BookingGetSession  | Section 4.2.8 |



Below flow show the step of how to add other ancillary revenue in the booking process.

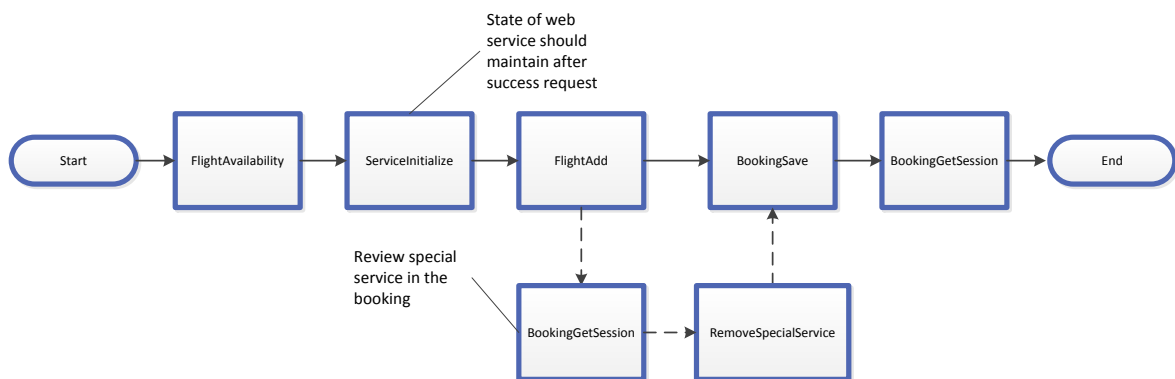
### Add Special Service



- |                         |                |
|-------------------------|----------------|
| 1. FlightAvailability   | Section 4.2.3  |
| 2. ServiceInitialize    | Section 4.2.1  |
| 3. FlightAdd            | Section 4.2.6  |
| 4. GetSpecialServiceFee | Section 4.2.11 |
| 5. AddSpecialService    | Section 4.2.12 |
| 6. BookingGetSession    | Section 4.2.8  |
| 7. BookingSave          | Section 4.2.9  |
| 8. BookingGetSession    | Section 4.2.8  |



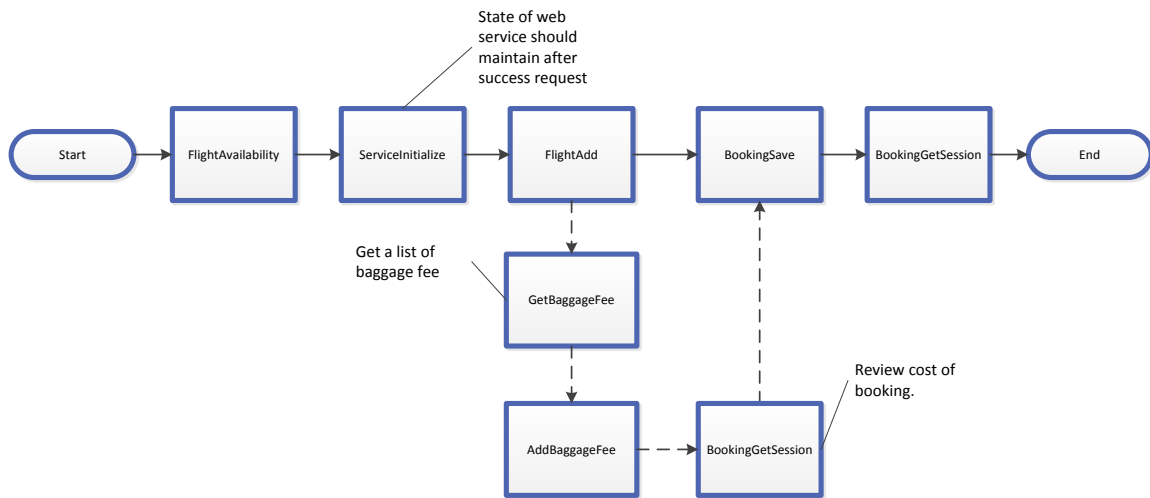
## Remove Special Service



1. FlightAvailability      Section **4.2.3**
2. ServiceInitialize      Section **4.2.1**
3. FlightAdd      Section **4.2.6**
4. BookingGetSession      Section **4.2.8**
5. RemoveSpecialService      Section **4.2.13**
6. BookingSave      Section **4.2.9**
7. BookingGetSession      Section **4.2.8**

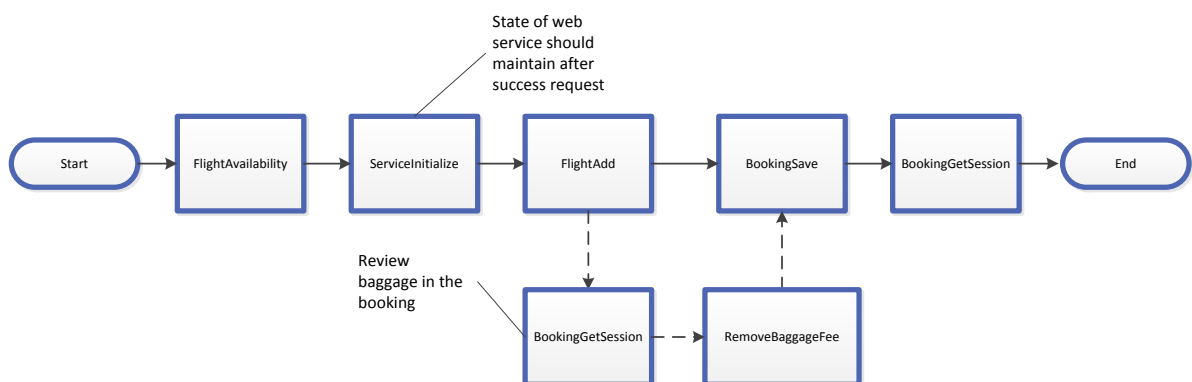


### Add Baggage Fee



- |                       |                |
|-----------------------|----------------|
| 1. FlightAvailability | Section 4.2.3  |
| 2. ServiceInitialize  | Section 4.2.1  |
| 3. FlightAdd          | Section 4.2.6  |
| 4. GetBaggageFee      | Section 4.2.14 |
| 5. AddBaggageFee      | Section 4.2.15 |
| 6. BookingGetSession  | Section 4.2.8  |
| 7. BookingSave        | Section 4.2.9  |
| 8. BookingGetSession  | Section 4.2.8  |

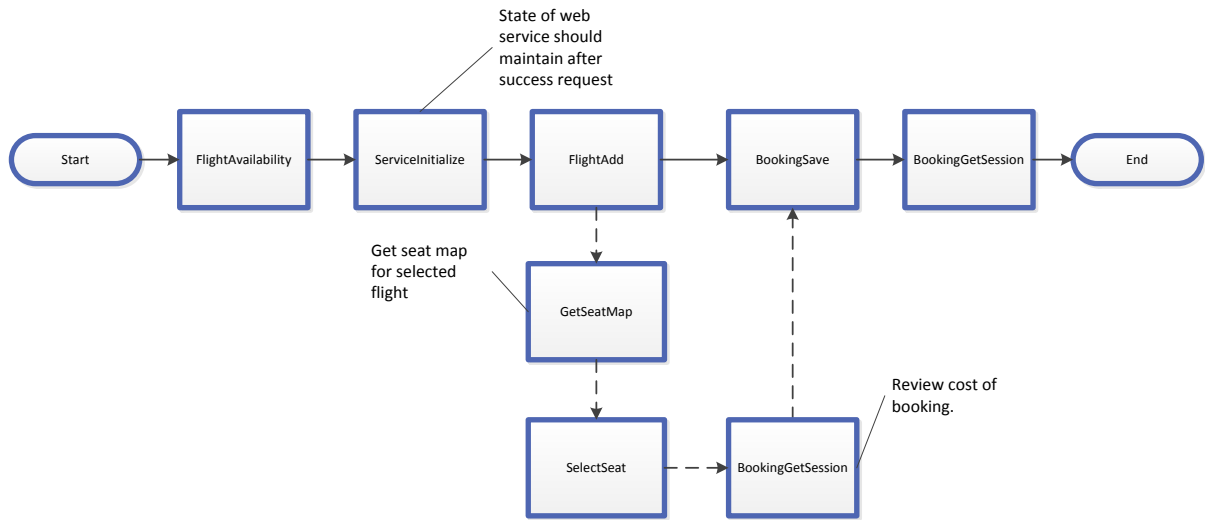
### Remove Baggage Fee



1. FlightAvailability      Section **4.2.3**
2. ServiceInitialize      Section **4.2.1**
3. FlightAdd              Section **4.2.6**
4. BookingGetSession      Section **4.2.8**
5. RemoveBaggageFee      Section **4.2.16**
6. BookingSave            Section **4.2.9**
7. BookingGetSession      Section **4.2.8**



## Seat Selection

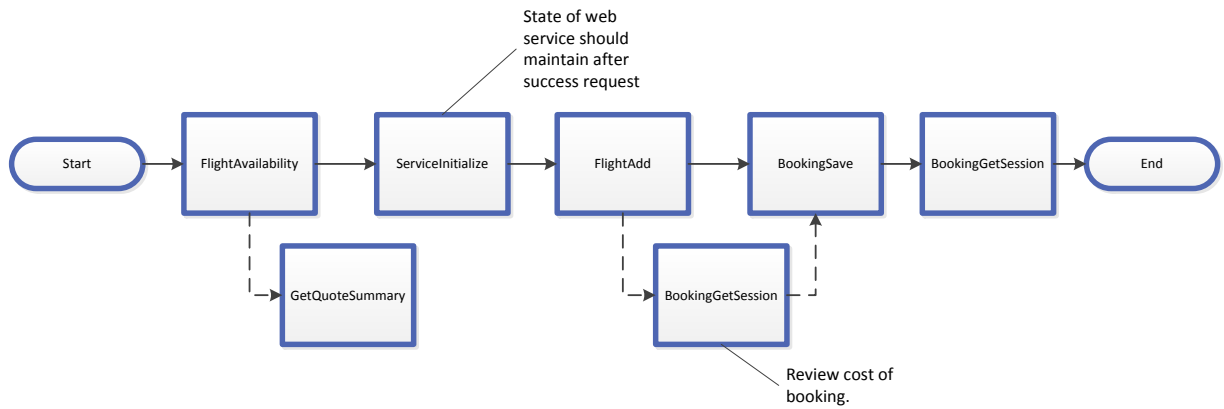


- |                       |                |
|-----------------------|----------------|
| 1. FlightAvailability | Section 4.2.3  |
| 2. ServiceInitialize  | Section 4.2.1  |
| 3. FlightAdd          | Section 4.2.6  |
| 4. GetSeatMap         | Section 4.2.22 |
| 5. SelectSeat         | Section 4.2.23 |
| 6. BookingGetSession  | Section 4.2.8  |
| 7. BookingSave        | Section 4.2.9  |
| 8. BookingGetSession  | Section 4.2.8  |





## GetQuoteSummary



- |                       |                       |
|-----------------------|-----------------------|
| 1. FlightAvailability | Section <b>4.2.3</b>  |
| 2. GetQuoteSummary    | Section <b>4.2.31</b> |
| 3. ServiceInitialize  | Section <b>4.2.1</b>  |
| 4. FlightAdd          | Section <b>4.2.6</b>  |
| 5. GetSeatMap         | Section <b>4.2.22</b> |
| 6. SelectSeat         | Section <b>4.2.23</b> |
| 7. BookingGetSession  | Section <b>4.2.8</b>  |
| 8. BookingSave        | Section <b>4.2.9</b>  |
| 9. BookingGetSession  | Section <b>4.2.8</b>  |



#### 4.2.1 ServiceInitialize()

Name	ServiceInitialize	
Description	Initialization and Authentication web service.	
Output	Error XML (See Appendix 6)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
strAgencyCode	String	Agency code.
strUserName	String	Web service login user name.
strPassword	String	Web service login password.
strLanguageCode	String	Two letter code for language.

When the web service is initiated it needs to retain its existence state between the calls to the web service. Below is a code sample on how to initialize and retain the web service state.

#### Sample



```

Private void InitializeWebservice()
{
    MyService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        //create the proxy
        MyService = new MyService();

        //create a container for the SessionID cookies
        MyService.CookieContainer = new CookieContainer();

        //Initialization and authentication web service.
        MyService.ServiceInitialize("testagent", " test1", " testpass", "EN");

        //store it in Session for the next usage.
        Session["MyService"] = MyService;
    }
}

```

Before you can start the booking process with the API you have to authenticate the web service first by calling **ServiceInitialize()**. This function requires you to pass in the agency code, user name, password, and language code. You have to retain the service state, as explained above. After the service has been initialized the **MyService** object is kept in the session to retain its state. You can find a sample and description of the Availability XML in Appendix 1.



#### 4.2.2 GetAgency()

Name	GetAgency
Description	Read Agency information.
Output	Agency and user XML (See Appendix 5 )
Parameter	
<u>Name</u>	<u>Description</u>
(none)	No parameter required.

This function will return the XML information of Agency and User. A sample XML can be found in Appendix 5. You can use this agency XML to find the available travel agency credit amount which is explained in further detail in the **BookingSave()** section.



### 4.2.3 FlightAvailability()

Name		Availability
Description		Search flight availability for all fares.
Output		Availability XML .
Parameter		
Name	Data Type	Description
strAgencyCode	String	Agency code.
strPassword	String	Web service login password.
strOrigin	String	Airport code of origin.
strDestination	String	Airport code of destination.
strDepartFrom	String	Departure date from.
strDepartTo	String	Departure date to.
strReturnFrom	String	Return date from.
strReturnTo	String	Return date to.
iAdult	Int	Number of adult passengers.
iChild	Int	Number of child passengers.
iInfant	Int	Number of infant passengers.
iOther	Int	Number of other passenger types.
strBookingClass	String	Booking class. This is an <b>optional</b> filter.
strBoardingClass	String	Boarding class. This is an <b>optional</b> filter.
strPromo	String	Promotion code. This is an <b>optional</b> filter.
strLanguageCode	String	Two letter code of language. This is an <b>optional</b> filter.
strOtherPassengerType	String	Name of other passenger type.
strCurrency	String	Currency

The function will be called when you want to search for flight availability. Below is a sample on how to call the function. The **ServiceInitiaize** is not required for the FlightAvailability Call.



**Sample 1**

```

Private void SearchFlight()
{
    MyService objService = New MyService ();

    //Call function GetFlightAvailability

    String xmlResult = string.Empty;

    xmlResult = objService.Availability("AgencyCode", "Password", "GCI",
                                      "LGW",
                                      20090310, 20090330, 20090310,
                                      20090330, 1, 1, 0, 0, "Y", "", "");
}

```

The function allows you to search availability for a date range. As in the code sample, if you want to search flights for an outbound and return travel date range between 10 March 2009 and 30 March 2009, you have to pass in a date value for all four date parameters (dtDepartFrom, dtDepartTo, dtReturnFrom, and dtReturnTo). The date format is as follows: **YYYYMMDD**.

If you want to search for a one way flight on a specific date, for example 10 March 2009, you need to pass in the dtDepartureFrom and dtDepartureTo with the same date value and leave dtReturnFrom and dtReturnTo blank. Below is a sample of how to search for a one way flight on a specific date.



**Sample 2**

```
xmlResult = objService.Availability("AgencyCode", "Password", "GCI", "LGW",  
20090310, 20090310, "", "", 1, 1, 0, 0,  
"Y", "", "");
```

After the function is called it will return the Availability XML. This XML contains all basic flight information with Flight ID and Fare ID which are required for the **FlightAdd()** function call. You can find a sample and description of the Availability XML in Appendix 2.



#### 4.2.4 FlightLowestFareAvailability()

Name	FlightLowestFareAvailability	
Description	Search flight availability for lowest fares.	
Output	Availability XML .	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
strAgencyCode	String	Agency code.
strPassword	String	Web service login password.
strOrigin	String	Airport code of origin.
strDestination	String	Airport code of destination.
strDepartFrom	String	Departure date from.
strDepartTo	String	Departure date to.
strReturnFrom	String	Return date from.
strReturnTo	String	Return date to.
iAdult	Int	Number of adult passengers.
iChild	Int	Number of child passengers.
iInfant	Int	Number of infant passengers.
iOther	Int	Number of other passenger types.
strBookingClass	String	Booking class. This is an <b>optional</b> filter.
strBoardingClass	String	Boarding class. This is an <b>optional</b> filter.
strPromo	String	Promotion code. This is an <b>optional</b> filter.
strLanguageCode	String	Two letter code of language. This is an <b>optional</b> filter.
strOtherPassengerType	String	Name of other passenger type.
strCurrency	String	Currency

The function will be called when you want to search for the lowest flight availability. Below is a sample on how to call the function. The **ServiceInitilaize** is not required for the FlightLowestFareAvailability Call.





### Sample 1

```
Private void SearchFlight()
{
    MyService objService = New MyService ();

    //Call function GetFlightAvailability

    String xmlResult = string.Empty;

    xmlResult = objService.FlightLowestFareAvailability("AgencyCode",
                                                    "Password",
                                                    "GCI", "LGW",
                                                    20090310, 20090330, 20090310,
                                                    20090330, 1, 1, 0, 0, "Y", "", "");
}
```

The function allows you to search availability for a date range. As in the code sample, if you want to search flights for an outbound and return travel date range between 10 March 2009 and 30 March 2009, you have to pass in a date value for all four date parameters (dtDepartFrom, dtDepartTo, dtReturnFrom, and dtReturnTo). The date format is as follows: **YYYYMMDD**.

If you want to search for a one way flight on a specific date, for example 10 March 2009, you need to pass in the dtDepartureFrom and dtDepartureTo with the same date value and leave dtReturnFrom and dtReturnTo blank. Below is a sample of how to search for a one way flight on a specific date.



**Sample 2**

```
xmlResult = objService.FlightLowestFareAvailability("AgencyCode",  
                                                    "Password", "GCI", "LGW",  
                                                    20090310, 20090310, "", "", 1, 1, 0, 0,  
                                                    "Y", "", "");
```

After the function is called it will return the Availability XML. You can find a sample and description of the Availability XML in Appendix 3.



#### 4.2.5 FlightGroupFareAvailability()

Name	FlightGroupFareAvailability	
Description	Search flight availability for the lowest of each group fares.	
Output	Availability XML .	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
strAgencyCode	String	Agency code.
strPassword	String	Web service login password.
strOrigin	String	Airport code of origin.
strDestination	String	Airport code of destination.
strDepartFrom	String	Departure date from.
strDepartTo	String	Departure date to.
strReturnFrom	String	Return date from.
strReturnTo	String	Return date to.
iAdult	Int	Number of adult passengers.
iChild	Int	Number of child passengers.
iInfant	Int	Number of infant passengers.
iOther	Int	Number of other passenger types.
strBookingClass	String	Booking class. This is an <b>optional</b> filter.
strBoardingClass	String	Boarding class. This is an <b>optional</b> filter.
strPromo	String	Promotion code. This is an <b>optional</b> filter.
strLanguageCode	String	Two letter code of language. This is an <b>optional</b> filter.
strOtherPassengerType	String	Name of other passenger type.
strCurrency	String	Currency

The function will be called when you want to search for flight availability of the lowest fare of each fare group.

Below is a sample on how to call the function. The **ServiceInitilaize** is not required for the FlightGroupFareAvailability Call.



### Sample 1

```
Private void SearchFlight()
{
    MyService objService = New MyService ();

    //Call function GetFlightAvailability

    String xmlResult = string.Empty;

    xmlResult = objService.FlightGroupFareAvailability("AgencyCode",
                                                    "Password",
                                                    "GCI", "LGW",
                                                    20090310, 20090330, 20090310,
                                                    20090330, 1, 1, 0, 0, "Y", "", "");
}
```

The function allows you to search availability for a date range. As in the code sample, if you want to search flights for an outbound and return travel date range between 10 March 2009 and 30 March 2009, you have to pass in a date value for all four date parameters (dtDepartFrom, dtDepartTo, dtReturnFrom, and dtReturnTo). The date format is as follows: **YYYYMMDD**.

If you want to search for a one way flight on a specific date, for example 10 March 2009, you need to pass in the dtDepartureFrom and dtDepartureTo with the same date value and leave dtReturnFrom and dtReturnTo blank. Below is a sample of how to search for a one way flight on a specific date.



**Sample 2**

```
xmlResult = objService.FlightGroupFareAvailability("AgencyCode",  
                                                    "Password", "GCI", "LGW",  
                                                    20090310, 20090310, "", "", 1, 1, 0, 0,  
                                                    "Y", "", "");
```

After the function is called it will return the Availability XML. You can find a sample and description of the Availability XML in Appendix 3.



#### 4.2.6 FlightRedemptionAvailability". ()

Name	FlightGroupFareAvailability	
Description	Search flight availability for the Redemption flight.	
Output	Availability XML .	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
strAgencyCode	String	Agency code.
strPassword	String	Web service login password.
strOrigin	String	Airport code of origin.
strDestination	String	Airport code of destination.
strDepartFrom	String	Departure date from.
strDepartTo	String	Departure date to.
strReturnFrom	String	Return date from.
strReturnTo	String	Return date to.
iAdult	Int	Number of adult passengers.
iChild	Int	Number of child passengers.
iInfant	Int	Number of infant passengers.
iOther	Int	Number of other passenger types.
strBookingClass	String	Booking class. This is an <b>optional</b> filter.
strBoardingClass	String	Boarding class. This is an <b>optional</b> filter.
strPromo	String	Promotion code. This is an <b>optional</b> filter.
strLanguageCode	String	Two letter code of language. This is an <b>optional</b> filter.
strOtherPassengerType	String	Name of other passenger type.
strCurrency	String	Currency



The function allows you to search the redemption flight availability for a date range. As in the code sample, if you want to search flights for an outbound and return travel date range between 10 March 2009 and 30 March 2009, you have to pass in a date value for all four date parameters (dtDepartFrom, dtDepartTo, dtReturnFrom, and dtReturnTo). The date format is as follows: **YYYYMMDD**.

If you want to search for a one way flight on a specific date, for example 10 March 2009, you need to pass in the dtDepartureFrom and dtDepartureTo with the same date value and leave dtReturnFrom and dtReturnTo blank. Below is a sample of how to search for a one way flight on a specific date.

#### Sample

```
xmlResult = objService.FlightGroupFareAvailability("AgencyCode",
                                                    "Password", "GCI", "LGW",
                                                    20090310, 20090310, "", "", 1, 1, 0, 0,
                                                    "Y", "", "");
```

After the function is called it will return the Availability XML. You can find a sample and description of the Availability XML in Appendix 3.



#### 4.2.7 FlightAdd()

Name	FlightAdd	
Description	Called when flights are selected from Availability XML.	
Output	Error XML (See Appendix 6)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
strFlightXml	String	Availability XML (See Appendix 1)

This function will be called when you want to select a flight from the Availability XML. The parameter(s) that is/are required are the Availability XML(s) of the flight(s) you want to select.

##### Sample 1

<Header>

<adult>\${#TestCase#iAdult}</adult>

<child>\${#TestCase#iChild}</child>

<infant>\${#TestCase#iInfant}</infant>

<currency\_rcd>GBP</currency\_rcd>

</Header>

<FlightSegment>

<flight\_id>{031CA90F-35A3-46F8-8AF0-CB675EE184D8}</flight\_id>

<fare\_id>{A96933ED-8135-4D4A-BF0D-8BCC53615187}</fare\_id>





```

    <origin_rcd>GCI</origin_rcd>

    <destination_rcd>LGW</destination_rcd>

    <transit_flight_id></transit_flight_id>

    <transit_airport_rcd></transit_airport_rcd>

</FlightSegment>

```

```

Private void SelectedFlight()
{
    MyService objService = (MyService)Session["MyService"];

    //Required Service initialized first.

    InitializeWebservice();

    objService = (MyService)Session["MyService"];

    //Call function FlightAdd

    int iResult;

    iResult = objService.FlightAdd(strSelectedFlightXml);

    if (iResult = 1)

```



```
{  
  
    //Add Complete  
  
    //show itinerary  
  
    .....  
  
}  
  
Else  
  
{  
  
    //Save Failed  
  
    .....  
  
}  
  
}
```

You have to first initialize the web service by call the **InitializeWebservice()** then extract your selected flight XML from the Availability XML before passing it into the function. In Sample 5, a sample of **FlightAdd()** xml.

## Sample 2



```

<Booking>

  <Header>

    <adult></adult>

    <child></child>

    <infant></infant>

    <other></other>

    <other_passenger_type></other_passenger_type>

    <currency_rcd></ currency_rcd >

  </Header>

  <FlightSegment>

    <flight_id>

      9b44393f-1728-460a-8ed3-c9a6336eefdb

    </flight_id>

    <fare_id>

      2a16c898-e301-4ddf-bacd-76ce4a158a4d

    </fare_id>

    <origin_rcd>BGY</origin_rcd>

    <destination_rcd>CTA</destination_rcd>

    <boarding_class_rcd>Y</ boarding_class_rcd >

    <booking_class_rcd>Y</ booking_class_rcd >

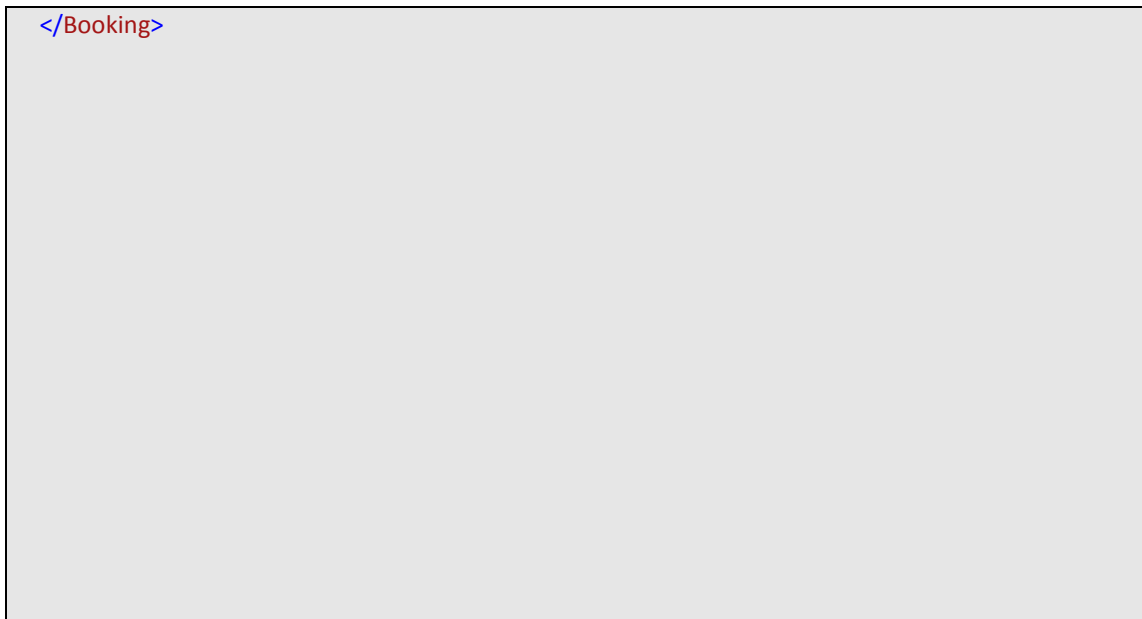
    <transit_flight_id></transit_flight_id>

    <transit_airport_rcd></transit_airport_rcd>

  </FlightSegment>

```





**Header Field Description**

Name	Description
Adult	Number of adult.
Child	Number of child.
Infant	Number of infant.
currency_rcd	Booking Currency code.

**FlightSegment Field Description**

Name	Description
flight_id	Unique flight ID. Used when calling <b>FlightAdd()</b> function.
origin_rcd	Origin airport code.
destination_rcd	Airport destination code.
fare_id	Unique fare ID.
transit_flight_id	Transit flight ID.
transit_airport_rcd	Transit airport code.
boarding_class_rcd	Boarding class code.
booking_class_rcd	Booking class code.
transit_boarding_class_rcd	Transit flight boarding class code.
transit_booking_class_rcd	Transit flight booking class code.

If you want to select **Flight 600** from the above sample XML for the outbound and **Flight 605** for the return you have to reconstruct the XML as in the sample below before passing it into the **FlightAdd()** function.



### Sample 3

```

<Booking>

  <Header>

    <adult>1</adult>

    <child>0</child>

    <infant>0</infant>

  </Header>

  <FlightSegment>

    <flight_id>

      cccb9f32-6571-4bdd-b75a-c9e1403409c0

    </flight_id>

    <fare_id>

      90f7b0e9-eeac-42d1-ab8d-5931fc5c1d64

    </fare_id>

    <origin_rcd>ACI</origin_rcd>

    <destination_rcd>GCI</destination_rcd>s

    <destination_rcd>GCI</destination_rcd>

    <boarding_class_rcd>Y</ boarding_class_rcd >

    <booking_class_rcd>Y</ booking_class_rcd >

    <transit_airport_rcd></transit_airport_rcd>

  </FlightSegment>

  <FlightSegment>

    <flight_id>
  
```



Y

```

5e87a89b-da69-11de-b54d-001641975e09

</flight_id>

<fare_id>

b1994a20-2103-11df-8509-001641975def

</fare_id>

<origin_rcd>GCI</origin_rcd>

<destination_rcd>BRS</destination_rcd>

<transit_flight_id>

ee88e6ce-f533-11de-b575-0016d30a935b

</transit_flight_id>

<transit_airport_rcd>GCI</transit_airport_rcd>

<transit_boarding_class_rcd>Y</transit_boarding_class_rcd>

<transit_booking_class_rcd>Y</transit_booking_class_rcd>

</FlightSegment>

</Booking>

```



The highlight flight segment in **Sample 3** is how the transit flight xml that extract from sample Availability XML in Appendix 3 look like.

The function will return code “1” if the segments have been successfully saved. All other code numbers returned are error codes. You can find a complete sample Availability XML in Appendix .





#### 4.2.8 BookingInitialize()

Name	BookingInitialize
Description	Clears temporary booking information.
Output	Error XML (See Appendix 6)
Parameter	
<u>Name</u>	<u>Description</u>
(none)	No parameter required.

When the booking process is completed the temporary booking information has to be cleared before starting a new booking again. The function **BookingInitialize()** needs to be called when you want to clear temporary booking information.

Below is a function call code sample.

#### Sample

```
Private void ProcessSaveBooking()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
```



```

//Service is not initialized so you need to initialize it first.

InitializeWebservice();

objService = (MyService)Session["MyService"];

}

//Call BookingInitialize Function to clear temp.

int iResult;

iResult = objService.BookingInitialize() ;

if (iResult = 1)

{

    //Clear temp success.

    .....

}

Else

{

    //Save Failed

    .....

}

}

```



#### 4.2.9 BookingGetSession()

Name	BookingGetSession
Description	Retrieves current temporary booking information.
Output	Temp Booking XML (See Appendix 2).
Parameter	
<u>Name</u>	<u>Description</u>
(none)	No parameter required.

This function will return the current state of your booking information that is kept in the API session. The function will return XML information which looks similar to the XML that you get from the **BookingGetItinerary()** function.

Below is a sample of how the **BookingGetSession()** function should be used.

#### Sample

```
Private void GetTempBooking()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        //Service is not initialized so you need to initialize it first.

        InitializeWebservice();
    }
}
```



```
objService = (MyService)Session["MyService"];  
  
}  
  
//This function is called to fill in passenger name and lastname.  
  
xmlResult = objService.BookingGetSession();  
  
}
```

The function does not require any parameter to be passed into, but the function can be called only after the **FlightAdd()** function has been called. When the **FlightAdd()** is called, the API will create temporary booking information so if you want to view or use this temporary booking you can get it by calling the function **BookingGetSession()**.



#### 4.2.10 BookingSave()

Name	BookingSave	
Description	Saves booking information.	
Output	Error XML (See Appendix 6)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
strXml	String	Pass in BookingHeader, Passenger and Payment XML. (See Appendix 4)

This function is called when you want to complete the booking and save it to the database. The **BookingSave()** function requires the XML BookingHeader and Passenger that you get from the **BookingGetSession()** function.

Before passing the XML into the function you are required to fill in the empty XML fields such as passenger first name and last name. The function will return code once the save action has been completed.



Below is a function call code sample

#### Sample 1

```
Private void ProcessSaveBooking()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        //Service is not initialized so you need to initialize it first.
        InitializeWebservice();
        objService = (MyService)Session["MyService"];
    }

    //This function is called to fill in passenger first name and last name.
    xmlResult = FillPassengerNameAndLastname(selectedFlightResultXml);

    //Call SaveBooking Function after the xml is completed.

    int iResult;

    iResult = objService.BookingSave(xmlResult) ;

    if (iResult = 1)
```



```

{
    //Save Complete

    //show itinerary

    .....

}

Else

{

    //Save Failed

    .....

}

}

```

When you call **BookingGetSession()** after the **FlightAdd()** function has been called, you will get a full Booking XML. A sample is provided in Appendix 2. Then you have to extract the Header Section and Passenger section and fill in the empty information before passing it into the **BookingSave()** function. Below is a sample booking XML that has to be passed into the function.



Sample 2

```

<Booking>

  <BookingHeader>

    <contact_name></contact_name>

    <email></email>

    <phone_mobile></phone_mobile>

    <phone_home></phone_home>

    <phone_business></phone_business>

    .

    . <!-- No other fields shown to simplify sample. -->

    .

  </BookingHeader>

  <Passenger>

    <passenger_id>90917fbb-04df-4706-8129-67598e5462b2</passenger_id>

    <passenger_type_rcd>ADULT</passenger_type_rcd>

    <lastname></lastname>

    <firstname></firstname>

    . <!-- No other fields shown to simplify sample. -->

    .

  </Passenger>

  <Payment>

    <form_of_payment_rcd>INV</form_of_payment_rcd>

    <currency_rcd>GBP</currency_rcd>

    <payment_amount>214.00</payment_amount>

  </Payment>

</Booking>

```





Sample 3

```

<Booking>

  <BookingHeader>

    <contact_name></contact_name>

    <email></email>

    <phone_mobile></phone_mobile>

    <phone_home></phone_home>

    <phone_business></phone_business>

    .

    . <!-- No other fields shown to simplify sample. -->

    .

  </BookingHeader>

  <Passenger>

    <passenger_id>90917fbb-04df-4706-8129-67598e5462b2</passenger_id>

    <passenger_type_rcd>ADULT</passenger_type_rcd>

    <lastname></lastname>

    <firstname></firstname>

    <known_traveler_number>20099975</known_traveler_number>

    . <!-- No other fields shown to simplify sample. -->

    .

  </Passenger>

  <Payment>

    <form_of_payment_rcd>INV</form_of_payment_rcd>

    <currency_rcd>GBP</currency_rcd>

    <payment_amount>214.00</payment_amount>

  </Payment>

</Booking>

```



In **Sample 2** only the mandatory fields that are required to be filled in before you pass the XML to the function are displayed, other fields that are not shown are optional. In the **Header** section you are required to fill in contact details and for the **Passenger** section the passenger first name and last name are required fields. For the **Payment** section you have to fill in the form of payment code (which can be referenced from Avantik). After all mandatory fields in the XML have been filled, just pass it into the **BookingSave()** function to complete the booking process. You can find the full XML request sample in Appendix 4.

In **Sample 3** for add known\_traveler\_number in passenger node that this field will be saved in Avantik System.

The Avantik WebAPI supports three types of forms of payment: credit agency account, credit card and voucher. The credit card payment will be available only when the Airline is able to process online payments through a Payment Gateway. For Credit Agency Account Payments, you will have to check the credit available by calling **GetAgency()** before submitting **BookingSave()** with the credit agency form of payment. The **GetAgency()** function will return the Agency XML, in this XML you will find two nodes **<booking\_payment>** and **<agency\_account>**. To find the available amount you have to subtract **booking\_payment** from **agency\_account** to ensure the agency has sufficient credit available to pay for the booking. If you submit a request and there is not a sufficient credit amount available you will get the error code 503.

If you want to book the passenger with client profile first you need to call the function **ClientLogon()**. Then when you come to the booking save you have to set the flag name **<apply\_client\_profile\_flag>** to 1 in the booking header section. Now you want to use the member of the client profile account to be a passenger of the booking. First you have to call the function **GetClientPassenger()**. Function **GetClientPassenger()** will return a list of passenger profile which are the member of the client that logon. You have to take the value of field **<passenger\_profile\_id>** of the selected passenger profile and put that Id into the field **<passenger\_profile\_id>** in Booking save xml input in **<Passenger>** section. For Voucher Payments, You need to specific **form\_of\_payment\_rcd** as VOUCHER and also fill in voucher number and password into the field **<voucher\_number>** and **<password>** in **<Payment>** section. You can find the sample BookingSave in XML in **Appendix 5**.



Now **BookingSave()** enhancement is supported multiple form of payments; multiple vouchers and voucher with credit card. To Enable or disable MultipleVoucher and MultipleFOP, both are controlled by application setting in Avantik WebAPI. The sample depict XML request for multiple voucher and voucher with credit card are also described in **Appendix 4**.

#### Multiple vouchers condition

- Due to API allows validate more than one voucher at a time, if there is multiple voucher enabled and balance is more than charge amount and other conditions are valid, API should deduct voucher upon priority of the sequence of Voucher validation.
- The maximum number of vouchers allowance per request is 2.
- The remaining balance on vouchers are not deducted and the transaction is incomplete with error code return if any invalid voucher is passed to function.

#### Voucher with credit card condition

- Same as multiple vouchers condition, API should deduct voucher upon priority of the sequence of Voucher validation. If voucher balance still not enough, credit card payment would be deducted to pay remaining amount.
- 2 vouchers with one credit card is the maximum allowance per request.



- The remaining balance on vouchers are not deducted and the transaction is incomplete with error code return if credit card authorisation fails – otherwise the credit card is not deducted when any invalid voucher is passed to function.

#### 4.2.11 BookingGetItinerary()

Name	BookingGetItinerary	
Description	Get booking itinerary.	
Output	Itinerary XML (See Appendix 4)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
RecordLocator	String	A Booking record locator (Booking reference) from booking header XML (See Appendix 3).

When you want the passenger to view the itinerary you have to call the **BookingGetItinerary()** function. This function will return the XML which you can use to generate an itinerary in HTML or any other format for the passenger to view.

Below is a sample explaining how to use the **BookingGetItinerary()** function.



**Sample**

```

Private void GetItinerary()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        //Service is not initialized so you need to initialize it first.
        InitializeWebservice();
        objService = (MyService)Session["MyService"];
    }

    //This function is called to fill in passenger first name and last name.

    xmlResult = objService.BookingGetItinerary("LVBC7R");
}

```

Only  
the

booking reference code (also referred to as a PNR/Record Locator) is required to get a response. The data that is returned by the function is the Itinerary XML that you can use to generate an itinerary view (for the passenger) which can be displayed in HTML or any other format.



**4.2.12 GetSpecialServiceFee()**

Name	GetSpecialServiceFee	
Description	Get a list of selected Special Service with the fee amount.	
Output	Special Service XML (See Appendix 8)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
ssrCode	String[]	An array of Special Server code.
flightSegment	GUID[]	An array of flight segment id.
LanguageCode	String	Language Code. Ex “EN”

This function will return a list of special services that are available to purchase for this booking. You can find the sample code in **Sample 12**. You will be using the information return from the function for use in the function AddSpecialService() to add the service to the booking.

**Sample**

```

Private void GetSpecialService()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        string[] ssrCode = { "TRAD", "WCHC"};

        Guid[] gBookingSegmentId = { new Guid("4C870464-3D6A-44A8-9C18-
                                     4DACF27E58CE") };
    }
}

```



```
xmlSSR = objService.BookingGetItinerary(ssrCode,  
                                         gBookingSegmentId,  
                                         "EN");  
  
}  
  
}
```



#### 4.2.13 AddSpecialService()

Name	AddSpecialService()	
Description	Use for add selected special service to the booking.	
Output	Error XML (See Appendix 6)	
Parameter (SpecialServiceRequest)		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
PassengerSegmentServiceId	Guid	Guid provide by requester. This will later use as a reference id from remove SSR.
PassengerId	Guid	Id for the passenger that request the SSR.
BookingSegmentId	Guid	Id of booking segment that request for SSR. Id is getting from Booking XML.
OriginRcd	String	Origin Airport code
DestinationRcd	String	Destination Airport code.
SpecialServiceRcd	String	SpecialService Code. (special_service_rcd)
ServiceText	String	Special Service Display name.
NumberOfUnit	Int16	Number of unit for the selected SSR.
ServiceOnRequestFlag	Boolean	Service on Request flag. Copy a value from the SSR xml return from function GetSpecialServiceFee().

This function is use for adding selected special service to the booking. Below is the detail of the input that is required to pass into the parameters.

##### PassengerSegmentServiceId:

You will have to generate a unique GUID for each request items. This parameter will be later use as a reference when you want to remove the selected Special Service out of the booking.

##### PassengerId:

This is the ID of the passenger that requests the Special Service.

##### BookingSegmentId:





This is the ID of the flight segment that selects the Special Service.

**SpecialServiceRcd:**

This will be the special service code.

**NumberOfUnit:**

This is the number of unit that passenger selects for the SSR.

**ServiceOnRequestFlag:**

You have to take the value of field `<service_on_request>` from Special Service Fee XML that you get from the function call `GetSpecialServiceFee()` and pass into this parameter.

Below is the code sample of how to call the function.

**Sample**

```
Private void AddSpecialService()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        objService.SpecialServiceRequest[] segmentService = new
        objService.SpecialServiceRequest[1];

        //Create new instance for each request item.
```



```
segmentService[0] = new ApiTest.tikApi.SpecialServiceRequest();

//Pass in the parameter.

segmentService[0].PassengerSegmentServiceId = new

    Guid(segmentServiceId);

segmentService[0].PassengerId = new Guid(passengerID);

segmentService[0].BookingSegmentId = new

    Guid(bookingSegmentId);

segmentService[0].OriginRcd = originRcd;

segmentService[0].DestinationRcd = destinationRcd;

segmentService[0].SpecialServiceRcd = specialServiceRcd;

segmentService[0].ServiceText = serviceDisplayName;

segmentService[0].NumberOfUnit = iNumberOfUnit;

segmentService[0].ServiceOnRequestFlag = true;
```



```
//Call Function.
```

```
string strResult = objService.AddSpecialService(segmentService);
```

```
}
```

```
}
```



#### 4.2.14 RemoveSpecialService()

Name	RemoveSpecialService()	
Description	Use for remove the selected special service out of the booking.	
Output	Error XML (See Appendix 6)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
PassengerSegmentServiceId	Guid[]	The array of Id of the SSR that you want to remove.

This function is used for removing the SSR out of the booking. You can remove the SSR by just passing the PassengerSegmentServiceId that was generate by you from the call of **AddSpecialService()**.

Below is the code sample of how to call the function.

#### Sample

```

Private void RemoveSpecialService()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        objService.SpecialServiceRequest[] segmentService = new
        objService.SpecialServiceRequest[1];

        Guid sid = new Guid("4C870464-3D6A-44A8-9C18-
        4DACF27E58CE");

        Guid[] ServiceId = { sid };
    }
}

```



```
//Call Function.
```

```
string strResult = objService.RemoveSpecialService(ServiceId);
```

```
}
```

```
}
```



#### 4.2.15 GetBaggageFee()

Name	GetBaggageFee()	
Description	Function will return a list of baggage information that can be selected.	
Output	Bagagge Fee XML (See Appendix 9)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
LanguageCode	String	Language Code.
BookingSegmentId	Guid	Segment ID
PassengerId	Guid	Passenger ID
outBoundFlight	Boolean	Flag to tell that the selected segment is an outbound or inbound.  <b>True</b> = OutBound.  <b>False</b> = InBound.

This function will return a list of baggage fee amount base on the number of unit that passenger can select. Due to a different flight segment and different passenger might have a different baggage fee amount so you have to identify when passenger and which segment by passing in the BookingSegmentId and PassengerID.

Below is the sample code to show how to call the function.

#### Sample

```
Private void GetBaggageFee()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
```



```
Guid gSegmentId = new Guid(("4C870464-3D6A-44A8-9C18-  
4DACF27E58CE"));  
  
Guid gPassengerId = new Guid(("4C870464-3D6A-44A8-9C18-  
4DACF27E58CE"));  
  
//Call Function.  
  
string strResult = objService.GetBaggageFee("EN",  
gSegmentId,  
gPassengerId,  
true);  
  
}  
  
}
```



#### 4.2.16 AddBaggageFee()

Name	AddBaggageFee()	
Description	Use for add the selected baggage to the booking.	
Output	Error XML (See Appendix 6)	
Parameter (BaggageRequest)		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
NumberOfUnit	Int16	Selected number of unit.
BookingSegmentId	Guid	Segment ID
PassengerId	Guid	Passenger ID
outBoundFlight	Boolean	Flag to tell that the selected segment is an outbound or inbound.  <b>True</b> = OutBound.  <b>False</b> = InBound.

This function is used for adding the baggage to the booking. Below is the code sample of how to call the function.

#### Sample

```
Private void AddSpecialService()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
```





```
objService.BaggageRequest [] BaggagetService = new

objService.BaggageRequest [1];

//Create new instance for each request item.

BaggagetService[0] = new ApiTest.tikApi.BaggageRequest();

//Pass in the parameter.

BaggagetService[0].BookingSegmentId = new

    Guid(bookingSegmentId);

BaggagetService[0].PassengerId = new Guid(passengerID);

BaggagetService[0].NumberOfUnit = iNumberOfUnit

segmentService[0].OutBoundFlight = true;

//Call Function.

string strResult = objService.AddBaggageFee(segmentService);

}

}
```



#### 4.2.17 RemoveBaggageFee()

Name	RemoveBaggageFee()	
Description	Use for remove the selected baggage out of the booking.	
Output	Error XML (See Appendix 6)	
Parameter (BaggageRequest)		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
NumberOfUnit	Int16	Selected number of unit you want to remove.
BookingSegmentId	Guid	Segment ID
PassengerId	Guid	Passenger ID
outBoundFlight	Boolean	Flag to tell that the selected segment is an outbound or inbound.  <b>True</b> = OutBound.  <b>False</b> = InBound.

This function is used for removing the baggage out of the booking. Below is the code sample of how to call the function.

#### Sample

```
Private void RemoveSpecialService()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        objService.BaggageRequest [] BaggaGetService = new
```



```
objService.BaggageRequest [1];

//Create new instance for each request item.

BaggageService[0] = new ApiTest.tikApi.BaggageRequest();

//Pass in the parameter.

BaggageService[0].BookingSegmentId = new

    Guid(bookingSegmentId);

BaggageService[0].PassengerId = new Guid(passengerID);

BaggageService[0].NumberOfUnit = iNumberOfUnit

segmentService[0].OutBoundFlight = true;

//Call Function.

string strResult = objService.RemoveBaggageFee(segmentService);

}

}
```



#### 4.2.18 ClientLogon()

Name	ClientLogon()	
Description	Client profile login.	
Output	Error XML (See Appendix 6)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
clientNumber	String	Client Profile number of register email address.
password	String	Login password

If you want to call the function that required client profile information, you have to login into the client profile system by call function ClientLogon(). Below is a list of function that required the client logon before it can be call.

- GetClientPassenger
- GetFlownBooking
- GetActiveBooking
- Make Booking with client profile.

Below is the code sample on how to call client logon function.

#### Sample

```
Private void ClientLogon()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        //Call Function.
    }
}
```



```
        string strResult = objService.ClientLogon(clientNumber, password);  
    }  
}
```



#### 4.2.19 ClientLogout

Name	ClientLogon()	
Description	Logout of client profile system.	
Output	Error XML (See Appendix 6)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
NONE	NONE	NONE

This function is used for logging out from the client profile system. Below is the code sample of how to call this function.

#### Sample

```
Private void ClientLogon()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        //Call Function.
        string strResult = objService.ClientLogout();
    }
}
```



#### 4.2.20 GetClientPassenger

Name	GetClientPassenger()	
Description	Function returns a list of member of the client that logon.	
Output	Client Passenger XML (See Appendix 9)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
NONE	NONE	NONE

This function will return a list of profile member of the client profile that logon into the system by calling function ClientLogon(). You can use this list to book the profile member as the passenger in the Booking (Detail description in **BookingSave** Topic). You can find the sample XML result in **Appendix 9**.

Below is the code sample of how to call the function.

#### Sample

```
Private void GetClientPassenger()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        //Call Function.

        string strResult = objService.GetClientPassenger();
    }
}
```



#### 4.2.21 GetPointHistory

This function will return all point history of the client profile that logon into the system by calling function GetPointHistory ().

Name	GetPointHistory ()	
Description	Function returns a list of point history of the client who logon.	
Output	Active booking XML (See Appendix 20)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
ClientProfileId	Guid	The GUID id or client profile
TransactionFrom	Date/time	Date range that you would like to search from.
TransactionTo	Date/time	Date range that you would like to search to
AllPoint	boolean	All point included expired number (True/False)

#### Sample

```
<soap:Body>
  <tem:GetPointHistory>
    <!--Optional:-->
    <tem:strClientProfileId>f2c8e43b-b4a8-4f92-88c5-01bce0202d30</tem:strClientProfileId>
    <!--Optional:-->
    <tem:strTransactionFrom>20160901</tem:strTransactionFrom>
    <!--Optional:-->
    <tem:strTransactionTo>20160930</tem:strTransactionTo>
    <!--Optional:-->
    <tem:strAllPoint>true</tem:strAllPoint>
```





```
</tem:GetPointHistory>

</soap:Body>
```



#### 4.2.22 GetActiveBooking

This function will return a list of active booking of the client profile that logon into the system by calling function ClientLogon(). You can use this list to view a list of booking history that flight status is not "FLOWN"

Name	GetActiveBooking()	
Description	Function returns a list of active booking of the client that logon.	
Output	Active booking XML (See Appendix 19)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
NONE	NONE	NONE

Below is the code sample of how to call the function.

#### Sample

```
Private void GetActiveBooking()
{
    MyService objService = (MyService)Session["MyService"];
```



```
If (MyService == null)

{

    //Call Function.

    string strResult = objService.GetAvtiveBookings();

}

}
```



#### 4.2.23 GetFlownBooking

Name	GetFlownBooking()	
Description	Function returns a list of booking that the fligh is already flown.	
Output	Flown booking XML (See Appendix 19)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
NONE	NONE	NONE

This function will return a list of flown booking of the client profile that logon into the system by calling function ClientLogon(). You can use this list to view a list of booking history that flight status is “FLOWN”

Below is the code sample of how to call the function.

#### Sample

```

Private void GetFlownBooking()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        //Call Function.
        string strResult = objService.GetFlownBookings();
    }
}

```



#### 4.2.24 GetSeatMap()

Name	GetSeatMap()	
Description	Use for remove the selected baggage out of the booking.	
Output	Get Seatmap XML (See Appendix 10)	
Parameter (BaggageRequest)		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
originRcd	String	Airport Origin Code.



destinationRcd	String	Airport Destination Code.
flightId	Guid	Flight Id of the seat map layout.
boardingClass	String	Flight boarding class. If this value passin, the function will return the layout based on boarding class value.
bookingClass	String	Flight booking class. If this value passin, the function will return the layout based on booking class value.

The software client will usually use the xml result from this function to generate the visual seatmap on their client application. You have to render the seatmap in table format base on the value of layout\_row and layout\_column. You can find the sample SetMap XML in **Appendix 10**.

Below is the code sample of how to call the function.

#### Sample

```
Private void GetSeatMap()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        //Call Function.

        string strResult = objService.GetSeatMap("KIX", "FUK",

                                                gFlightId, "",

                                                "");
    }
}
```



```
}  
}
```



#### 4.2.25 GetSeatMapLayout()

Name	GetSeatMap()	
Description	Use for remove the selected baggage out of the booking.	
Output	Get Seatmap XML (See Appendix 10)	
Parameter (BaggageRequest)		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
originRcd	String	Airport Origin Code.
destinationRcd	String	Airport Destination Code.
flightId	Guid	Flight Id of the seat map layout.
boardingClass	String	Flight boarding class. If this value passin, the function will return the layout based on boarding class value.
bookingClass	String	Flight booking class. If this value passin, the function will return the layout based on booking class value.

The software client will usually use the xml result from this function to generate the visual seatmap on their client application. You have to render the seatmap in table format base on the value of layout\_row and layout\_column. You can find the sample SetMap XML in **Appendix 10**.

Below is the code sample of how to call the function.

#### Sample

```
Private void GetSeatMapLayout ()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
```





```
{  
  
    //Call Function.  
  
    string strResult = objService.GetSeatMapLayout("KIX", "FUK",  
  
                                                    gFlightId, "",  
  
                                                    "");  
  
}  
  
}
```



#### 4.2.26 SelectSeat()

Name	SelectSeat()	
Description	This function used for select seat for passenger in the booking.	
Output	Error XML	
Parameter (SeatRequest)		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
BookingSegmentId	Guid	Flight Segment Id.
PassengerId	Guid	Passenger Id.
SeatRow	Int32	Seat Row (number).
SeatColumn	String	Seat Column (alphabet).
SeatFeeRcd	String	Fee Code. <b>(Optional)</b>

You can use this function to select seat for the passenger by passing all the required parameter to the function. For parameter **SeatRow** and **SeatColumn** you have to take it from the SeatMap xml, filed name **seat\_row** and **seat\_column**. If you want the function to calculate the fee, you have to take the value of filed fee\_rcd from Seatmap Xml, if present, and pass into parameter **SeatFeeRcd**.

Below is the code sample of how to calculate the function.

#### Sample

```
Private void SelectSeat()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        objService.SeatRequest[] SeatService = new
```



```
objService.SeatRequest[1];

//Create new instance for each request item.

SeatService [0] = new ApiTest.tikApi.SeatRequest();

//Pass in the parameter.

SeatService[0].BookingSegmentId = new

    Guid(bookingSegmentId);

SeatService[0].PassengerId = new Guid(passengerID);

SeatService[0].SeatRow = 10

SeatService[0].SeatColumn = "A";

SeatService[0].SeatFeeRcd = "SEAT";

//Call Function.

string strResult = objService.SelectSeat (SeatService);

}

}
```



#### 4.2.27 GetFeeDefinition()

Name	GetFeeDefinition()	
Description	This function is used for reading fee definition from Avantik fee setting.	
Output	XML Result (See Appendix 12)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
feeRcd	String	Fee Code.
bookingClass	String	Booking Class.
fareCode	String	Fare basis code.
originRcd	String	Airport origin code.
destinationRcd	String	Airport destination code.
flightNumber	String	Flight Number ( <b>Optional</b> )

This function is used for read the fee definition from the Avantik System. This function is call when the client system wants to display fee detail for passenger, for example Credit Card fee.

This function is also used when client system want to add fee information from external source, for example insurance premium. You will see how the function GetFeeDefinition is used for adding fee in the next section, **AddFee()** function. **Sample 23** is the code sample in C# .net on how to call the function.

#### Sample

```
Private void GetFeeDefinition()
{
    MyService objService = (MyService)Session["MyService"];
```



```

If (MyService == null)
{
    //Call Function.

    string strResult = objService.GetFeeDefinition("INSU", "S", "FARECODE1",

                                                "KIX",

                                                "FUK", "");
}
}

```

Please see the XML response structure and field definition in **Appendix 10**.

#### 4.2.28 AddFee()

Name	AddFee()		
Description	This function used for adding fee information into the Booking.		
Output	Error XML (See Appendix 6)		
Parameter (CustomFeeRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
PassengerId	Yes	GUID	GUID of passenger Id.  If this fee is not related to any passenger, please set this field with the Id of the first passenger.
BookingSegmentId	Yes	GUID	GUID of Booking Segment Id.  If this fee is not related to any flight segment, please



			set this field with the Id of the first segment.
FeeId	Yes	GUID	GUID of Fee Id
FeeCategoryRcd	Yes	String	Fee Category code.
FeeRcd	Yes	String	Fee code.
CurrencyRcd	Yes	String	Currency code.
FeeAmount	Yes	Decimal	Fee amount.
FeeAmountIncl	Yes	Decimal	Fee amount + VAT.
VatPercentage	Yes	Decimal	VAT percentage calculate base on fee amount.
ChargeCurrencyRcd	Yes	String	Charge currency other than sale currency.  If single currency is used set the value to be the same as <b>CurrencyRcd</b> .
ChargeAmount	Yes	Decimal	Charge amount in other currency that is not sale currency.  If single currency is used set the value to be the same as <b>FeeAmount</b> .
ChargeAmountIncl	Yes	Decimal	Charge amount + VAT.  If single currency is used set the value to be the same as <b>FeeAmountIncl</b> .
NumberOfUnits	No	Decimal	Number of unit.
Units	No	String	Unit Type.  <b>PCS</b> = Piece.  <b>LBS</b> = Pound.  <b>KGS</b> = Kilogram.
DocumentNumber	No	String	External document number reference. Example will be premium number if it is an insurance fee.
DocumentDateTime	No	DateTime	External document reference date time.
Comment	No	String	Comment text.
ExternalReference	No	String	Open field for External reference.



VendorRcd	No	String	Vender code. Example will be the vender code of Hotel, or car rental.
OdOriginRcd	No	String	First origin airport code.
OdDestinationRcd	No	String	Final destination airport code.

This function is use for adding fee manually to the booking information. Below are the step on how to used AddFee() function.

1. Call **ServiceInitialize()** function.
2. Call **AddFlight()** function.
3. Call **GetFeeDefinition()** function to get the fee definition.
4. Complete the **AddFee()** request parameter with the information get from **GetFeeDefinition()**.



Below is the code sample in .Net C# on how to call the AddFee function

### Sample

```
Private void AddFee()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        //Call GetFeeDefinition Function.

        string strXml = objService.GetFeeDefinition("INSU", "S", "FARECODE1",
                                                    "KIX", "FUK", "");

        objService.CustomFeeRequest[] objFeeRequest = new
        objService.CustomFeeRequest[1];

        XPathDocument xmlDoc = new XPathDocument(new
                                                    System.IO.StringReader(strXml));

        XPathNavigator nv = xmlDoc.CreateNavigator();

        objFeeRequest[0] = new ApiTest.tikApi.CustomFeeRequest();
    }
}
```





```

//Fill Fee information from GetFeeDefinition.

objFeeRequest[0].PassengerId = new Guid(txtPassengerId.Text);

objFeeRequest[0].BookingSegmentId = new Guid(txtBookingSegmentId.Text);

objFeeRequest[0].FeeId = new Guid(nv.SelectSingleNode("Fees/Fee/fee_id").InnerXml);

objFeeRequest[0].FeeCategoryRcd = nv.SelectSingleNode("Fees/Fee/fee_category_rcd").InnerXml;

objFeeRequest[0].FeeRcd = nv.SelectSingleNode("Fees/Fee/fee_rcd").InnerXml;

objFeeRequest[0].CurrencyRcd = nv.SelectSingleNode("Fees/Fee/currency_rcd").InnerXml;

objFeeRequest[0].ChargeCurrencyRcd = nv.SelectSingleNode("Fees/Fee/currency_rcd").InnerXml;

objFeeRequest[0].FeeAmount =
Convert.ToDecimal(nv.SelectSingleNode("Fees/Fee/fee_amount").InnerXml);

objFeeRequest[0].FeeAmountIncl =
Convert.ToDecimal(nv.SelectSingleNode("Fees/Fee/fee_amount_incl").InnerXml);

objFeeRequest[0].VatPercentage =
Convert.ToDecimal(nv.SelectSingleNode("Fees/Fee/vat_percentage").InnerXml);

objFeeRequest[0].ChargeAmount =
Convert.ToDecimal(nv.SelectSingleNode("Fees/Fee/fee_amount").InnerXml);

objFeeRequest[0].ChargeAmountIncl =
Convert.ToDecimal(nv.SelectSingleNode("Fees/Fee/fee_amount_incl").InnerXml);


//Display response result.

string strAddResult = tikApi.AddFee(objFeeRequest);

}

}

```



#### 4.2.29 RemoveFee()

Name	AddFee()		
Description	This function used for remove fee that is added by the AddFee() function.		
Output	Error XML (See Appendix 6)		
Parameter (RemoveFeeRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
BookingFeeld	Yes	GUID	Booking Fee reference Id. You can find this id in Booking XML in section fee. Please see sample XML in <b>Appendix 2</b> .

This function is used for remove fee that is added by the function AddFee(). Below is the code sample on how to remove fee from AddFee() function.

#### Sample

```

Private void RemoveFee()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService == null)
    {
        objService.RemoveFeeRequest[] objFeeRequest = new

        objService.RemoveFeeRequest[1];

        objFeeRequest[0] = new ApiTest.tikApi.RemoveFeeRequest();
    }
}

```



```
//Fill Booking Fee information.
```

```
objFeeRequest[0].BookingFeeld = new Guid(txtBooingFeeld.Text);
```

```
//Call remove function and display result.
```

```
string strAddResult = tikApi.RemoveFee(objFeeRequest);
```

```
}
```

```
}
```



#### 4.2.30 AddRemark()

Name	AddRemark()		
Description	This function used for adding remarks into the Booking.		
Output	Error XML (See Appendix 17)		
Parameter (RemarkRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
ClientProfileId	No	GUID	GUID of Client Profile Id, if the remark is adding by the client profile.
remarkTypeRcd	Yes	String	Remark code that defined in Avantik.
RemarkText	Yes	String	Remark text detail.

This function is use for adding remark to the booking. Below are the step on how to used AddRemarks() function.

1. Call **ServiceInitialize()** function.
2. Call **AddFlight()** function.
3. Call **AddRemark()** function.

Below is the code sample in .Net C# on how to call the AddRemark function

#### Sample

```
Private void AddRemark()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
```



```

Guid gClientProfileId = new Guid(txtClientId.Text);

string strRemarkType = txtRemarkType.Text;

string strRemarktext = txtRemarkText.Text;

//Creat request parameter

tikApi.RemarkRequest[] objRequest = new tikApi.RemarkRequest[1];

objRequest[0] = new ApiTest.tikApi.RemarkRequest();

objRequest[0].ClientProfileId = gClientProfileId; //Optional.

objRequest[0].RemarkTypeRcd = strRemarkType;

objRequest[0].RemarkText = strRemarktext;

//Call Add remark function.

string strResult = MyService.AddRemark(objRequest);

//Validate request result.

.....

}

}

```



#### 4.2.31 RemoveRemark()

Name	RemoveRemark()		
Description	This function used for remove remarks that are added by AddRemark() function call.		
Output	Error XML (See Appendix 6)		
Parameter (RemoveRemarkRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
BookingRemarkId	Yes	GUID	Remark Id Get from BookingGetSession().

This function is use for adding remark to the booking. Below are the step on how to used RemoveRemarks() function.

1. Call **ServiceInitialize()** function.
2. Call **AddFlight()** function.
3. Call **AddRemark()** function.
4. Call **BookingGetSession()** to get a booking\_remark\_id.
5. Call **RemoveRemark()** and passin the booking\_remark\_id to remove the selected remark.



Below is the code sample in .Net C# on how to call the RemoveRemark function

### Sample

```
Private void RemoveRemark()
{
    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        //Take remark Id from BookingGetSession.
        Guid gRemarkId = new Guid(txtBookingRemarkId.Text);

        //Create request parameter.
        tikApi.RemoveRemarkRequest[] objRequest = new
            tikApi.RemoveRemarkRequest[1];

        objRequest[0] = new ApiTest.tikApi.RemoveRemarkRequest();
        objRequest[0].BookingRemarkId = gRemarkId;

        //Call remove remark function.
        string strResult = tikAp.RemoveRemark(objRequest);

        //Validate request result.
        .....
    }
}
```



```
}  
}
```





#### 4.2.32 GetPassenger()

Name	GetPassenger()		
Description	This function will return a list of passenger manifest.		
Output	Passengers XML (See Appendix 14)		
Parameter (PassengerManifestRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
airline	No	String	Airline two letter code.
flightNumber	No	String	Flight Number.
flightFrom	No	DateTime	Departure Date from.
flightTo	No	DateTime	Departure Date to.
recordLocator	No	String	Booking number.
origin	No	String	Origin airport three letter code.
destination	No	String	Destination airport three letter code.
passengerName	No	String	Passenger on flight name.
seatNumber	No	String	Passsneger on flight seat number.
ticketNumber	No	String	Ticket number.
phoneNumber	No	String	Contact detail phone number.
passengerStatus	No	String	Passenger status.
checkInStatus	No	String	Passenger check in status base on Avantik setting.
clientNumber	No	String	Client profile number.
memberNumber	No	String	Member number of the client profile.
listed	No	Boolean	Search for waitlist only.
eTicketOnly	No	Boolean	Search for E-ticket only.
includeCancelled	No	Boolean	Passenger status that include cancel segmemt.
openSegments	No	Boolean	Search for open segment only.
showHistory	No	Boolean	Include passenger booking history.
language	No	String	Two letter language code.



This function is use for getting a list of passenger on flight. All search parameter of GetPassenger function are optional. If none of the parameters pass into API, the API will return top 500 earliest departure date.



Below is the code sample in .Net C# on how to call the GetPassenger function

#### Sample

```
Private void GetPassenger()
{
    //Initialize web service is required.

    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        ApiTest.tikApi.PassengerManifestRequest objRequest = new ApiTest.tikApi. PassengerManifestRequest();

        //Pass in remark as a request parameter.

        objRequest.airline = "";
        objRequest.flightNumber = "";
        objRequest.flightID = Guid.Empty;
        objRequest.flightFrom = DateTime.MinValue;
        objRequest.flightTo = DateTime.MinValue;
        objRequest.recordLocator = "";
        objRequest.origin = "";
        objRequest.destination = "";
```



```
objRequest.passengerName = "";

objRequest.seatNumber = "";

objRequest.ticketNumber = "";

objRequest.phoneNumber = "";

objRequest.passengerStatus = "";

objRequest.checkInStatus = "";

objRequest.clientNumber = "";

objRequest.memberNumber = "";

objRequest.clientID = Guid.Empty;

objRequest.passengerId = Guid.Empty;

objRequest.booked = false;

objRequest.listed = false;

objRequest.eTicketOnly = false;

objRequest.includeCancelled = false;

objRequest.openSegments = false;

objRequest.showHistory = false;

objRequest.language = "EN";

string strResult = tikApi.GetPassenger(objRequest);

.....

}

}
```



### 4.2.33 GetBookings

Name	GetBookings()		
Description	This function will return a list of booking information.		
Output	Result XML (See Appendix 18)		
Parameter			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
airline	No	String	Airline code.
flightNumber	No	String	Flight number.
flightId	Yes	GUID	GUID for flight Id. If no flight id is identify an empty GUID is pass in.
flightFrom	Yes	Date	Start date range of flight departure date.
flightTo	Yes	Date	End date range of flight departure date.
recordLocator	No	String	Booking reference number.
origin	No	String	Airport of origin code.
destination	No	String	Airport destination code.
passengerName	No	String	Passenger name on flight.
seatNumber	No	String	Seat number on flight.
ticketNumber	No	String	Ticket number.
phoneNumber	No	String	Phone number of contact person in the booking.
agencyCode	No	String	Search for agency code that make the booking.
clientNumber	No	String	Client profile number.
memberNumber	No	String	Member number of the client profile in the booking.
clientId	Yes	GUID	Unique client profile ID.
showHistory	Yes	Boolean	If showHistory is set if it is travelagency it will show maximum 20 items, but if it is airline agency it will show 200 items. If this flag is not set it will show 200 maximum on both travel and airline agency.  0 = Not set.  1 = Set.
language	No	String	Two letter language code.



bIndividual	Yes	Boolean	Search only individual booking.
bGroup	Yes	Boolean	Search only group booking.
createFrom	Yes	Date	Start date range of booking create date.
createTo	Yes	Date	End date range of booking creates date.

Below is the code sample in .Net C# on how to call the GetBookings function

### Sample

```

Private void GetBookings()
{
    //Initialize web service is required.
    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        ApiTest.tikApi.BookingGetRequest objRequest = new ApiTest.tikApi.BookingGetRequest();

        //Pass in remark as a request parameter.

        objRequest.airline = "";
        objRequest.flightNumber = "";
        objRequest.flightId = "";
        objRequest.flightFrom = DateTime.MinValue;
        objRequest.flightTo = DateTime.MinValue;
        objRequest.recordLocator = "";
    }
}

```



```
objRequest.origin = "";

objRequest.destination = "";

objRequest.passengerName = "";

objRequest.seatNumber = "";

objRequest.ticketNumber = "";

objRequest.phoneNumber = "";

objRequest.agencyCode = "";

objRequest.clientNumber = "";

objRequest.memberNumber = "";

objRequest.clientId = "";

objRequest.showHistory = false;

objRequest.language = "EN";

objRequest.bIndividual = false;

objRequest.bGroup = false;

objRequest.createFrom = DateTime.MinValue;

objRequest.createTo = DateTime.MinValue;

string strResult = tikApi.GetBookings(objRequest);

.....

}

}
```



#### 4.2.34 GetQuoteSummary

Name	GetQuoteSummary()		
Description	This function will return a list of selected flight fare and taxes summary.		
Output	Result XML (See Appendix 2)		
Parameter			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
Adult	Yes	Int16	Number of adults in the booking.
Child	Yes	Int16	Number of child in the booking.
Infant	Yes	Int16	Number of infant in the booking.
Other	Yes	Int16	Number of other type in the booking.
FlightSegmentRequest	Yes	FlightSegmentRequest	Flight information request parameter. Please see table <b>FlightSegmentRequest()</b> .
AgencyCode	Yes	String	Airline or travel agency code.
Language	Yes	String	Language code.
CurrencyCode	Yes	String	Currency code.
NoVat	Yes	Boolean	Identify if VAT should be calculate with in the summary  <b>True</b> = “No VAT should be calculate”.  <b>False</b> = “VAT should be calculate”.
Password	Yes	String	Airline or agency admin password.
OtherPassengerType	Yes	String	Name of other passenger type ( if number of other type is greater than zero).





Name	FlightSegmentRequest()		
Description	Flight information request paramater.		
Parameter			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
FlightId	Yes	Guid	Flight ID.
FareId	Yes	Guid	Fare ID.
OriginRcd	Yes	String	Airport origin code.
DestinationRcd	Yes	String	Airport destination code.
BoardingClassRcd	No	String	Boarding class code.
BookingClassRcd	No	String	Booking class code.
DepartureDate	Yes	Date	Flight departure date.
TransitDepartureDate	No	Date	Transit flight departure date.
TransitFlightId	No	Guid	Transit flight ID.
TransitAirportRcd	No	String	Transit airport code.
TransitBoardingClassRcd	No	String	Transit boarding class code.
TransitBookingClassRcd	No	String	Transit booking class code.

GetQuoteSummary function will return a list of financial summary of the selected flight. The request parameter of function GetQuoteSummary can be getting from response information of FlightAvailability function. ServiceInitialize is not required for this function call. Please see **Appendix 2** for the field description of GetQuoteSummary response.

#### Sample SOAP request message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
```



```

<tem:GetQuoteSummary>

  <tem:getQuoteRequest>

    <tem:Adult>1</tem:Adult>

    <tem:Child>0</tem:Child>

    <tem:Infant>0</tem:Infant>

    <tem:FlightSegmentRequest>

      <tem:FlightId>24849482-3322-41B3-861C-7DEBFAA42414</tem:FlightId>

      <tem:FareId>24849482-3322-41B3-861C-7DEBFAA42414</tem:FareId>

      <tem:OriginRcd>KIX</tem:OriginRcd>

      <tem:DestinationRcd>FUK</tem:DestinationRcd>

      <tem:BoardingClassRcd>Y</tem:BoardingClassRcd>

      <tem:BookingClassRcd>Y</tem:BookingClassRcd>

      <tem:DepartureDate>2015-01-20</tem:DepartureDate>

      <tem:TransitDepartureDate>2015-01-20</tem:TransitDepartureDate>

      <tem:TransitFlightId>24849482-3322-41B3-861C-7DEBFAA42515</tem:TransitFlightId>

      <tem:TransitAirportRcd>OKD</tem:TransitAirportRcd>

      <tem:TransitBoardingClassRcd>Y</tem:TransitBoardingClassRcd>

      <tem:TransitBookingClassRcd>Y</tem:TransitBookingClassRcd>

    </tem:FlightSegmentRequest>

    <tem:AgencyCode>B2C</tem:AgencyCode>

    <tem:Language>EN</tem:Language>

    <tem:CurrencyCode>JPY</tem:CurrencyCode>

    <tem:NoVat>false</tem:NoVat>

    <tem>Password>123456</tem>Password>

  </tem:getQuoteRequest>

```



```

    </tem:GetQuoteSummary>
  </soapenv:Body>
</soapenv:Envelope>

```



#### 4.2.35 GetInfantCapacity()

This function will return the XML information described the number of available infant for the flight following sample XML below. User can retrieve XML response using for validate number of infant input and display response message as required.

Name	GetInfantCapacity	
Description	Get infant capacity available.	
Parameter		
<u>Name</u>	<u>Description</u>	<u>Required</u>
flightId	Flight ID.	Yes
originRcd	Original airport.	Yes
destinationRcd	Destination airport.	Yes
boardingClassRcd	Board class code.	Yes

#### Sample Response XML

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <GetInfantCapacityResponse xmlns="http://tempuri.org/">

      <GetInfantCapacityResult>5</GetInfantCapacityResult>

    </GetInfantCapacityResponse>

  </soap:Body>

</soap:Envelope>
```



#### 4.2.36 GetCoperateSessionProfile

This function will return coporate client profile and employee under that coporate following request parameter by calling function GetCoperateSessionProfile ().

Name	GetCoperateSessionProfile ()	
Description	Function returns a list of coporate client profile .	
Output	Active booking XML (See Appendix 20)	
Parameter		
<u>Name</u>	<u>Data Type</u>	<u>Description</u>
Agency	String	Agency code

#### Sample

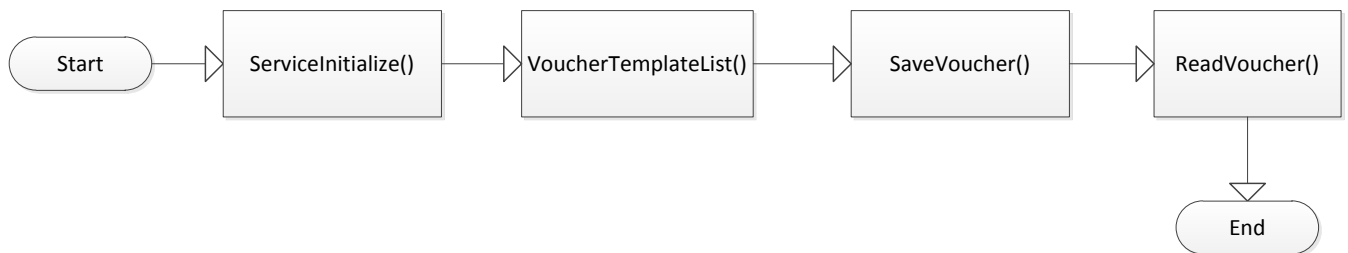
```
<soap:Body>
  <tem:GetCoperateSessionProfile>
    <tem:agency>B2EAAS</tem:agency>
  </tem:GetCoperateSessionProfile>
</soap:Body>
```



### 4.3 Purchase Voucher API

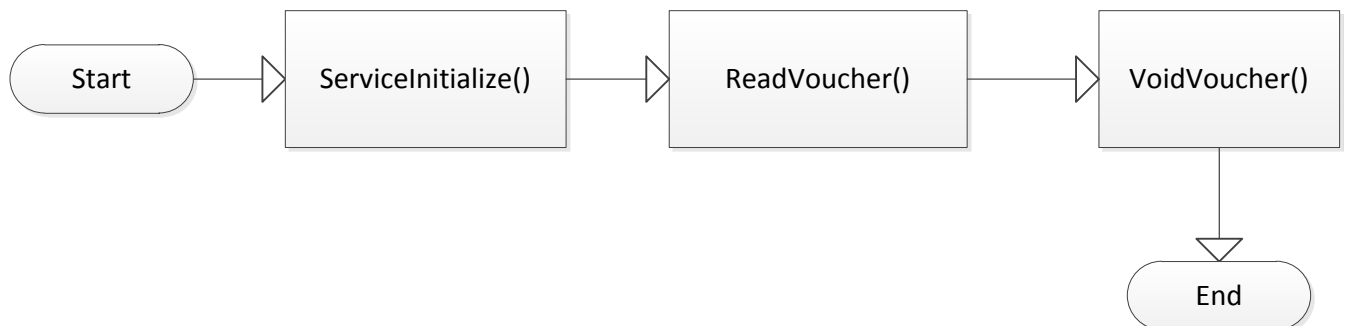
The Purchase voucher payment API is an API interface that allowed third party to make a request for purchase voucher through Avantik system.

Below is the function flow of purchase voucher.



1. API **ServiceInitialize()** function need to be call for user authentication.
2. Call function **VoucherTemplateList()** to get voucher template information.
3. Programmer take all the information from the voucher template XML that return from VoucherTemplateList() function and pass to **SaveVoucher()** function to purchase voucher.
4. SaveVoucher() function will return the a success XML result with a reference ID back as a response. Programmer have to use the reference ID to read the created voucher from **ReadVoucher()** function for display and emailing purpose.

Below is the function flow of void voucher.



1. API **ServiceInitialize()** function need to be call for user authentication.
2. Call **ReadVoucher()** function by passin Voucher Number to retrieve the voucher information that you want to void.
3. Take voucher ID from the XML response of ReadVoucher() and pass into the function **VoidVoucher()** to void the selected voucher.



#### 4.3.1 VoucherTemplateList

Name	VoucherTemplateList()		
Description	This function is use for getting a list of voucher template.		
Output	Voucher template list XML (See Appendix 13 )		
Parameter (VoucherTemplateListRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
VoucherTemplateId	No	GUID	GUID of voucher template ID.
VoucherTemplate	No	String	Display name of voucher template.
FromDate	No	Date	Flight from date.
ToDate	No	Date	Flight to date.
Status	NO	String	Voucher template status code.

This function is use for getting a list of voucher template. Voucher template is standard voucher information that will be use when **SaveVoucher()** function is call. Service Initialize is required to used VoucherTemplateList() function.

Below is the code sample in .Net C# on how to call the VoucherTemplateList function

#### Sample





```

Private void ReadVoucherTemplate()

{
    //Initialize web service is required.

    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        //Declare request parameter variable.

        ApiTest.tikApi.VoucherTemplateListRequest objVTListRequest = new
        ApiTest.tikApi.VoucherTemplateListRequest();

        //Assign request parameter.

        objVTListRequest.VoucherTemplateId = txtTemplateID.Text.Trim();
        objVTListRequest.VoucherTemplate = txtTemplateName.Text.Trim();
        objVTListRequest.Status = "A";
        if(!String.IsNullOrEmpty( txtToDate.Text))
        {
            objVTListRequest.ToDate = Convert.ToDateTime( txtToDate.Text.Trim());
        }

        if (!String.IsNullOrEmpty(txtFromDate.Text))
        {
            objVTListRequest.FromDate = Convert.ToDateTime(txtFromDate.Text.Trim());
        }
    }
}

```



```

    }

    //Request for template list.
    string strTemplateListResult = objService.VoucherTemplateList(objVTListRequest);

    .....
}
}

```



### 4.3.2 SaveVoucher

Name		SaveVoucher()	
Description		Function use for purchase voucher.	
Output		Error XML (See Appendix 14)	
Parameter (SaveVoucherRequest)			
Parameter (VoucherTemplateRequest)			
Name	Required	Data Type	Description
voucher_template_id	Yes	GUID	ID use to identify individual template.
currency_rcd	Yes	String	Currency code.
valid_to_date	No	DateTime	Voucher valid until.
valid_from_date	No	String	Voucher valid from date.
display_name	No	String	Voucher display name.
status_code	Yes	String	Voucher status.  A = Active  I = Inactive  D = Delete.
Origins	No	String	The airport origin that this template can use.
Destinations	No	String	The airport destnation that this template can use.
passenger_segments	Yes	Short	Number of passenger in segment that can use voucher.  Default Value is 0.
voucher_use_code	Yes	String	P = Percentage  A = Amount  *P Still not implements.
recipient_only_flag	Yes	Byte	0 = Any recipient can use voucher.



			<p><b>1</b> = Only specific recipient name can be use.</p> <p>This flag is use with recipient_name if the value is set to 1.</p>
discount_percentage	Yes	Decimal	*Not implements. Default as 0.
valid_for_class	No	String	Identify the booking class that this voucher allowed to use.
ticket_flag	Yes	Byte	<p><b>0</b> = Can't use for ticket payment.</p> <p><b>1</b> = Can use for ticket payment.</p>
seat_fee_flag	Yes	Byte	<p><b>0</b> = Can't make payment with seat fee.</p> <p><b>1</b>= Can use for seat fee payment.</p>
other_fee_flag	Yes	Byte	<p><b>0</b> =Can't make payment with other fee.</p> <p><b>1</b>= Can make payment with other fee.</p>
b2c_flag	Yes	Byte	<p><b>0</b> = Not allow to use with B2C.</p> <p><b>1</b> = Allowed to use with B2C.</p>
b2b_flag	Yes	Byte	<p><b>0</b> = Not allow to use with B2B.</p> <p><b>1</b> = Allowed to use with B2B.</p>
b2e_flag	Yes	Byte	<p><b>0</b> = Not allow to use with B2E.</p> <p><b>1</b> = Allowed to use with B2E.</p>
airline_flag	Yes	Byte	<p><b>0</b> = Not allow to use with airline agency.</p> <p><b>1</b> = Allowed to use with airline agency.</p>
fare_only_flag	Yes	Byte	<p><b>0</b> = Allow only payment for both fare and fee.</p> <p><b>1</b> = Allow only payment for fare only.</p>



voucher_value	Yes	Decimal	Value of voucher
charge_amount	Yes	Decimal	Voucher charge amount.
multiple_payment_flag	No	Byte	Flag identify that this voucher can use multiple time.  0 = One time use.  1 = Can use multiple time.
expiry_date_time	No	Datetime	Voucher expiry date. This field have to be calculate from the valid_days field that return from VoucherTemplateList() function.
refundable_flag	Yes	Byte	* Not implement. Set as 0.
percentage_flag	Yes	Byte	* Not implement. Set as 0.
single_flight_flag	Yes	Byte	* Not implement. Set as 0.
single_passenger_flag	Yes	Byte	* Not implement. Set as 0.
recipient_name	No	String	Name of the recipient.
voucher_password	No	String	Voucher password.
form_of_payment_rcd	No	String	Form of payment code. Usually set as "VOUCHER"
form_of_payment_subtype_rcd	No	String	Form of payment subtype code.
<b>Parameter (PaymentRequest)</b>			
<b><u>Name</u></b>	<b><u>Required</u></b>	<b><u>Data Type</u></b>	<b><u>Description</u></b>
form_of_payment_rcd	Yes	String	Form of payment code:  CC = credit card  INV = invoice  CRAG = credit agency account  VOUCHER = voucher
form_of_payment_subtype_rcd	Yes	String	Form of payment subtype code. The code is based on form of payment



			subtype settings in Avantik.
payment_amount	Yes	Decimal	Payment Amount.
currency_rcd	Yes	String	Payment currency.
NameOnCard	Yes	String	Name on card.
CreditCardNumber	Yes	String	Credit card number
IssueMonth	No	Int	Credit card issue month.
IssueYear	No	Int	Credit card issue year.
IssueNumber	No	Int	Credit card issue number.
ExpiryMonth	No	Int	Credit card expiry month.
ExpiryYear	No	Int	Credit card expiry year.
Addr1	No	String	Address line1
Addr2	No	String	Address line 2
Street	No	String	Street
State	No	String	State
City	No	String	City
County	No	String	Country
ZipCode	No	String	Post code.
Country	No	String	Country
CCV	No	String	Credit card CVV code.

The function is used for create voucher. There are two type of request for SaveVoucher() function.

1. Purchase voucher

- a. For purchase voucher, both **VoucherTemplateRequest** and **PaymentRequest** parameters have to be passin.
- b. Only credit card payment is allowed for purchase voucher.

2. Create voucher

- a. Create voucher required only **VoucherTemplateRequest** parameter.



**VoucherTemplateList()** function call is required before **SaveVoucher()** call. Information that pass to **VoucherTemplateRequest** parameter need to be taking from voucher template.

The success request will return response XML with code "000". The response XML also includes a reference ID. You can use this reference ID as an input parameter of **ReadVoucher()** function to get the voucher detail.

Below is the code sample in .Net C# on how to call the SaveVoucher function

### Sample

```
Private void SaveVoucher()
{
    //Initialize web service is required.

    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        DateTime dtStartDate = DateTime.Now;

        tikApi.BookingService tikApi = (tikApi.BookingService)Session["ApiService"];

        ApiTest.tikApi.SaveVoucherRequest objSaveRequest = new ApiTest.tikApi.SaveVoucherRequest();

        ApiTest.tikApi.VoucherPaymentRequest pmr = new tikApi.VoucherPaymentRequest();

        ApiTest.tikApi.VoucherTemplateRequest vcr = new ApiTest.tikApi.VoucherTemplateRequest();
```



```
//Request parameter for VoucherPaymentRequest.
```

```
pmr.booking_payment_id = Guid.NewGuid();

pmr.currency_rcd = "JPY";

pmr.payment_amount = (decimal)100;

pmr.form_of_payment_rcd = "CC";

pmr.form_of_payment_subtype_rcd = "VISA";

pmr.NameOnCard = "Angkoon Jiriyasin";

pmr.CreditCardNumber = txtCardNo.Text.Trim();

pmr.IssueMonth = "";

pmr.IssueYear = "";

pmr.IssueNumber = "";

pmr.ExpiryMonth = "03";

pmr.ExpiryYear = "2017";

pmr.Addr1 = "1009/67";

pmr.Addr1 = "Thailand";

pmr.Street = "Suhumvit 71";

pmr.State = "";

pmr.City = "Bangkok";

pmr.ZipCode = "10110";

pmr.Country = "TH";

pmr.CCV = "111";

objSaveRequest.Payment = pmr;
```





```
//Request parameter for VoucherTemplate request.

vcr.voucher_template_id = new Guid("8c109add-f0b3-4e9f-9218-aeff2edfc297");

vcr.form_of_payment_rcd = "VOUCHER";

vcr.form_of_payment_subtype_rcd = "VOUCHER";

vcr.currency_rcd = "JPY";

vcr.valid_to_date = Convert.ToDateTime("2014-02-28");

vcr.valid_days = 365;

vcr.valid_from_date = Convert.ToDateTime("2012-04-25");

vcr.display_name = "Free voucher All";

vcr.status_code = "A";

vcr.origins = "KIX";

vcr.destinations = "FUK";

vcr.passenger_segments = 2;

vcr.voucher_use_code = "A";

vcr.recipient_only_flag = 1;

vcr.discount_percentage = 10;

vcr.valid_for_class = "FCY";

vcr.ticket_flag = 1;

vcr.seat_fee_flag = 1;

vcr.other_fee_flag = 1;

vcr.b2c_flag = 1;

vcr.b2b_flag = 1;
```



```
vcr.b2e_flag = 1;

vcr.airline_flag = 1;

vcr.fare_only_flag = 1;

vcr.voucher_value = 100000;

vcr.charge_amount = 100;

vcr.multiple_payment_flag = 1;

vcr.expiry_date_time = DateTime.Now.AddYears(1);

vcr.percentage_flag = 1;

vcr.recipient_name = "Name Lastname";

vcr.refundable_flag = 1;

vcr.single_flight_flag = 1;

vcr.single_passenger_flag = 1;

objSaveRequest.Voucher = vcr;

string strSaveVoucherResult = tikApi.SaveVoucher(objSaveRequest);

.....

}

}
```



### 4.3.3 ReadVoucher

Name	ReadVoucher()		
Description	This function is use for search voucher in the Avantik system.		
Output	Voucher XML (See Appendix 14 )		
Parameter (VoucherTemplateListRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
VoucherId	No	GUID	GUID of voucher ID.
VoucherNumber	No	Double	Voucher Number.

ReadVoucher() function is use for reading voucher information from Avantik system.

For the purchase voucher process, ReadVoucher() function is used after SaveVoucher() function. Programmer can use a reference ID from the SaveVoucher() XML response and pass into VoucherId parameter of ReadVoucher() function to get a detail created voucher XML for use as a display or email to customer.

#### Sample

```

Private void ReadVoucher ()
{
    //Initialize web service is required.

    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        ApiTest.tikApi.VoucherReadRequest objReadVoucherRequest = new ApiTest.tikApi.VoucherReadRequest();

        //Assign Request parameter.
    }
}

```



```

if (!String.IsNullOrEmpty(txtVoucherID.Text) )
{
    objReadVoucherRequest.VoucherId = new Guid(txtVoucherID.Text.Trim());
}

if (!String.IsNullOrEmpty(txtVoucherNumber1.Text))
{
    objReadVoucherRequest.VoucherNumber =

        Convert.ToDouble(txtVoucherNumber1.Text.Trim());
}

//Request for template list.

string strReadVoucherResult = tikApi.ReadVoucher(objReadVoucherRequest);

.....
}
}

```

#### 4.3.4 VoidVoucher

Name	VoidVoucher()
Description	This function is use for void existing voucher in the Avantik system.
Output	Error XML (See Appendix 6)
Parameter (VoucherTemplateListRequest)	



<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
VoucherId	Yes	GUID	GUID of voucher ID.

Below is the code sample in .Net C# on how to call the VoidVoucher function

### Sample

```

Private void VoidVoucher()
{
    //Initialize web service is required.

    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        ApiTest.tikApi.VoucherVoidRequest objVoidVoucherRequest = new
            ApiTest.tikApi.VoucherVoidRequest();

        //Pass in voucher id as a request parameter.

        objVoidVoucherRequest.VoucherId = new Guid(txtVoucherID2.Text.Trim());

        string strVoidVoucherResult = tikApi.VoidVoucher(objVoidVoucherRequest);

        .....
    }
}

```





## 4.4 Manage Booking API

The Manage booking API is an API interface that allowed third party to make a request for updating or make any change to the existing passenger booking.

All function in manage booking API required ServiceInitialize call. Please refer to section **4.2.1 ServiceInitialize**.

### 4.4.1 AddRemarkToBooking

Name	AddRemarkToBooking()		
Description	This function is use for adding remark to existing booking.		
Output	Error XML (See Appendix 6)		
Parameter (RemarkRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
BookingId	Yes	GUID	GUID of Booking ID.
BookingRemarkId	Yes	GUID	GUID of Booking remark Id. This ID is use to identify individual remark. If empty GUID is passin system will auto generate this id.
ClientProfileId	Yes	GUID	GUID of Client profile Id. This Id is use to identify the client profile that add the remark. If no client profile identify an empty GUID is pass in.
RemarkTypeRcd	Yes	String	Remark type code. Developer can get the code from AVANTIK setting. This filed is use to identify type of the remark.
RemarkText	Yes	String	This filed is use for remark description.

Below is the code sample in .Net C# on how to call the AddRemarkToBooking function

#### Sample

```
Private void AddRemarkToBooking()
```



```
{

//Initialize web service is required.

MyService objService = (MyService)Session["MyService"];

If (MyService != null)

{

    ApiTest.tikApi.RemarkRequest objRequest = new ApiTest.tikApi.RemarkRequest();

    //Pass in remark as a request parameter.


    objRequest.BookingId = new Guid(txtBookingId.Text);

    if (string.IsNullOrEmpty(txtClientProfileID.Text) == false)

    {

        objRequest.ClientProfileId = new Guid(txtClientProfileID.Text);

    }

    else

    {

        //Empty Guid pass in when client profile is not identify.

        objRequest.ClientProfileId = 00000000-0000-0000-0000-000000000000;

    }

    objRequest.RemarkTypeRcd = txtRemarkType.Text.Trim();

}
```





```
objRequest.RemarkText = txtRemarkText.Text.Trim();  
  
string strResult = tikApi.AddRemarkToBooking(objRequest);  
  
.....  
}  
  
}
```



#### 4.4.2 DeleteRemarkFromBooking

Name	DeleteRemarkFromBooking()		
Description	This function is use for delete remark from existing booking.		
Output	Error XML (See Appendix 6)		
Parameter (RemoveRemarkRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
BookingRemarkId	Yes	GUID	GUID of Booking remark Id. This ID is use to identify individual remark. Empty GUID is not allowed.

Below is the code sample in .Net C# on how to call the DeleteRemarkFromBooking function

#### Sample

```
Private void DeleteRemarkFromBooking()
{
    //Initialize web service is required.

    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        ApiTest.tikApi.RemarkRequest objRequest = new ApiTest.tikApi.RemarkRequest();

        //Pass in remark as a request parameter.

        objRequest.BookingRemarkId = new Guid(txtBookingRemarkId.Text);

        string strResult = tikApi.DeleteRemarkFromBooking(objRequest);
    }
}
```



```

    .....
}
}

```



Developer need to follow below step to delete remark from existing booking.

1. Call ServiceInitialize Function. Developer can find information of ServiceInitialize in section 4.2.1.
2. Call BookingGetItinerary function. This function will return a booking xml which include remark information. Please refer to Appendix 3 Itinerary XML
3. Get value of **booking\_remark\_id** from remark section and use it as a parameter for DeleteRemarkFromBooking function.
4. Make a request to DeleteRemarkFromBooking function and pass in a request parameter to delete the item.

#### 4.4.3 UpdateBookingRemark

Name	UpdateBookingRemark()		
Description	This function is use for editing a remark item of the booking.		
Output	Error XML (See Appendix 6)		
Parameter (RemarkRequest[])			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
BookingId	Yes	GUID	GUID of Booking ID.
BookingRemarkId	Yes	GUID	GUID of Booking remark Id. This ID is use to identify individual remark. If empty GUID is not allowed for this request.
ClientProfileId	Yes	GUID	GUID of Client profile Id. This Id is use to identify the client profile that add the remark. If no client profile identify an empty GUID is pass in.
RemarkTypeRcd	Yes	String	Remark type code. Developer can get the code from AVANTIK setting. This filed is use to identify type of the remark.
RemarkText	Yes	String	This filed is use for remark description.

Below is the code sample in .Net C# on how to call the UpdateBookingRemark function

#### Sample



```
Private void UpdateBookingRemark()
{
    //Initialize web service is required.
    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        ApiTest.tikApi.RemarkRequest[] obRequest = new ApiTest.tikApi.RemarkRequest[1];

        //Pass in remark as a request parameter.
        objRequest[0] = new ApiTest.tikApi.RemarkRequest();
        objRequest[0].BookingId = new Guid(txtBookingId.Text);
        objRequest[0].BookingRemarkId = new Guid(txtBookingRemarkId.Text);

        if (string.IsNullOrEmpty(txtClienProfileID.Text) == false)
        {
            objRequest[0].ClientProfileId = new Guid(txtClienProfileID.Text);
        }
        else
        {
            //Empty Guid pass in when client profile is not identify.
            objRequest[0].ClientProfileId = 00000000-0000-0000-0000-000000000000;
        }
    }
}
```



```
objRequest[0].RemarkTypeRcd = txtRemarkType.Text.Trim();

objRequest[0].RemarkText = txtRemarkText.Text.Trim();

string strResult = tikApi.UpdateBookingRemark(objRequest);

.....

}

}
```

Developer need to follow below step for update remark item(s) in the booking.

1. Call ServiceInitialize Function. Developer can find information of ServiceInitialize in section 4.2.1.
2. Call BookingGetItinerary function. This function will return a booking xml which include remark information. Please refer to Appendix 3 Itinerary XML
3. Get value of **booking\_remark\_id** from remark section and use it as a parameter for DeleteRemarkFromBooking function.
4. Get value of **booking\_id** from BookingHeader section and use it as a parameter for DeleteRemarkFromBooking function.
5. Make change to the field **RemarkTypeRcd** or **RemarkText**.
6. Make a request to UpdateBookingRemark function and pass in a request parameter to delete the item.



#### 4.4.4 GetActivities

Name	GetActivities()		
Description	This function is use for getting a list of activities remark.		
Output	Result XML (See Appendix 16)		
Parameter (ActivityGetRequestMessage)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
RemarkType	No	String	Remark type code.
Nickname	No	String	Search for nickname of remark that defines in Avantik.
TimelimitFrom	Yes	Date	Start date range for search ticket timelimit.
TimelimitTo	Yes	Date	End date range for search ticket timelimit.
PendingOnly	Yes	Boolean	Show only pending remarks.
IncompleteOnly	Yes	Boolean	Show only incomplete remarks.
IncludeRemarks	Yes	Boolean	Show remark that are not in queue.
ShowUnassigned	Yes	Boolean	Show remark that are not assign to agency.

Below is the code sample in .Net C# on how to call the GetActivities function

#### Sample

```

Private void GetActivity()
{
    //Initialize web service is required.

    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {

```



```
ApiTest.tikApi.RemarkRequest objRequest = new ApiTest.tikApi.RemarkRequest();
```

```
//Pass in remark as a request parameter.
```

```
objRequest.RemarkType = string.Empty;
```

```
objRequest.Nickname = "";
```

```
objRequest.TimelimitFrom = DateTime.MinValue;
```

```
objRequest.TimelimitTo = DateTime.MinValue;
```

```
objRequest.PendingOnly = false;
```

```
objRequest.IncompleteOnly = true;
```

```
objRequest.IncludeRemarks = false;
```

```
objRequest.showUnassigned = false;
```

```
string xmlResult = tikApi.GetActivities(objRequest);
```

```
.....
```

```
}
```

```
}
```





#### 4.4.5 CompleteRemark

Name	CompleteRemark()		
Description	This function is use for complete activity remark.		
Output	Error XML (See Appendix 6)		
Parameter (CompleteRemarkRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
BookingRemarkId	Yes	GUID	GUID of Booking remark Id. This ID is use to identify individual remark. Empty GUID is not allowed.

Below is the code sample in .Net C# on how to call the CompleteRemark function

#### Sample

```
Private void CompleteRemark()
{
    //Initialize web service is required.
    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        ApiTest.tikApi.CompleteRemarkRequest objRequest = new ApiTest.tikApi.CompleteRemarkRequest();

        //Pass in remark as a request parameter.
        objRequest.BookingRemarkId = new Guid(txtBookingRemarkId.Text);

        string strResult = tikApi.CompleteRemark(objRequest);
    }
}
```



```

.....
}
}

```

Developer need to follow below step to delete remark from existing booking.

1. Call ServiceInitialize Function. Developer can find information of ServiceInitialize in section 4.2.1.
2. Call GetActivities function. This function will return an activity remark xml. Please refer to section 4.4.4 for GetActivities function call.
3. Get value of **booking\_remark\_id** from the xml and use it as a parameter for CompleteRemarkRequest function.
4. Make a request to CompleteRemarkRequest function and pass in a request parameter to complete the activity.



#### 4.4.6 GetQueueCount

Name	GetQueueCount()		
Description	Count the pending remark by category and agency.		
Output	Result XML (See Appendix 16)		
Parameter			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
unassigned	Yes	Boolean	Search for the remark that are not assign to agency.  0 = Search for remark that already assign to agency.  1 = Search for remark that are not assign to agency.

Below is the code sample in .Net C# on how to call the GetQueueCount function

#### Sample

```

Private void GetQueueCount()
{
    //Initialize web service is required.

    MyService objService = (MyService)Session["MyService"];

    If (MyService != null)
    {
        //Pass in remark as a request parameter.

        string strResult = tikApi.GetQueueCount(true);

        .....
    }
}

```



```
}  
}
```



## 4.5 Manage client profile

The Manage client profile API is an API interface that allowed third party to make a request for adding and updating passenger profile.

All function in manage booking API required ServiceInitialize call. Please refer to section **4.2.1 ServiceInitialize**.

### 4.5.1 CreateClientProfile

Name	CreateClientProfile()		
Description	This function is use for creating the client profile in system.		
Output	Error XML (See Appendix 6)		
Parameter (CreateClientProfileRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
title_rcd	Y	String	Client profile title code.
Lastname	Y	String	Client profile lastname.
Firstname	Y	String	Client profile firstname.
Middlename	N	String	Client middlename.
language_rcd	Y	String	Language code.
nationality_rcd	N	String	Nationality country code.
passenger_weight	Y	Decimal	Passenger weight.  Default Value: 0
gender_type_rcd	Y	String	Gender type code.  M = Male  F = Female
passenger_type_rcd	Y	String	Passenger type code:  ADULT = Adult  CHD = Child



address_line1	N	String	Address line 1.
address_line2	N	String	Address line 2.
State	N	String	State.
zip_code	N	String	Postal code.
country_rcd	N	String	Country code.
Street	N	String	Street.
City	N	String	City.
document_type_rcd	N	String	Document type code.  P = Passport.  B = Birth Certificate  DL = Driving License  I = ID Card  M = US Military ID  R = Refugee  V = Visa
passport_number	N	String	Document number.
passport_issue_date	Y	Date	Passport issue date.  <b>Default Value:</b> 0001-01-01
passport_expiry_date	Y	Date	Passport expiry date.  <b>Default Value:</b> 0001-01-01
passport_issue_place	N	String	Passport issue place.
passport_birth_place	N	String	Passport birth place.
date_of_birth	Y	String	Date of birth.
passport_issue_country_rcd	N	String	Passport issuing country.
contact_email	Y	String	Contact email.



phone_mobile	N	String	Mobile phone number.
phone_home	N	String	Home phone.
phone_business	N	String	Business phone.
wheelchair_flag	Y	Byte	Identify whether the member required wheelchair.  <b>0</b> = not required.  <b>1</b> = required.
vip_flag	Y	Byte	Identify whether the client profile is a VIP member.
member_level_rcd	N	String	FFP member level. Base on Avantik setting.
window_seat_flag	Y	Byte	Identify that the member is prefer to set close to the window.  <b>0</b> = Can be any place.  <b>1</b> = Prefer.
redress_number	N	String	Department of Homeland Security's Traveler Redress Inquiry Program (TRIP).
status_code	Y	String	Client profile status code.  A = Active  I = Inactive  D = Delete.
client_password	Y	String	Client profile password.
company_flag	Y	Byte	Identify if this client profile is a B2E profile.  <b>Default Value: 0</b>
client_type_rcd	Y	String	Client type code.  <b>COR</b> = Corporate profile. User of B2E.



			<b>IND</b> = Individual profile. For passenger client profile.  <b>STF</b> = Staff passenger.
member_since_date	Y	Date	Member since date.

When CreateClientProfile function is call, API will save main client profile information with one default passenger profile with role "MYSELF" in the Avantik System. The response of the function include the ID of the client profile that will be use for reading client profile by calling function PassengerProfileRead (Section 4.5.1). Please see **Appendix 6** for the response sample with the reference ID.

Below is the step of how the function is use together with PassengerProfileRead function.

1. User fills in their information.
2. When user click submit button, system call CreateClientProfile to save client profile information.
3. System takes client\_profile\_id in the reference filed of CreateClientProfile response and use as request parameter for function PassengerProfileRead.
4. System use response of PassengerProfileRead to display client profile information, example client number and password for user to use as their login.

### Sample SOAP request message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:CreateClientProfile>
      <tem:ClientRequest>
        <tem:title_rcd>MR</tem:title_rcd>
      </tem:ClientRequest>
    </tem:CreateClientProfile>
  </soapenv:Body>
</soapenv:Envelope>
```





```

<tem:Lastname>TESTLASTNAME</tem:Lastname>

<tem:Firstname>TESTFIRSTNAME</tem:Firstname>

<tem:Middlename></tem:Middlename>

<tem:language_rcd>EN</tem:language_rcd>

<tem:nationality_rcd>JP</tem:nationality_rcd>

<tem:passenger_weight>0</tem:passenger_weight>

<tem:gender_type_rcd>M</tem:gender_type_rcd>

<tem:passenger_type_rcd>ADULT</tem:passenger_type_rcd>

<tem:address_line1>Address line 1</tem:address_line1>

<tem:address_line2>Address line 2</tem:address_line2>

<tem:State>State</tem:State>

<tem:District>District</tem:District>

<tem:Province>Province</tem:Province>

<tem:zip_code>1111</tem:zip_code>

<tem:po_box>2222</tem:po_box>

<tem:country_rcd>JP</tem:country_rcd>

<tem:Street>Street</tem:Street>

<tem:City>City</tem:City>

<tem:document_type_rcd>P</tem:document_type_rcd>

<tem:residence_country_rcd>JP</tem:residence_country_rcd>

<tem:passport_number>P12345</tem:passport_number>

<tem:passport_issue_date>2014-01-01</tem:passport_issue_date>

<tem:passport_expiry_date>2015-01-01</tem:passport_expiry_date>

<tem:passport_issue_place>Japan</tem:passport_issue_place>

<tem:passport_birth_place>Tokyo</tem:passport_birth_place>

```



```

<tem:date_of_birth>1977-12-12</tem:date_of_birth>

<tem:passport_issue_country_rcd>JP</tem:passport_issue_country_rcd>

<tem:contact_name>MR TESTLASTNAME TESTFIRSTNAME</tem:contact_name>

<tem:contact_email>email@test.com</tem:contact_email>

<tem:mobile_email>mobile@test.com</tem:mobile_email>

<tem:phone_mobile>0212521251</tem:phone_mobile>

<tem:phone_home></tem:phone_home>

<tem:phone_fax></tem:phone_fax>

<tem:phone_business></tem:phone_business>

<tem:employee_number></tem:employee_number>

<tem:wheelchair_flag>0</tem:wheelchair_flag>

<tem:vip_flag>0</tem:vip_flag>

<tem:member_level_rcd></tem:member_level_rcd>

<tem:member_number></tem:member_number>

<tem>window_seat_flag>0</tem>window_seat_flag>

<tem:redress_number></tem:redress_number>

<tem:status_code>A</tem:status_code>

<tem:client_password>password123</tem:client_password>

<tem:company_flag>0</tem:company_flag>

<tem:client_type_rcd>IND</tem:client_type_rcd>

<tem:member_since_date>2010-02-02</tem:member_since_date>

</tem:ClientRequest>

</tem:CreateClientProfile>

</soapenv:Body>

</soapenv:Envelope>

```



#### 4.5.2 EditClientProfile

Name	EditClientProfile()		
Description	This function is use for changing client profile information.		
Output	Error XML (See Appendix 6)		
Parameter (EditClientProfileRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
client_profile_id	Y	Guid	Key use to identify individual client profile.
title_rcd	Y	String	Client profile title code.
lastname	Y	String	Client profile lastname.
firstname	Y	String	Client profile firstname.
middlename	N	String	Client middlename.
language_rcd	Y	String	Language code.
nationality_rcd	N	String	Nationality country code.
passenger_weight	Y	Decimal	Passenger weight.  Default Value: 0
gender_type_rcd	Y	String	Gender type code.  M = Male  F = Female
passenger_type_rcd	Y	String	Passenger type code:  ADULT = Adult  CHD = Child
address_line1	N	String	Address line 1.
address_line2	N	String	Address line 2.
State	N	String	State.
zip_code	N	String	Postal code.
country_rcd	N	String	Country code.
Street	N	String	Street.
City	N	String	City.



document_type_rcd	N	String	Document type code.  P = Passport.  B = Birth Certificate  DL = Driving License  I = ID Card  M = US Military ID  R = Refugee  V = Visa
passport_number	N	String	Document number.
passport_issue_date	Y	Date	Passport issue date.  <b>Default Value:</b> 0001-01-01
passport_expiry_date	Y	Date	Passport expiry date.  <b>Default Value:</b> 0001-01-01
passport_issue_place	N	String	Passport issue place.
passport_birth_place	N	String	Passport birth place.
date_of_birth	Y	String	Date of birth.
passport_issue_country_rcd	N	String	Passport issuing country.
contact_email	Y	String	Contact email.
phone_mobile	N	String	Mobile phone number.
phone_home	N	String	Home phone.
phone_business	N	String	Business phone.
wheelchair_flag	Y	Byte	Identify whether the member required wheelchair.  <b>0</b> = not required.



			<b>1</b> = required.
vip_flag	Y	Byte	Identify whether the client profile is a VIP member.
member_level_rcd	N	String	FFP member level. Base on Avantik setting.
window_seat_flag	Y	Byte	Identify that the member is prefer to set close to the window.  <b>0</b> = Can be any place.  <b>1</b> = Prefer.
redress_number	N	String	Department of Homeland Security's Traveler Redress Inquiry Program (TRIP).
status_code	Y	String	Client profile status code.  A = Active  I = Inactive  D = Delete.
client_password	Y	String	Client profile password.
company_flag	Y	Byte	Identify if this client profile is a B2E profile.  <b>Default Value: 0</b>
client_type_rcd	Y	String	Client type code.  <b>COR</b> = Corporate profile. User of B2E.  <b>IND</b> = Individual profile. For passenger client profile.  <b>STF</b> = Staff passenger.
member_since_date	Y	Date	Member since date.



EditClientProfile function is use for change existing client profile information. The change from this request will not reflex to passenger profile role "MYSELF". If you want to change the passenger profile role "MYSELF", function EditPassengerProfile need to be call instead. Please see **Appendix 6** for the response of EditClientProfile.

### Sample SOAP request message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:EditClientProfile>
      <tem:ClientRequest>
        <tem:title_rcd>MR</tem:title_rcd>
        <tem:Lastname>TESTLASTNAME</tem:Lastname>
        <tem:Firstname>TESTFIRSTNAME</tem:Firstname>
        <tem:Middlename></tem:Middlename>
        <tem:language_rcd>EN</tem:language_rcd>
        <tem:nationality_rcd>JP</tem:nationality_rcd>
        <tem:passenger_weight>0</tem:passenger_weight>
        <tem:gender_type_rcd>M</tem:gender_type_rcd>
        <tem:passenger_type_rcd>ADULT</tem:passenger_type_rcd>
        <tem:address_line1>Address line 1</tem:address_line1>
        <tem:address_line2>Address line 2</tem:address_line2>
        <tem:State>State</tem:State>
        <tem:District>District</tem:District>
      </tem:ClientRequest>
    </tem:EditClientProfile>
  </soapenv:Body>
</soapenv:Envelope>
```



```

<tem:Province>Province</tem:Province>

<tem:zip_code>1111</tem:zip_code>

<tem:po_box>2222</tem:po_box>

<tem:country_rcd>JP</tem:country_rcd>

<tem:Street>Street</tem:Street>

<tem:City>City</tem:City>

<tem:document_type_rcd>P</tem:document_type_rcd>

<tem:residence_country_rcd>JP</tem:residence_country_rcd>

<tem:passport_number>P12345</tem:passport_number>

<tem:passport_issue_date>2014-01-01</tem:passport_issue_date>

<tem:passport_expiry_date>2015-01-01</tem:passport_expiry_date>

<tem:passport_issue_place>Japan</tem:passport_issue_place>

<tem:passport_birth_place>Tokyo</tem:passport_birth_place>

<tem:date_of_birth>1977-12-12</tem:date_of_birth>

<tem:passport_issue_country_rcd>JP</tem:passport_issue_country_rcd>

<tem:contact_name>MR TESTLASTNAME TESTFIRSTNAME</tem:contact_name>

<tem:contact_email>email@test.com</tem:contact_email>

<tem:mobile_email>mobile@test.com</tem:mobile_email>

<tem:phone_mobile>0212521251</tem:phone_mobile>

<tem:phone_home></tem:phone_home>

<tem:phone_fax></tem:phone_fax>

<tem:phone_business></tem:phone_business>

<tem:employee_number></tem:employee_number>

<tem:wheelchair_flag>0</tem:wheelchair_flag>

<tem:vip_flag>0</tem:vip_flag>

```



```

<tem:member_level_rcd></tem:member_level_rcd>

<tem:member_number></tem:member_number>

<tem>window_seat_flag>0</tem>window_seat_flag>

<tem:redress_number></tem:redress_number>

<tem:status_code>A</tem:status_code>

<tem:client_password>password123</tem:client_password>

<tem:company_flag>0</tem:company_flag>

<tem:client_type_rcd>IND</tem:client_type_rcd>

<tem:member_since_date>2010-02-02</tem:member_since_date>

<tem:passenger_role_rcd>SON</tem:passenger_role_rcd>

<tem:client_profile_id> ee88e6ce-f533-11de-b575-0016d30a935b</tem:client_profile_id>

</tem:ClientRequest>

</tem>EditClientProfile>

</soapenv:Body>

</soapenv:Envelope>

```





### 4.5.3 AddPassengerProfile

Name	AddPassengerProfile()		
Description	This function is use for adding passenger profile to client profile.		
Output	Error XML (See Appendix 6)		
Parameter (AddPassengerProfileRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
client_profile_id	Y	Guid	Key use to identify individual client profile.
Parameter (PassengerProfile)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
document_type_rcd	N	String	Document type code.  P = Passport.  B = Birth Certificate  DL = Driving License  I = ID Card  M = US Military ID  R = Refugee  V = Visa
title_rcd	Y	String	Client profile title code.
Lastname	Y	String	Client profile lastname.
firstname	Y	String	Client profile firstname.
middlename	N	String	Client middlename.
language_rcd	Y	String	Language code.
nationality_rcd	N	String	Nationality country code.
passenger_weight	Y	Decimal	Passenger weight.  Default Value: 0
gender_type_rcd	Y	String	Gender type code.



			<p>M = Male</p> <p>F = Female</p>
passenger_type_rcd	Y	String	<p>Passenger type code:</p> <p>ADULT = Adult</p> <p>CHD = Child</p> <p>INF = Infant</p>
passport_number	N	String	Document number.
passport_issue_date	Y	Date	<p>Passport issue date.</p> <p><b>Default Value:</b> 0001-01-01</p>
passport_expiry_date	Y	Date	<p>Passport expiry date.</p> <p><b>Default Value:</b> 0001-01-01</p>
passport_issue_place	N	String	Passport issue place.
passport_birth_place	N	String	Passport birth place.
date_of_birth	Y	String	Date of birth.
passport_issue_country_rcd	N	String	Passport issuing country.
employee_number	N	String	Employee number use for B2E user.
wheelchair_flag	Y	Byte	<p>Identify whether the member required wheelchair.</p> <p><b>0</b> = not required.</p> <p><b>1</b> = required.</p>
vip_flag	Y	Byte	Identify whether the client profile is a VIP member.
member_level_rcd	N	String	FFP member level. Base on Avantik setting.
member_number	N	String	Member number of the client profile in the booking.
window_seat_flag	Y	Byte	Identify that the member is prefer to set close to



			<p>the window.</p> <p><b>0</b> = Can be any place.</p> <p><b>1</b> = Prefer.</p>
redress_number	N	String	Department of Homeland Security's Traveler Redress Inquiry Program (TRIP).
passenger_role_rcd	Y	String	Passenger role code, base on Avantik setting.

This function is use for adding passenger profile information to existing client profile. Base on Avantik restriction, one client profile can only have one passenger profile with role "MYSELF", adding passenger profile with role "MYSELF" is restricted in this function. Please see **Appendix 6** for the response of AddPassengerProfile.



**Sample SOAP request message:**

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:tem="http://tempuri.org/"
    <soapenv:Header/>
    <soapenv:Body>
        <tem:AddPassengerProfile>
            <tem:PassengerRequest>
                <tem:client_profile_id>ee88e6ce-f533-11de-b575-0016d30a935b</tem:client_profile_id>
                <tem:PassengerProfile>
                    <!--Zero or more repetitions:-->
                    <tem:AddPassengerProfileMessage>
                        <tem:document_type_rcd>P</tem:document_type_rcd>
                        <tem:title_rcd>MR</tem:title_rcd>
                        <tem:Lastname>TESTLASTNAME</tem:Lastname>
                        <tem:Firstname>TESTNAME</tem:Firstname>
                        <tem:Middlename></tem:Middlename>
                        <tem:language_rcd>EN</tem:language_rcd>
                        <tem:nationality_rcd>JP</tem:nationality_rcd>
                        <tem:passenger_weight>0</tem:passenger_weight>
                        <tem:gender_type_rcd>M</tem:gender_type_rcd>
                        <tem:passenger_type_rcd>ADULT</tem:passenger_type_rcd>
                        <tem:passport_number>P12345</tem:passport_number>
                        <tem:passport_issue_date>2000-12-12</tem:passport_issue_date>
                        <tem:passport_expiry_date>2015-12-12</tem:passport_expiry_date>
                        <tem:passport_issue_place>Japan</tem:passport_issue_place>
```



```

<tem:passport_birth_place>Tokyo</tem:passport_birth_place>

<tem:date_of_birth>1977-12-12</tem:date_of_birth>

<tem:passport_issue_country_rcd>JP</tem:passport_issue_country_rcd>

<tem:employee_number></tem:employee_number>

<tem:wheelchair_flag>0</tem:wheelchair_flag>

<tem:vip_flag>1</tem:vip_flag>

<tem:member_level_rcd></tem:member_level_rcd>

<tem:member_number></tem:member_number>

<tem>window_seat_flag>0</tem>window_seat_flag>

<tem:redress_number></tem:redress_number>

<tem:passenger_role_rcd>SON</tem:passenger_role_rcd>

</tem:AddPassengerProfileMessage>

</tem:PassengerProfile>

</tem:PassengerRequest>

</tem:AddPassengerProfile>

</soapenv:Body>

</soapenv:Envelope>

```



#### 4.5.4 EditPassengerProfile

Name	EditPassengerProfile()		
Description	This function is use for change passenger profile information.		
Output	Error XML (See Appendix 6)		
Parameter (EditPassengerProfileRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
client_profile_id	Y	Guid	Key use to identify individual client profile.
Parameter (PassengerProfile)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
passenger_profile_id	Y	Guid	Key use to identify individual passenger profile.
document_type_rcd	N	String	Document type code.  P = Passport.  B = Birth Certificate  DL = Driving License  I = ID Card  M = US Military ID  R = Refugee  V = Visa
title_rcd	Y	String	Client profile title code.
Lastname	Y	String	Client profile lastname.
firstname	Y	String	Client profile firstname.
middlename	N	String	Client middlename.
language_rcd	Y	String	Language code.
nationality_rcd	N	String	Nationality country code.
passenger_weight	Y	Decimal	Passenger weight.  Default Value: 0



gender_type_rcd	Y	String	Gender type code.  M = Male  F = Female
passenger_type_rcd	Y	String	Passenger type code:  ADULT = Adult  CHD = Child  INF = Infant
passport_number	N	String	Document number.
passport_issue_date	Y	Date	Passport issue date.  <b>Default Value:</b> 0001-01-01
passport_expiry_date	Y	Date	Passport expiry date.  <b>Default Value:</b> 0001-01-01
passport_issue_place	N	String	Passport issue place.
passport_birth_place	N	String	Passport birth place.
date_of_birth	Y	String	Date of birth.
passport_issue_country_rcd	N	String	Passport issuing country.
employee_number	N	String	Employee number use for B2E user.
wheelchair_flag	Y	Byte	Identify whether the member required wheelchair.  <b>0</b> = not required.  <b>1</b> = required.
vip_flag	Y	Byte	Identify whether the client profile is a VIP member.
member_level_rcd	N	String	FFP member level. Base on Avantik setting.
member_number	N	String	Member number of the client profile in the booking.



window_seat_flag	Y	Byte	Identify that the member is prefer to set close to the window.  <b>0</b> = Can be any place.  <b>1</b> = Prefer.
redress_number	N	String	Department of Homeland Security's Traveler Redress Inquiry Program (TRIP).
passenger_role_rcd	Y	String	Passenger role code, base on Avantik setting.

This function is use for editing passenger profile information. Please see **Appendix 6** for the response of EditPassengerProfile.





**Sample SOAP request message:**

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:tem="http://tempuri.org/"
    <soapenv:Header/>
    <soapenv:Body>
        <tem:EditPassengerProfile>
            <tem:PassenegerRequest>
                <tem:client_profile_id>ee88e6ce-f533-11de-b575-0016d30a935b</tem:client_profile_id>
                <tem:PassengerProfile>
                    <!--Zero or more repetitions:-->
                    <tem:AddPassengerProfileMessage>
                        <tem:document_type_rcd>P</tem:document_type_rcd>
                        <tem:title_rcd>MR</tem:title_rcd>
                        <tem:Lastname>TESTLASTNAME</tem:Lastname>
                        <tem:Firstname>TESTNAME</tem:Firstname>
                        <tem:Middlename></tem:Middlename>
                        <tem:language_rcd>EN</tem:language_rcd>
                        <tem:nationality_rcd>JP</tem:nationality_rcd>
                        <tem:passenger_weight>0</tem:passenger_weight>
                        <tem:gender_type_rcd>M</tem:gender_type_rcd>
                        <tem:passenger_type_rcd>ADULT</tem:passenger_type_rcd>
                        <tem:passport_number>P12345</tem:passport_number>
                        <tem:passport_issue_date>2000-12-12</tem:passport_issue_date>
                        <tem:passport_expiry_date>2015-12-12</tem:passport_expiry_date>
                        <tem:passport_issue_place>Japan</tem:passport_issue_place>
```



```

    <tem:passport_birth_place>Tokyo</tem:passport_birth_place>

    <tem:date_of_birth>1977-12-12</tem:date_of_birth>

    <tem:passport_issue_country_rcd>JP</tem:passport_issue_country_rcd>

    <tem:employee_number></tem:employee_number>

    <tem:wheelchair_flag>0</tem:wheelchair_flag>

    <tem:vip_flag>1</tem:vip_flag>

    <tem:member_level_rcd></tem:member_level_rcd>

    <tem:member_number></tem:member_number>

    <tem>window_seat_flag>0</tem>window_seat_flag>

    <tem:redress_number></tem:redress_number>

    <tem:passenger_role_rcd>SON</tem:passenger_role_rcd>

    <tem:passenger_profile_id>

        ee88e6ce-f533-11de-b575-0016d30a935b

    </tem:passenger_profile_id>

</tem:AddPassengerProfileMessage>

</tem:PassengerProfile>

</tem:PassengerRequest>

</tem>EditPassengerProfile>

</soapenv:Body>

</soapenv:Envelope>

```



#### 4.5.5 PassengerProfileRead

Name	PassengerProfileRead()		
Description	This function is use for read client profile and passenger profile information.		
Output	Client Passenger XML (See Appendix 9)		
Parameter (ClientProfileReadRequest)			
<u>Name</u>	<u>Required</u>	<u>Data Type</u>	<u>Description</u>
client_profile_id	Y	Guid	Key use to identify individual client profile.

This function is use for reading client profile information by using client\_profile\_id as a request parameter, which is different from function GetClientPassenger (section 4.2.19) that required ClientLogon (section 4.2.17) call before GetClientPassenger function can be call.

PassengerProfileRead function is use after CreateClientProfile function for display success created client information.

#### Sample SOAP request message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:PassengerProfileRead>
      <tem:ClientReadRequest>
        <tem:client_profile_id>ee88e6ce-f533-11de-b575-0016d30a935b</tem:client_profile_id>
      </tem:ClientReadRequest>
    </tem:PassengerProfileRead>
  </soapenv:Body>
</soapenv:Envelope>
```



</soapenv:Body>

</soapenv:Envelope>



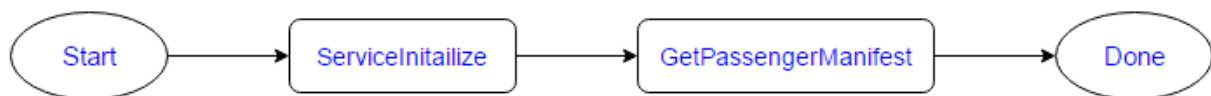
## 4.6 Retrieving Passenger Details

The retrieving passenger details are function to support 3<sup>rd</sup> parties able to get passenger information in booking each flight.

### 4.6.1 Get Passenger Manifest

The **Passenger Manifest API** Engine enables you to integrate a new or existing airline booking engine to the Avantik WebAPI. This document also shows a code sample in .Net C# which can be used as a guide for your application integration.

Below is the workflow of how each function of Passenger Manifest API is call to function GetPassengerManifest to get passenger information which is booked in each of routes as user selected.



**Workflow for GetPassengerManifest**

<b>Description</b>	To Get Passenger Manifest.
<b>Service Name</b>	BookingService
<b>XML Response</b>	Appendix 22

Body Request			
<u>Name</u>	<u>Type</u>	<u>Description</u>	<u>Required</u>
origin_rcd	String	Origin	No



destination_rcd	String	Destination	No
airline_rcd	String	Airline name	Yes
flight_number	Integer	Flight number	Yes
departure_date_from	Date	Destination date from	Yes
departure_date_to	Date	Destination date to	No
bReturnServices	Boolean	Get special services	No
bReturnBagTags	Boolean	Get bag tags	No
bReturnRemarks	Boolean	Get remarks	No
bNotCheckedIn	Boolean	Get not check-in passengers	No
bCheckedIn	Boolean	Get checked-in passengers	No
bOffloaded	Boolean	Get offloaded passengers	No
bNoShow	Boolean	Get no show passengers	No
bInfants	Boolean	Get passengers include infants	No
bConfirmed	Boolean	Get confirmed passengers	No
bWaitlisted	Boolean	Get waitlisted passengers	No
bCancelled	Boolean	Get cancelled passengers	No
bStandby	Boolean	Get standby passengers	No
bIndividual	Boolean	Get individual passengers	No
bGroup	Boolean	Get group passengers	No
bTransit	Boolean	Get transit passengers	No

## XML Request

```
<GetPassengerManifest>

  <PassengersManifestRequest>

    <origin_rcd>string</origin_rcd>

    <destination_rcd>string</destination_rcd>

    <airline_rcd>string</airline_rcd>

    <flight_number>string</flight_number>
```



```

<departure_date_from>dateTime</departure_date_from>

<departure_date_to>dateTime</departure_date_to>

<bReturnServices>boolean</bReturnServices>

<bReturnBagTags>boolean</bReturnBagTags>

<bReturnRemarks>boolean</bReturnRemarks>

<bNotCheckedIn>boolean</bNotCheckedIn>

<bCheckedIn>boolean</bCheckedIn>

<bOffloaded>boolean</bOffloaded>

<bNoShow>boolean</bNoShow>

<bInfants>boolean</bInfants>

<bConfirmed>boolean</bConfirmed>

<bWaitlisted>boolean</bWaitlisted>

<bCancelled>boolean</bCancelled>

<bStandby>boolean</bStandby>

<bIndividual>boolean</bIndividual>

<bGroup>boolean</bGroup>

<bTransit>boolean</bTransit>

</PassengersManifestRequest>

</GetPassengerManifest>

```



## 5. Appendix 1: Serviceinitialize xml

The serviceinitialize XML consists of success or failure authentication and the serverID that config from each server to let partner know the server number.

```
<ServiceInitializeResponse xmlns="http://tempuri.org/">

    <ServiceInitializeResult><![CDATA[<error>

        <code>000</code>

        <message>Success Request Transaction</message>

        <server_id>server4</server_id>

    </error>]]></ServiceInitializeResult>

</ServiceInitializeResponse>
```

Name	Description
Code	Number of code response
message	Success or failure response
server_id	The Server name which configured in the server

## 6. APPENDIX 2: AVAILABILITY XML

The Availability XML consists of two main sections, **AvailabilityOutbound** and **AvailabilityReturn**. If you search for a one way flight the XML will only show the **AvailabilityOutbound** section.

```
<Availability>

    <AvailabilityOutbound>
```





<AvailabilityFlight>

```

<airline_rcd>GR</airline_rcd>

<flight_number>203</flight_number>

<booking_class_rcd>T</booking_class_rcd>

<boarding_class_rcd>Y</boarding_class_rcd>

<flight_id>5e87a89b-da69-11de-b54d-001641975e09</flight_id>

<origin_rcd>ACI</origin_rcd>

<destination_rcd>BRS</destination_rcd>

<origin_name>Alderney</origin_name>

<destination_name>Bristol</destination_name>

<flight_status_rcd>ACTIVE</flight_status_rcd>

<departure_date>2010-03-31T00:00:00+07:00</departure_date>

<planned_departure_time>745</planned_departure_time>

<planned_arrival_time>800</planned_arrival_time>

<fare_id>b1994a20-2103-11df-8509-001641975def</fare_id>

<fare_code>TWEB</fare_code>

<transit_flight_id>

    ee88e6ce-f533-11de-b575-0016d30a935b

</transit_flight_id>

<transit_booking_class_rcd>T</transit_booking_class_rcd>

<transit_boarding_class_rcd>Y</transit_boarding_class_rcd>

<transit_airport_rcd>GCI</transit_airport_rcd>

<transit_departure_date>

    2010-03-31T00:00:00+07:00
    
```



```

</transit_departure_date>

<transit_planned_departure_time>

1100

</transit_planned_departure_time>

<transit_planned_arrival_time>

1155

</transit_planned_arrival_time>

<transit_fare_id>

b1994a20-2103-11df-8509-001641975def

</transit_fare_id>

<transit_name>Guernsey</transit_name>

<transit_waitlist_open_flag>1</transit_waitlist_open_flag>

<transit_airline_rcd>GR</transit_airline_rcd>

<transit_flight_number>642</transit_flight_number>

<transit_flight_status_rcd>ACTIVE</transit_flight_status_rcd>

<transit_flight_duration>55</transit_flight_duration>

<transit_class_open_flag>1</transit_class_open_flag>

<transit_nesting_string>

STBWDKMVILROCHXZNAUJP

</transit_nesting_string>

<nesting_string>TBWDKMVILROCHXZNAUJP</nesting_string>

<nested_book_available>10</ nested_book_available >

<full_flight_flag>1</full_flight_flag>

<class_open_flag>0</class_open_flag>

```



```

<close_web_sales>0</close_web_sales>

<currency_rcd />

<total_adult_fare>120</total_adult_fare>

<total_child_fare>100</total_child_fare>

<total_infant_fare>50</total_infant_fare>

<fare_column>1</fare_column>

<flight_comment>Public Comment</flight_comment>

<restriction_text>

    Free to change - Refund permitted

</restriction_text>

<flight_duration>15</flight_duration>

<class_open_flag>0</class_open_flag>

<waitlist_open_flag>0</waitlist_open_flag>

<refundable_flag>1</refundable_flag>

<currency_rcd>GBP</currency_rcd>

</AvailabilityFlight>

</AvailabilityOutbound>

<AvailabilityReturn>

    .

    .

    <!--Same structure as <AvailabilityOutbound/> -->

    .

    .

</AvailabilityReturn>

```



</Availability>

The Fields used in the Availability XML are described in the following table:

Name	Description
flight_id	Unique flight ID. Used when calling <b>FlightAdd()</b> function.
airline_rcd	Airline IATA code.
flight_number	Flight number.
flight_status_rcd	Flight status code.
aircraft_type_rcd	Aircraft type code.
restriction_text	Restriction description.
origin_rcd	Origin airport code.
origin_name	Origin airport display name.
destination_rcd	Airport destination code.
destination_name	Destination airport display name.
departure_date	Departure date.
planned_departure_time	Departure time.
planned_arrival_time	Arrival time.
fare_code	Fare basis code.
fare_id	Unique fare ID.
booking_class_rcd	Booking class code.
boarding_class_rcd	Boarding class code.
class_open_flag	Used to identify if this class is open.
waitlist_open_flag	Used to identify if this flight is open for waitlist.
nesting_string	Booking class nesting string.
nested_book_available	Actual number of availability.
refundable_flag	Identifies if this a refundable fare.
close_web_sales	Identifies if this flight is closed for web sales.
full_flight_flag	Identifies whether this flight is full or not.
total_adult_fare	Adult fare including tax and VAT.



total_child_fare	Child fare including tax and VAT.
total_infant_fare	Infant fare including tax and VAT.
currency_rcd	Currency Code.
flight_duration	flight duration.
transit_airport_rcd	Transit airport code.
transit_name	Full transit airport name.
transit_flight_id	Transit flight ID.
transit_airline_rcd	Transit airline IATA code.
transit_flight_number	Transit flight number.
transit_flight_status_rcd	Transit flight status code.
transit_departure_date	Transit departure date.
transit_planned_departure_time	Transit departure time.
transit_planned_arrival_time	Transit arrival time.
transit_flight_duration	Transit flight duration.
transit_fare_code	Transit fare code.
transit_fare_id	Transit unique fare ID.
transit_booking_class_rcd	Transit flight booking class.
transit_boarding_class_rcd	Transit flight boarding class.
transit_class_open_flag	Used to identify if this class is open for the transit flight.
transit_waitlist_open_flag	Used to identify if this transit flight is open for waitlist.
transit_nesting_string	Transit flight booking class nesting string.



## 7. APPENDIX 3: BOOKING XML

The Booking XML consists of eight main sections, **BookingHeader**, **FlightSegment**, **Passenger**, **Payment**, **Mapping**, **Tax**, **Quote** and **Fee**.

<Booking>

<BookingHeader>

<booking\_id>a90c12d8-bd15-40fb-816e-85c0bb9b2eca</booking\_id>

<booking\_source\_rcd>B2B</booking\_source\_rcd>

<currency\_rcd>GBP</currency\_rcd>

<booking\_number>667233496</booking\_number>

<record\_locator>LVBC7R</record\_locator>

<number\_of\_adults>2</number\_of\_adults>

<number\_of\_children>0</number\_of\_children>

<number\_of\_infants>0</number\_of\_infants>

<language\_rcd>EN</language\_rcd>

<contact\_name>njoy ck</contact\_name>

<contact\_email>chulanee@tiksystems.com</contact\_email>

<phone\_mobile>08888888888</phone\_mobile>

<phone\_home>0222222222</phone\_home>

<phone\_business>09999999999</phone\_business>



<received\_from>WEB</received\_from>

<phone\_fax />

<phone\_search>0888888888</phone\_search>

<comment />

<lastname>njoy</lastname>

<firstname>ck</firstname>

<city>hug city</city>

<street>lovely street</street>

<middlename>CK</middlename>

<address\_line1>test add line 1</address\_line1>

<address\_line2>test add line 2</address\_line2>

<state />

<district>sweet district</district>

<province>kiss province</province>

<zip\_code>zip12345</zip\_code>

<po\_box>PO123</po\_box>

<country\_rcd>CC</country\_rcd>

<title\_rcd>MISS</title\_rcd>

</BookingHeader>



<FlightSegment>

<booking\_segment\_id>

45904819-575d-44fd-b679-a6a3a9627758

</booking\_segment\_id>

<flight\_id>c7619bfe-d0e4-44c5-96df-5c7958c6a74a</flight\_id>

<flight\_connection\_id>

00000000-0000-0000-0000-000000000000

</flight\_connection\_id>

<airline\_rcd>GR</airline\_rcd>

<flight\_number>602</flight\_number>

<departure\_date>2009-04-05T10:00:00-07:00</departure\_date>

<departure\_time>1030</departure\_time>

<booking\_class\_rcd>H</booking\_class\_rcd>

<boarding\_class\_rcd>Y</boarding\_class\_rcd>

<segment\_status\_rcd>XX</segment\_status\_rcd>

<number\_of\_units>2</number\_of\_units>

<arrival\_time>1130</arrival\_time>

<origin\_rcd>GCI</origin\_rcd>

<destination\_rcd>LGW</destination\_rcd>





<segment\_change\_status\_rcd>KK</segment\_change\_status\_rcd>

<info\_segment\_flag>0</info\_segment\_flag>

<high\_priority\_waitlist\_flag>0</high\_priority\_waitlist\_flag>

<od\_origin\_rcd>GCI</od\_origin\_rcd>

<od\_destination\_rcd>LGW</od\_destination\_rcd>

<origin\_name>Guernsey</origin\_name>

<destination\_name>London Gatwick</destination\_name>

<segment\_status\_name>Cancelled</segment\_status\_name>

<seatmap\_flag>1</seatmap\_flag>

</FlightSegment>

<Passenger>

<passenger\_id>90917fbb-04df-4706-8129-67598e5462b2</passenger\_id>

<booking\_id>a90c12d8-bd15-40fb-816e-85c0bb9b2eca</booking\_id>

<client\_number>0</client\_number>

<passenger\_profile\_id>

00000000-0000-0000-0000-000000000000

</passenger\_profile\_id>

<passenger\_type\_rcd>ADULT</passenger\_type\_rcd>

<lastname>ARIKANNI</lastname>



<firstname>B</firstname>

<employee\_number />

<date\_of\_birth>19870422</date\_of\_birth>

<title\_rcd>MR</title\_rcd>

<gender\_type\_rcd>M</gender\_type\_rcd>

<passport\_issue\_date>20100531</passport\_issue\_date>

<passport\_expiry\_date>20161231</passport\_expiry\_date>

<wheelchair\_flag>0</wheelchair\_flag>

<passenger\_weight>0</passenger\_weight>

<client\_profile\_id>

00000000-0000-0000-0000-000000000000

</client\_profile\_id>

</Passenger>

<Payment>

<booking\_payment\_id>

39b4ef4c-0186-4fca-905b-2ed66c8909a8

</booking\_payment\_id>

<booking\_segment\_id>

00000000-0000-0000-0000-000000000000



</booking\_segment\_id>

<booking\_id>a90c12d8-bd15-40fb-816e-85c0bb9b2eca</booking\_id>

<voucher\_payment\_id>

00000000-0000-0000-0000-000000000000

</voucher\_payment\_id>

<form\_of\_payment\_rcd>INV</form\_of\_payment\_rcd>

<currency\_rcd>GBP</currency\_rcd>

<receive\_currency\_rcd>GBP</receive\_currency\_rcd>

<agency\_code>BEE</agency\_code>

<debit\_agency\_code />

<payment\_amount>214.00</payment\_amount>

<receive\_payment\_amount>214.00</receive\_payment\_amount>

<acct\_payment\_amount>214.00</acct\_payment\_amount>

<payment\_by>a48e7722-dcc6-4340-a572-d105fb42d675</payment\_by>

<payment\_date\_time>2009-04-02T19:22:52-07:00</payment\_date\_time>

<payment\_due\_date\_time>

2009-05-02T10:00:00-07:00

</payment\_due\_date\_time>

<document\_amount>214.00</document\_amount>



<void\_by>00000000-0000-0000-0000-000000000000</void\_by>

<void\_date\_time>0001-01-01T00:00:00</void\_date\_time>

<record\_locator>LVBC7R</record\_locator>

<document\_number />

<form\_of\_payment\_subtype\_rcd />

</Payment>

<Mapping>

<passenger\_id>90917fbb-04df-4706-8129-67598e5462b2</passenger\_id>

<booking\_segment\_id>

62d8401a-50e0-49bb-8b12-68313c5bd9d3

</booking\_segment\_id>

<origin\_rcd>LGW</origin\_rcd>

<destination\_rcd>GCI</destination\_rcd>

<lastname>ARIKANNI</lastname>

<firstname>B</firstname>

<gender\_type\_rcd>M</gender\_type\_rcd>

<title\_rcd>MR</title\_rcd>

<passenger\_type\_rcd>ADULT</passenger\_type\_rcd>

<date\_of\_birth>0001-01-01T00:00:00</date\_of\_birth>



<flight\_id>e7a3e072-744b-458f-922a-9a0e2d0de157</flight\_id>

<airline\_rcd>GR</airline\_rcd>

<flight\_number>607</flight\_number>

<departure\_date>2009-04-14T10:00:00-07:00</departure\_date>

<booking\_class\_rcd>H</booking\_class\_rcd>

<boarding\_class\_rcd>Y</boarding\_class\_rcd>

<inventory\_class\_rcd>P</inventory\_class\_rcd>

<seat\_number />

<seat\_row>0</seat\_row>

<seat\_column />

<currency\_rcd>GBP</currency\_rcd>

<net\_total>51.50</net\_total>

<tax\_amount>23.50</tax\_amount>

<fare\_amount>28.00</fare\_amount>

<fare\_id>908aea6f-4c2a-4b9e-9e97-96a69c7318bd</fare\_id>

<fare\_code>HWEB</fare\_code>

<refundable\_flag>0</refundable\_flag>

<e\_ticket\_flag>1</e\_ticket\_flag>

<standby\_flag>0</standby\_flag>



<agency\_code>BEE</agency\_code>

<passenger\_status\_rcd>OK</passenger\_status\_rcd>

<ticket\_number>9244500582501</ticket\_number>

<ticketing\_date\_time>

2009-04-02T19:32:47-07:00

</ticketing\_date\_time>

<restriction\_text>Fee to Change - No Refund</restriction\_text>

<void\_date\_time>0001-01-01T00:00:00</void\_date\_time>

<payment\_amount>51.50</payment\_amount>

</Mapping>

<Tax>

<passenger\_segment\_tax\_id>

bff12033-f551-4fe5-a8e7-fed7977ae969

</passenger\_segment\_tax\_id>

<tax\_amount>7.10</tax\_amount>

<tax\_amount\_incl>7.10</tax\_amount\_incl>

<acct\_amount>7.10</acct\_amount>

<acct\_amount\_incl>7.10</acct\_amount\_incl>

<sales\_amount>7.10</sales\_amount>



<sales\_amount\_incl>7.10</sales\_amount\_incl>

<tax\_rcd>UB</tax\_rcd>

<passenger\_id>90917fbb-04df-4706-8129-67598e5462b2</passenger\_id>

<tax\_id>f763a4d5-ba30-4cc5-a914-a3343169532d</tax\_id>

<booking\_segment\_id>

9cb0e994-c867-4d2a-98c5-a064a20cf884

</booking\_segment\_id>

<tax\_currency\_rcd>GBP</tax\_currency\_rcd>

<sales\_currency\_rcd>GBP</sales\_currency\_rcd>

<display\_name>Passenger Service Charge</display\_name>

</Tax>

<Quote>

<booking\_segment\_id>

00000000-0000-0000-0000-000000000000

</booking\_segment\_id>

<passenger\_type\_rcd>ADULT</passenger\_type\_rcd>

<currency\_rcd>GBP</currency\_rcd>

<charge\_type>FARE</charge\_type>

<charge\_name>Fare</charge\_name>



```

<charge_amount>56</charge_amount>

<total_amount>103</total_amount>

<tax_amount>0</tax_amount>

<passenger_count>2</passenger_count>

<sort_sequence>0</sort_sequence>

</Quote>

<Fee>

  <booking_fee_id>

    ec49dabc-9cd4-497e-8d01-090223ffd8a8

  </booking_fee_id>

  <fee_amount>20.00</fee_amount>

  <booking_id>a90c12d8-bd15-40fb-816e-85c0bb9b2eca</booking_id>

  <passenger_id>00000000-0000-0000-0000-000000000000</passenger_id>

  <currency_rcd>GBP</currency_rcd>

  <acct_fee_amount>20.00</acct_fee_amount>

  <fee_id>53ad85fc-bbb7-4576-a56d-26b807adf892</fee_id>

  <vat_percentage>0</vat_percentage>

  <fee_amount_incl>20.00</fee_amount_incl>

  <acct_fee_amount_incl>20.00</acct_fee_amount_incl>

```





```

<fee_rcd>PNRUPD</fee_rcd>

<display_name>Change Booking Fee</display_name>

<account_fee_date_time>

2009-04-03T01:32:17-07:00

</account_fee_date_time>

<void_date_time>0001-01-01T00:00:00</void_date_time>

<void_by>00000000-0000-0000-0000-000000000000</void_by>

<payment_amount>20.00</payment_amount>

<booking_segment_id>

00000000-0000-0000-0000-000000000000

</booking_segment_id>

<agency_code>bee</agency_code>

</Fee>

</Booking>

```

The Fields used in the Booking XML are described in the following tables:



Name	BookingHeader
Description	Shows booking information and booking contact details.
Field name and description	
Name	Description
booking_id	Unique booking ID.
currency_rcd	Booking currency.
booking_number	Booking number.
record_locator	Booking reference.
number_of_adults	Number of adults in the booking.
number_of_children	Number of children in the booking.
number_of_infants	Number of infants in the booking.
language_rcd	Language code.
agency_code	Code of agency that has made the booking.
contact_name	Name of contact person.
contact_email	E-mail address of contact person.
phone_mobile	Mobile phone number of contact person.
phone_home	Home phone of contact person.
phone_business	Business phone of contact person.
received_from	Booking source information.
phone_fax	Fax number of contact person.
Comment	Booking comments.
group_name	Name of the group if the booking is a group booking.
group_booking_flag	Identifies whether this booking is a group booking or not.
agency_name	Name of agency that has made the booking.
own_agency_flag	Flag identifying the booking as being made by an airline internal agency.
web_agency_flag	Flag identifying the booking as being made by a web agency.



Name	FlightSegment
Description	Shows detailed flight segment information.
Field name and description	
Name	Description
booking_segment_id	Unique flight segment ID.
flight_id	Flight ID.
flight_connection_id	If the segment is part of a connection (married segment), then the same flight_connection_id will be displayed for each of the segments in the connection that are married together (i.e. same flight_connection_id for flight A-B and flight B-C for an A-C connection via B).
airline_rcd	Airline IATA code.
flight_number	Flight number.
departure_date	Departure date.
departure_time	Departure time.
booking_class_rcd	Booking class code.
boarding_class_rcd	Boarding class code.
segment_status_rcd	Flight segment status:  HK = Confirmed  HL = Waitlisted  XX = Cancelled
booking_id	Unique booking ID.
number_of_units	Number of passengers on each segment booked.
journey_time	Journey time in minutes.
arrival_time	Arrival time.
origin_rcd	Origin airport code.
destination_rcd	Destination airport code.



origin_name	Origin airport name.
destination_name	Destination airport name.
segment_status_name	Flight segment status name.
seatmap_flag	Used to identify whether this segment allows advance seat assignment.
close_web_sales_flag	Identifies if this flight is closed for web sales.

Name	Passenger
Description	Shows detailed passenger information.
Field name and description	
<u>Name</u>	<u>Description</u>
passenger_id	Unique passenger ID.
booking_id	Unique booking ID.
passenger_type_rcd	Passenger type code:  ADULT  CHD  INF
Lastname	Passenger last name.
Firstname	Passenger first name.
date_of_birth	Passenger date of birth.
title_rcd	Passenger title.
gender_type_rcd	Passenger gender code:  M = Male  F = Female
passport_issue_date	Passport issue date.
passport_expiry_date	Passport expiry date.
wheelchair_flag	Identifies if a special service request for a wheelchair exists.



passenger_weight	Passenger weight.
client_profile_id	Unique client profile ID.



Name	Payment
Description	Shows detailed payment information.
Field name and description	
<u>Name</u>	<u>Description</u>
booking_payment_id	Unique payment ID.
booking_segment_id	Flight segment ID.
booking_id	Booking ID.
voucher_payment_id	Unique voucher payment ID.
form_of_payment_rcd	Form of payment code:  CC = credit card  INV = invoice  CRAG = credit agency account  VOUCHER = voucher
currency_rcd	Currency code.
agency_code	Code of agency that has made the booking.
payment_amount	Payment amount.
payment_by	Payment user ID.
payment_date_time	Date of payment.
payment_due_date_time	Payment due date.
document_amount	Payment amount.
void_by	User ID of user that has voided a payment.
void_date_time	Date and time when payment has been voided.
record_locator	Booking reference.
document_number	Document number (credit card number).
form_of_payment_subtype_rcd	Form of payment subtype code. The code is based on form of payment subtype settings in Avantik.



Name	Tax
Description	Shows detailed tax information.
Field name and description	
<u>Name</u>	<u>Description</u>
passenger_segment_tax_id	Unique segment tax ID.
tax_amount	Tax amount.
tax_rcd	Tax code (based on airline specific tax definition in Avantik).
passenger_id	Unique passenger ID.
tax_id	Unique tax ID.
booking_segment_id	Flight segment ID.
tax_currency_rcd	Tax currency code.
sales_currency_rcd	Sale currency code.
display_name	Tax display name.

Name	Quote
Description	Shows detailed Fare Quote information.
Field name and description	
<u>Name</u>	<u>Description</u>
passenger_type_rcd	Passenger type code:  ADULT  CHD  INF
charge_type	Charge type code.
charge_name	Full charge name.
charge_amount	Charge amount.



total_amount	Total amount including tax (i.e. VAT, GST etc.).
tax_amount	Tax amount (i.e. VAT, GST etc.).
passenger_count	Number of passengers.
sort_sequence	Quote sort sequence.





Name	Fee
Description	Shows detailed booking fee information.
Field name and description	
<u>Name</u>	<u>Description</u>
booking_fee_id	Unique booking fee ID.
fee_amount	Fee amount.
booking_id	Booking ID.
passenger_id	Passenger ID.
currency_rcd	Fee currency code.
fee_id	Fee ID.
vat_percentage	Fee VAT percentage.
fee_rcd	Fee code.
display_name	Fee display name.
void_date_time	Fee void date time.
void_by	User ID of user that has voided a payment.
payment_amount	Fee payment amount.
booking_segment_id	Flight segment ID.
agency_code	Code of agency that has made the booking.



Name	Mapping
Description	Shows detailed ticket information.
Field name and description	
<u>Name</u>	<u>Description</u>
passenger_id	Passenger ID.
booking_segment_id	Flight segment ID.
origin_rcd	Origin airport code.
destination_rcd	Destination airport code.
Lastname	Passenger last name.
Firstname	Passenger first name.
gender_type_rcd	Passenger gender code:  M = Male  F = Female
title_rcd	Passenger title code.
passenger_type_rcd	Passenger type code:  ADULT  CHD  INF
flight_id	Flight ID.
airline_rcd	Airline IATA code.
flight_number	Flight number.
departure_date	Flight departure date.
booking_class_rcd	Booking class code.
boarding_class_rcd	Boarding class code.
seat_number	Seat number.
seat_row	Seat row.
seat_column	Seat column.
net_total	Total ticket price.



tax_amount	Tax amount.
fare_amount	Fare amount.
fare_id	Fare ID.
fare_code	Fare basis code.
e_ticket_flag	Identifies if this ticket is an e-ticket.
standby_flag	Identifies if this ticket is a standby ticket.
transferable_fare_flag	Identifies if this ticket is a transferable ticket.
agency_code	Code of agency that has made the booking.
ticket_number	Ticket number.
Ticketing_date_time	Ticketing date.
ticketing_by	Ticketing by.
check_in_sequence	Flight check-in sequence.
group_sequence	Flight check-in group sequence.
number_of_pieces	Number of baggage.
baggage_weight	Baggage allowance.
excess_baggage_weight	Excess baggage weight.
check_in_baggage_weight	Check-in baggage weight.



## 8. APPENDIX 4: ITINERARY XML

The Itinerary XML consists of nine main sections, **Header**, **Itinerary**, **Passengers**, **Remarks**, **Payments**, **Tickets**, **TicketTaxes**, **Fees** and **TicketQuotes**.

```
<?xml version="1.0"?>

<Booking>

  <Header>

    <BookingHeader>

      <booking_id>

        {0F6CFF4A-C218-4C28-9B77-E174DDFAE64E}

      </booking_id>

      <client_profile_id></client_profile_id>

      <booking_number>667031630</booking_number>

      <record_locator>FO63YD</record_locator>

      <booking_source_rcd>INT</booking_source_rcd>

      <create_date_time>20080422 04:15:46</create_date_time>

      <create_by>

        {95CF3057-9F4D-45FA-886A-064AA6BECF24}

      </create_by>
```



<number\_of\_adults>0</number\_of\_adults>

<number\_of\_children>0</number\_of\_children>

<number\_of\_infants>0</number\_of\_infants>

<update\_date\_time>20080422 04:17:12</update\_date\_time>

<update\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</update\_by>

<currency\_rcd>GBP</currency\_rcd>

<language\_rcd>EN</language\_rcd>

<agency\_code>TESTAGENT</agency\_code>

<contact\_name>pax</contact\_name>

<contact\_email></contact\_email>

<contact\_email\_cc></contact\_email\_cc>

<phone\_mobile>232434</phone\_mobile>

<phone\_mobile\_cc></phone\_mobile\_cc>

<phone\_home></phone\_home>

<phone\_business></phone\_business>

<received\_from>no</received\_from>

<phone\_fax></phone\_fax>



<phone\_search>232434</phone\_search>

<comment></comment>

<notify\_by\_rcd>PAX</notify\_by\_rcd>

<notify\_by\_email\_flag></notify\_by\_email\_flag>

<notify\_by\_sms\_flag></notify\_by\_sms\_flag>

<group\_name></group\_name>

<group\_booking\_flag></group\_booking\_flag>

<package\_booking\_flag></package\_booking\_flag>

<lastname></lastname>

<firstname></firstname>

<middlename></middlename>

<address\_line1></address\_line1>

<address\_line2></address\_line2>

<street></street>

<state></state>

<district></district>

<province></province>

<city></city>

<zip\_code></zip\_code>



<po\_box></po\_box>

<title\_rcd></title\_rcd>

<country\_rcd></country\_rcd>

<invoice\_receiver></invoice\_receiver>

<tax\_id></tax\_id>

<purchase\_order>ABCDEF</purchase\_order>

<project\_number>8888888</project\_number>

<cost\_center>Cost Center</cost\_center>

<agency\_name>TestAgent</agency\_name>

<agency\_name>TestAgent</agency\_name>

<own\_agency\_flag></own\_agency\_flag>

<web\_agency\_flag>1</web\_agency\_flag>

<iata\_number></iata\_number>

<ticket\_agency\_address></ticket\_agency\_address>

<client\_number>0</client\_number>

<client\_lastname></client\_lastname>

<client\_firstname></client\_firstname>

<client\_middlename></client\_middlename>

<client\_member\_number></client\_member\_number>



<client\_contact\_name></client\_contact\_name>

<client\_address\_line1></client\_address\_line1>

<client\_address\_line2></client\_address\_line2>

<client\_street></client\_street>

<client\_state></client\_state>

<client\_district></client\_district>

<client\_province></client\_province>

<client\_city></client\_city>

<client\_zip\_code></client\_zip\_code>

<client\_po\_box></client\_po\_box>

<client\_country\_rcd></client\_country\_rcd>

<client\_title\_rcd></client\_title\_rcd>

<client\_company\_flag></client\_company\_flag>

<client\_number\_of\_children>0</client\_number\_of\_children>

<client\_phone\_home></client\_phone\_home>

<client\_phone\_business></client\_phone\_business>

<client\_phone\_mobile></client\_phone\_mobile>

<client\_phone\_fax></client\_phone\_fax>

<client\_contact\_email></client\_contact\_email>





<client\_language\_rcd></client\_language\_rcd>

<client\_member\_since\_date></client\_member\_since\_date>

<client\_member\_level\_rcd></client\_member\_level\_rcd>

<create\_name>System Administrator</create\_name>

<update\_name>System Administrator</update\_name>

<utc\_date\_time>20080422 05:27:23</utc\_date\_time>

<ticket\_total>120</ticket\_total>

<fee\_total>0</fee\_total>

<ticket\_payment\_total>120</ticket\_payment\_total>

<fee\_payment\_total>0</fee\_payment\_total>

</BookingHeader>

</Header>

<Itinerary>

<FlightSegment>

<booking\_segment\_id>

{0E7E5DBC-55B4-4BE4-9E16-28316A45E5D0}

</booking\_segment\_id>

<flight\_id>

{BA5F6853-CF05-4ECB-B595-3012607DA61D}



</flight\_id>

<airline\_rcd>GR</airline\_rcd>

<flight\_number>805</flight\_number>

<departure\_date>20080516 00:00:00</departure\_date>

<booking\_class\_rcd>T</booking\_class\_rcd>

<create\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</create\_by>

<create\_date\_time>20080422 04:15:46</create\_date\_time>

<update\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</update\_by>

<update\_date\_time>20080422 04:15:46</update\_date\_time>

<departure\_time>730</departure\_time>

<arrival\_time>750</arrival\_time>

<journey\_time>20</journey\_time>

<segment\_status\_rcd>HK</segment\_status\_rcd>

<segment\_change\_status\_rcd>KK</segment\_change\_status\_rcd>

<booking\_id>



{0F6CFF4A-C218-4C28-9B77-E174DDFAE64E}

</booking\_id>

<number\_of\_units>3</number\_of\_units>

<origin\_rcd>JER</origin\_rcd>

<destination\_rcd>GCI</destination\_rcd>

<od\_origin\_rcd>JER</od\_origin\_rcd>

<od\_destination\_rcd>GCI</od\_destination\_rcd>

<flight\_connection\_id></flight\_connection\_id>

<status\_name>Confirmed</status\_name>

<class\_name>Economy</class\_name>

<airline\_name>Aurigny Air Services</airline\_name>

<origin\_name>Jersey</origin\_name>

<destination\_name>Guernsey</destination\_name>

<origin\_terminal></origin\_terminal>

<destination\_terminal></destination\_terminal>

<origin\_phone>01481 822886</origin\_phone>

<destination\_phone>01481 822886</destination\_phone>

<planned\_arrival\_time>750</planned\_arrival\_time>

<flight\_flown\_date\_time></flight\_flown\_date\_time>



<allow\_web\_checkin\_flag></allow\_web\_checkin\_flag>

<flight\_check\_in\_status\_rcd></flight\_check\_in\_status\_rcd>

<flight\_flown\_name></flight\_flown\_name>

<create\_name>System Administrator</create\_name>

<update\_name>System Administrator</update\_name>

</FlightSegment>

</Itinerary>

<Passengers>

<Passenger>

<passenger\_id>

{ACD01445-88F0-48DC-8D36-12006ADC633E}

</passenger\_id>

<booking\_id>

{0F6CFF4A-C218-4C28-9B77-E174DDFAE64E}

</booking\_id>

<passenger\_type\_rcd>ADULT</passenger\_type\_rcd>

<lastname>TEST</lastname>

<firstname>PHET</firstname>

<date\_of\_birth></date\_of\_birth>



<title\_rcd>MR</title\_rcd>

<create\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</create\_by>

<create\_date\_time>20080422 04:15:46</create\_date\_time>

<update\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</update\_by>

<update\_date\_time>20080422 04:15:46</update\_date\_time>

<gender\_type\_rcd>M</gender\_type\_rcd>

<client\_number>0</client\_number>

<member\_number></member\_number>

<member\_level\_rcd></member\_level\_rcd>

<nationality\_rcd></nationality\_rcd>

<residence\_country\_rcd></residence\_country\_rcd>

<passport\_number></passport\_number>

<passport\_issue\_date></passport\_issue\_date>

<passport\_expiry\_date></passport\_expiry\_date>

<passport\_issue\_place></passport\_issue\_place>



<passport\_birth\_place></passport\_birth\_place>

<wheelchair\_flag></wheelchair\_flag>

<vip\_flag></vip\_flag>

<passenger\_weight>0</passenger\_weight>

<passenger\_type>Adult</passenger\_type>

<title>MR</title>

<gender\_type>Male</gender\_type>

<create\_name>System Administrator</create\_name>

<update\_name>System Administrator</update\_name>

</Passenger>

</Passengers>

<Remarks>

<Remark>

<booking\_remark\_id>

{43C1FA31-7C28-4E73-A7FA-A36B34AA8D60}

</booking\_remark\_id>

<client\_profile\_id></client\_profile\_id>

<remark\_type\_rcd>HISTORY</remark\_type\_rcd>

<timelimit\_date\_time></timelimit\_date\_time>



<complete\_flag></complete\_flag>

<process\_message\_flag></process\_message\_flag>

<remark\_text>Change Reference: no</remark\_text>

<create\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</create\_by>

<create\_date\_time>20080422 04:17:12</create\_date\_time>

<update\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</update\_by>

<update\_date\_time>20080422 04:17:12</update\_date\_time>

<booking\_id>

{0F6CFF4A-C218-4C28-9B77-E174DDFAE64E}

</booking\_id>

<added\_by>LBARCGR / SYS</added\_by>

<display\_name>Booking History</display\_name>

<create\_name>System Administrator</create\_name>

<update\_name>System Administrator</update\_name>

</Remark>



</Remarks>

<Payments>

<Payment>

<booking\_payment\_id>

{9B588CC8-A51C-44FF-85C0-E4D8FF4DEED6}

</booking\_payment\_id>

<booking\_id>

{0F6CFF4A-C218-4C28-9B77-E174DDFAE64E}

</booking\_id>

<form\_of\_payment\_rcd>BANK</form\_of\_payment\_rcd>

<currency\_rcd>GBP</currency\_rcd>

<payment\_amount>49.68</payment\_amount>

<fee\_amount>0</fee\_amount>

<acct\_payment\_amount>49.68</acct\_payment\_amount>

<payment\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</payment\_by>

<payment\_date\_time>20080422 04:16:01</payment\_date\_time>

<payment\_due\_date\_time></payment\_due\_date\_time>





<document\_number>tt11/2222/333-555</document\_number>

<payment\_number>1000039394</payment\_number>

<approval\_code></approval\_code>

<expiry\_month>0</expiry\_month>

<expiry\_year>0</expiry\_year>

<name\_on\_card></name\_on\_card>

<due\_date\_flag></due\_date\_flag>

<document\_amount>0</document\_amount>

<form\_of\_payment\_subtype\_rcd>

EBANK

</form\_of\_payment\_subtype\_rcd>

<issue\_date></issue\_date>

<payment\_remark></payment\_remark>

<issue\_by></issue\_by>

<create\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</create\_by>

<create\_date\_time>20080422 11:16:00</create\_date\_time>



<update\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</update\_by>

<update\_date\_time>20080422 11:16:00</update\_date\_time>

<void\_by></void\_by>

<void\_date\_time></void\_date\_time>

<booking\_reference></booking\_reference>

<agency\_code>LBARCGR</agency\_code>

<debit\_agency\_code></debit\_agency\_code>

<form\_of\_payment>Bank Counter</form\_of\_payment>

<form\_of\_payment\_subtype>

EBANK COUNTER

</form\_of\_payment\_subtype>

<ticket\_booking\_id>0</ticket\_booking\_id>

<void\_name></void\_name>

<create\_name>System Administrator</create\_name>

<update\_name>System Administrator</update\_name>

</Payment>

</Payments>



<Tickets>

<Ticket>

<booking\_id>

{0F6CFF4A-C218-4C28-9B77-E174DDFAE64E}

</booking\_id>

<passenger\_id>

{4EFAC125-EF4E-41A7-B70A-61371B9062B6}

</passenger\_id>

<booking\_segment\_id>

{824879F9-A8A4-4522-ABBD-EE425E73B56F}

</booking\_segment\_id>

<origin\_rcd>GCI</origin\_rcd>

<destination\_rcd>JER</destination\_rcd>

<lastname>TEST</lastname>

<firstname>LILLY</firstname>

<gender\_type\_rcd>F</gender\_type\_rcd>

<title\_rcd>MRS</title\_rcd>

<passenger\_type\_rcd>ADULT</passenger\_type\_rcd>

<date\_of\_birth></date\_of\_birth>



<flight\_id>

{0D7D5758-94F5-40EE-B4CD-D61469EFF313}

</flight\_id>

<airline\_rcd>GR</airline\_rcd>

<flight\_number>806</flight\_number>

<departure\_date>20080514 00:00:00</departure\_date>

<boarding\_class\_rcd>Y</boarding\_class\_rcd>

<booking\_class\_rcd>T</booking\_class\_rcd>

<inventory\_class\_rcd>A</inventory\_class\_rcd>

<boarded\_class\_rcd></boarded\_class\_rcd>

<record\_locator></record\_locator>

<seat\_number></seat\_number>

<seat\_row>0</seat\_row>

<seat\_column></seat\_column>

<currency\_rcd>GBP</currency\_rcd>

<net\_total>42.8</net\_total>

<tax\_amount>0</tax\_amount>

<fare\_amount>40</fare\_amount>



<yq\_amount>0</yq\_amount>

<base\_ticketing\_fee\_amount>0</base\_ticketing\_fee\_amount>

<ticketing\_fee\_amount>0</ticketing\_fee\_amount>

<reservation\_fee\_amount>0</reservation\_fee\_amount>

<commission\_amount>0</commission\_amount>

<commission\_percentage>0</commission\_percentage>

<refund\_charge>0</refund\_charge>

<refundable\_amount>42.8</refundable\_amount>

<fare\_vat>7</fare\_vat>

<tax\_vat>0</tax\_vat>

<yq\_vat>0</yq\_vat>

<ticketing\_fee\_vat>0</ticketing\_fee\_vat>

<reservation\_fee\_vat>0</reservation\_fee\_vat>

<fare\_amount\_incl>42.8</fare\_amount\_incl>

<tax\_amount\_incl>0</tax\_amount\_incl>

<yq\_amount\_incl>0</yq\_amount\_incl>

<commission\_amount\_incl>0</commission\_amount\_incl>

<ticketing\_fee\_amount\_incl>0</ticketing\_fee\_amount\_incl>

<reservation\_fee\_amount\_incl>



0

</reservation\_fee\_amount\_incl>

<acct\_net\_total>42.8</acct\_net\_total>

<acct\_tax\_amount>0</acct\_tax\_amount>

<acct\_fare\_amount>40</acct\_fare\_amount>

<acct\_yq\_amount>0</acct\_yq\_amount>

<acct\_ticketing\_fee\_amount>

0

</acct\_ticketing\_fee\_amount>

<acct\_reservation\_fee\_amount>

0

</acct\_reservation\_fee\_amount>

<acct\_commission\_amount>0</acct\_commission\_amount>

<acct\_fare\_amount\_incl>42.8</acct\_fare\_amount\_incl>

<acct\_tax\_amount\_incl>0</acct\_tax\_amount\_incl>

<acct\_yq\_amount\_incl>0</acct\_yq\_amount\_incl>

<acct\_commission\_amount\_incl>

0

</acct\_commission\_amount\_incl>



<acct\_ticketing\_fee\_amount\_incl>

0

</acct\_ticketing\_fee\_amount\_incl>

<acct\_reservation\_fee\_amount\_incl>

0

</acct\_reservation\_fee\_amount\_incl>

<acct\_refundable\_amount>42.8</acct\_refundable\_amount>

<flight\_connection\_id></flight\_connection\_id>

<od\_origin\_rcd>GCI</od\_origin\_rcd>

<od\_destination\_rcd>JER</od\_destination\_rcd>

<prorate\_acct\_fare\_amount>0</prorate\_acct\_fare\_amount>

<prorate\_acct\_yq\_amount>0</prorate\_acct\_yq\_amount>

<prorate\_sales\_fare\_amount>0</prorate\_sales\_fare\_amount>

<prorate\_sales\_yq\_amount>0</prorate\_sales\_yq\_amount>

<fare\_id>{E93D3B6A-118D-4217-90F3-8FDCD59DAD5F}</fare\_id>

<fare\_code>T</fare\_code>

<fare\_date\_time>20080422 11:15:23</fare\_date\_time>

<accounting\_refund\_batch>0</accounting\_refund\_batch>



```

<refundable_flag>1</refundable_flag>

<transferable_fare_flag></transferable_fare_flag>

<refund_with_charge_hours>0</refund_with_charge_hours>

<refund_not_possible_hours>1</refund_not_possible_hours>

<e_ticket_flag>1</e_ticket_flag>

<standby_flag></standby_flag>

<exclude_pricing_flag></exclude_pricing_flag>

<passenger_check_in_status_rcd>

    </passenger_check_in_status_rcd>

<priority_code></priority_code>

<agency_code>LBARCGR</agency_code>

<passenger_status_rcd>XX</passenger_status_rcd>

<ticket_number>9244500104543</ticket_number>

<ticketing_date_time>

    20080422 04:17:34

</ticketing_date_time>

<ticketing_by>

    {95CF3057-9F4D-45FA-886A-064AA6BECF24}

</ticketing_by>

```





<public\_fare\_amount>0</public\_fare\_amount>

<public\_fare\_amount\_incl>0</public\_fare\_amount\_incl>

<document\_number></document\_number>

<form\_of\_payment></form\_of\_payment>

<tour\_code></tour\_code>

<fim\_flag></fim\_flag>

<duty\_travel\_flag></duty\_travel\_flag>

<boarding\_pass\_number></boarding\_pass\_number>

<check\_in\_date\_time></check\_in\_date\_time>

<check\_in\_sequence>0</check\_in\_sequence>

<group\_sequence>0</group\_sequence>

<unload\_by></unload\_by>

<unload\_date\_time></unload\_date\_time>

<unload\_comment></unload\_comment>

<number\_of\_pieces>0</number\_of\_pieces>

<baggage\_weight>15</baggage\_weight>

<excess\_baggage\_weight>0</excess\_baggage\_weight>

<check\_in\_baggage\_weight>0</check\_in\_baggage\_weight>

<check\_in\_user\_code></check\_in\_user\_code>



<check\_in\_by></check\_in\_by>

<cancel\_by></cancel\_by>

<cancel\_date\_time></cancel\_date\_time>

<create\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</create\_by>

<create\_date\_time>

20080422 04:15:46

</create\_date\_time>

<update\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</update\_by>

<update\_date\_time>20080422 04:19:21</update\_date\_time>

<refund\_agency\_code>LBARCGR</refund\_agency\_code>

<refund\_by>

{95CF3057-9F4D-45FA-886A-064AA6BECF24}

</refund\_by>

<refund\_date\_time>20080422 11:19:02</refund\_date\_time>



<void\_by></void\_by>

<void\_date\_time></void\_date\_time>

<exchanged\_date\_time></exchanged\_date\_time>

<exchanged\_segment\_id></exchanged\_segment\_id>

<onward\_airline\_rcd></onward\_airline\_rcd>

<onward\_flight\_number></onward\_flight\_number>

<onward\_departure\_date></onward\_departure\_date>

<onward\_destination\_rcd></onward\_destination\_rcd>

<onward\_booking\_class\_rcd></onward\_booking\_class\_rcd>

<restriction\_text>

Restricted Changes - No Refund

</restriction\_text>

<endorsement\_text></endorsement\_text>

<not\_valid\_before\_date></not\_valid\_before\_date>

<not\_valid\_after\_date></not\_valid\_after\_date>

<fare\_line></fare\_line>

<check\_in\_code></check\_in\_code>

<present\_flag></present\_flag>

<through\_fare\_flag></through\_fare\_flag>



<bsp\_checked\_flag></bsp\_checked\_flag>

<adm\_raised\_flag></adm\_raised\_flag>

<adm\_document></adm\_document>

<bsp\_period>0</bsp\_period>

<bsp\_month>0</bsp\_month>

<bsp\_year>0</bsp\_year>

<bsp\_comment></bsp\_comment>

<advanced\_seating\_flag></advanced\_seating\_flag>

<it\_fare\_flag></it\_fare\_flag>

<ticket\_id></ticket\_id>

<coupon\_number>0</coupon\_number>

<fare\_description>0</fare\_description>

<fare\_note>0</fare\_note>

<day\_of\_year>135</day\_of\_year>

<payment\_amount>0</payment\_amount>

<origin\_sequence>1</origin\_sequence>

<destination\_sequence>2</destination\_sequence>

<create\_name>System Administrator</create\_name>

<update\_name>System Administrator</update\_name>



<record\_locator\_display>FO63YD</record\_locator\_display>

</Ticket>

</Tickets>

<TicketTaxes>

<Tax>

<passenger\_segment\_tax\_id>

{C6FDBD2C-EA26-469F-A674-8305930339AD}

</passenger\_segment\_tax\_id>

<tax\_amount>6000</tax\_amount>

<tax\_amount\_incl>6000</tax\_amount\_incl>

<acct\_amount>3</acct\_amount>

<acct\_amount\_incl>3</acct\_amount\_incl>

<sales\_amount>3</sales\_amount>

<sales\_amount\_incl>3</sales\_amount\_incl>

<vat\_percentage>0</vat\_percentage>

<tax\_rcd>IWJR</tax\_rcd>

<accounting\_add\_batch>0</accounting\_add\_batch>

<accounting\_void\_batch>0</accounting\_void\_batch>

<void\_date\_time></void\_date\_time>



<void\_by></void\_by>

<create\_by>

{4BC6062D-B45E-4D81-B83A-934AD3CE09DC}

</create\_by>

<create\_date\_time>20081112 08:55:09</create\_date\_time>

<update\_by>

{4BC6062D-B45E-4D81-B83A-934AD3CE09DC}

</update\_by>

<update\_date\_time>20081112 08:55:09</update\_date\_time>

<passenger\_id>

{8C61540C-2AC3-4977-B3A1-F86DB09C6EC4}

</passenger\_id>

<tax\_id>{4977A297-E89A-4FF3-AD6B-4E79114A0AD2}</tax\_id>

<booking\_segment\_id>

{68EDF0DA-7767-4F26-8A53-FB46721B339B}

</booking\_segment\_id>

<tax\_currency\_rcd>IDR</tax\_currency\_rcd>

<sales\_currency\_rcd>GBP</sales\_currency\_rcd>

<display\_name>INSURANCE</display\_name>



```

<summarize_up>AS</summarize_up>

<coverage_type>GT</coverage_type>

<passenger_type_rcd>ADULT</passenger_type_rcd>

</Tax>

</TicketTaxes>

<Fees>

<Fee>

  <booking_fee_id>

    {36378BEA-92C0-4CC0-9BCF-545AB7D7D9A3}

  </booking_fee_id>

  <fee_amount>2.5</fee_amount>

  <booking_id>

    {C42B58DA-2F13-4D70-A680-B83FDF1ADB3D}

  </booking_id>

  <agency_code>bee</agency_code>

  <currency_rcd>GBP</currency_rcd>

  <acct_fee_amount>2.5</acct_fee_amount>

  <change_comment></change_comment>

  <accounting_add_batch>0</accounting_add_batch>

```



```

<accounting_void_batch>0</accounting_void_batch>

<booking_segment_id></booking_segment_id>

<passenger_segment_service_id>

    </passenger_segment_service_id>

<fee_id>{8AFCF096-2BF8-4C80-ACB8-05E536127793}</fee_id>

<vat_percentage>0</vat_percentage>

<passenger_id></passenger_id>

<fee_amount_incl>2.5</fee_amount_incl>

<acct_fee_amount_incl>2.5</acct_fee_amount_incl>

<charge_amount>2.5</charge_amount>

<charge_amount_incl>2.5</charge_amount_incl>

<charge_currency_rcd>GBP</charge_currency_rcd>

<void_by></void_by>

<void_date_time></void_date_time>

<account_fee_by>

    {A48E7722-DCC6-4340-A572-D105FB42D675}

    </account_fee_by>

<account_fee_date_time>

    20090513 16:56:17
    
```





```

</account_fee_date_time>

<waive_by></waive_by>

<waive_date_time></waive_date_time>

<create_by>

{A48E7722-DCC6-4340-A572-D105FB42D675}

</create_by>

<create_date_time>20090513 09:56:28</create_date_time>

<update_by>

{A48E7722-DCC6-4340-A572-D105FB42D675}

</update_by>

<update_date_time>20090513 09:56:28</update_date_time>

<fee_rcd>SEAT</fee_rcd>

<display_name>Seat Assignment Charge</display_name>

<payment_amount>2.5</payment_amount>

<user_create>njoy ck</user_create>

<user_update>njoy ck</user_update>

<user_void></user_void>

<user_account>njoy ck</user_account>

<user_waive></user_waive>

```



</Fee>

</Fees>

<TicketQuotes>

<Total>

<booking\_segment\_id></booking\_segment\_id>

<origin\_rcd></origin\_rcd>

<destination\_rcd></destination\_rcd>

<departure\_date></departure\_date>

<airline\_rcd></airline\_rcd>

<flight\_number></flight\_number>

<passenger\_type\_rcd>ADULT</passenger\_type\_rcd>

<passenger\_count>3</passenger\_count>

<currency\_rcd>GBP</currency\_rcd>

<charge\_type>FARE</charge\_type>

<charge\_name>Fare</charge\_name>

<charge\_amount>240</charge\_amount>

<charge\_amount\_incl>248.4</charge\_amount\_incl>

<total\_amount>248.4</total\_amount>

<tax\_amount>8.4</tax\_amount>



```

    <sort_sequence>0</sort_sequence>

  </Total>

</TicketQuotes>

<TicketQuotes>

  <Flight>

    <booking_segment_id>

      {824879F9-A8A4-4522-ABBD-EE425E73B56F}

    </booking_segment_id>

    <origin_rcd></origin_rcd>

    <destination_rcd></destination_rcd>

    <departure_date>20080514</departure_date>

    <airline_rcd>GR</airline_rcd>

    <flight_number>806</flight_number>

    <passenger_type_rcd>ADULT</passenger_type_rcd>

    <passenger_count>3</passenger_count>

    <currency_rcd>GBP</currency_rcd>

    <charge_type>FARE</charge_type>

    <charge_name>Fare</charge_name>

    <charge_amount>120</charge_amount>
  
```



<charge\_amount\_incl>128.4</charge\_amount\_incl>

<total\_amount>128.4</total\_amount>

<tax\_amount>8.4</tax\_amount>

<sort\_sequence>0</sort\_sequence>

</Flight>

</TicketQuotes>

<TicketQuotes>

<OD>

<booking\_segment\_id></booking\_segment\_id>

<origin\_rcd>GCI</origin\_rcd>

<destination\_rcd>JER</destination\_rcd>

<departure\_date>20080514</departure\_date>

<airline\_rcd></airline\_rcd>

<flight\_number></flight\_number>

<passenger\_type\_rcd>ADULT</passenger\_type\_rcd>

<passenger\_count>3</passenger\_count>

<currency\_rcd>GBP</currency\_rcd>

<charge\_type>FARE</charge\_type>

<charge\_name>Fare</charge\_name>



<charge\_amount>120</charge\_amount>

<charge\_amount\_incl>128.4</charge\_amount\_incl>

<total\_amount>128.4</total\_amount>

<tax\_amount>8.4</tax\_amount>

<sort\_sequence>0</sort\_sequence>

</OD>

</TicketQuotes>

</Booking>



The Fields used in the Itinerary XML are described in the following tables:

Name	Header/BookingHeader
Description	Shows booking information and booking contact details.
Field name and description	
<u>Name</u>	<u>Description</u>
booking_id	Unique booking ID.
currency_rcd	Booking currency.
Client_profile_id	Unique ID of client profile.
booking_number	Booking Number.
record_locator	Booking Reference.
number_of_adults	Number of adults in the booking.
number_of_children	Number of children in the booking.
number_of_infants	Number of infants in the booking.
language_rcd	Language code.
agency_code	Code of agency that has made the booking.
contact_name	Name of contact person.
contact_email	E-mail address of contact person.
Phone_mobile	Mobile phone number of contact person.
Phone_home	Home phone of contact person.
Phone_business	Business phone of contact person.
received_from	Booking source information.
Phone_fax	Fax number of contact person.
Comment	Booking comments.
Ticket_total	Total ticket price.
Fee_total	Total fees.
Ticket_payment_total	Total ticket payments.
fee_payment_total	Total fee payments.
Group_name	Booking comments.
Group_booking_flag	Name of the group if the booking is a group booking.
agency_name	Identifies whether this booking is a group booking or not.



own_agency_flag	Name of agency that has made the booking.
web_agency_flag	Flag identifying the booking as being made by an airline internal agency.



Name	Itinerary/FlightSegment
Description	Shows detailed flight segment information.
Field name and description	
<u>Name</u>	<u>Description</u>
booking_segment_id	Unique flight segment ID.
flight_id	Flight ID.
flight_connection_id	If the segment is part of a connection (married segment), then the same flight_connection_id will be displayed for each of the segments in the connection that are married together (i.e. same flight_connection_id for flight A-B and flight B-C for an A-C connection via B).
airline_rcd	Airline IATA code.
flight_number	Flight number.
departure_date	Departure date.
departure_time	Departure time.
booking_class_rcd	Booking class code.
boarding_class_rcd	Boarding class code.
segment_status_rcd	Flight segment status:  HK = Confirmed  HL = Waitlisted  XX = Cancelled
booking_id	Unique booking ID.
number_of_units	Number of passengers on each segment booked.
journey_time	Journey time in minutes.
arrival_time	Arrival time.
origin_rcd	Origin airport code.
destination_rcd	Destination airport code.
origin_name	Origin airport name.





destination_name	Destination airport name.
------------------	---------------------------



Name	Passengers/Passenger
Description	Shows detailed passenger information.
Field name and description	
<u>Name</u>	<u>Description</u>
passenger_id	Unique passenger ID.
booking_id	Unique booking ID.
Client_number	Client profile number.
passenger_profile_id	Unique passenger profile ID.
passenger_type_rcd	Passenger type code:  ADULT  CHD  INF
Lastname	Passenger last name.
Firstname	Passenger first name.
date_of_birth	Passenger date of birth.
title_rcd	Passenger title.
gender_type_rcd	Passenger gender code:  M = Male  F = Female
passport_issue_date	Passport issue date.
passport_expiry_date	Passport expiry date.
wheelchair_flag	Identifies if a special service request for a wheelchair exists.
passenger_weight	Passenger weight.
Client_profile_id	Unique client profile ID.



Name	Payments/Payment
Description	Show detailed payment information.
Field name and description	
<u>Name</u>	<u>Description</u>
booking_payment_id	Unique payment ID.
booking_segment_id	Flight segment ID.
booking_id	Booking ID.
voucher_payment_id	Unique voucher payment ID.
form_of_payment_rcd	Form of payment code:  CC = credit card  INV = invoice  CRAG = credit agent  VOUCHER = voucher
currency_rcd	Currency code.
agency_code	Code of agency that has made the booking.
payment_amount	Payment amount.
payment_by	Payment user ID.
payment_date_time	Date of payment.
payment_due_date_time	Payment due date.
document_amount	Payment amount.
void_by	User ID of user that has voided a payment.
void_date_time	Date and time when payment has been voided.
record_locator	Booking reference.
document_number	Document number (credit card number).
form_of_payment_subtype_rcd	Form of payment subtype code. The code is based on form of payment subtype settings in Avantik.



Name	TicketTaxes /Tax
Description	Shows detailed tax information.
Field name and description	
<u>Name</u>	<u>Description</u>
passenger_segment_tax_id	Unique segment tax ID.
tax_amount	Tax amount.
tax_amount_incl	Tax amount include VAT.
tax_rcd	Tax code (based on airline specific tax definition in Avantik).
passenger_id	Unique passenger ID.
tax_id	Unique tax ID.
Booking_segment_id	Flight segment ID.
tax_currency_rcd	Tax currency code.
sales_currency_rcd	Sale currency code.
display_name	Tax display name.



Name	TicketQuote/Total
Description	Shows fare quote information in total view. The information is grouped by passenger_type and charge_type.
Field name and description	
<u>Name</u>	<u>Description</u>
airline_rcd	Airline IATA code.
flight_number	Flight number.
currency_rcd	Currency code.
departure_date	Departure date.
passenger_type_rcd	Passenger type code:  ADULT  CHD  INF
charge_type	Charge type code.
charge_name	Full charge name.
charge_amount	Charge amount.
total_amount	Total amount including tax (i.e. VAT, GST etc.).
tax_amount	Tax amount (i.e. VAT, GST etc.).
passenger_count	Number of passengers.
sort_sequence	Quote sort sequence.



Name	TicketQuote/Flight
Description	Shows fare quote information in flight view. The information is grouped by passenger_type and charge_type.
Field name and description	
<u>Name</u>	<u>Description</u>
airline_rcd	Airline IATA code.
flight_number	Flight number.
currency_rcd	Currency Code.
departure_date	Departure date.
passenger_type_rcd	Passenger type code:  ADULT  CHD  INF
charge_type	Charge type code.
charge_name	Full charge name.
charge_amount	Charge amount.
total_amount	Total amount including tax (i.e. VAT, GST etc.).
tax_amount	Tax amount (i.e. VAT, GST etc.).
passenger_count	Number of passengers.
sort_sequence	Quote sort sequence.



Name	TicketQuote/OD
Description	Shows fare quote information in route view. This quote is used to provide information about connections (married segments). If the booking has no connections (married segments), the XML will look similar to <b>TicketQuotes/Flight</b> .
Field name and description	
<u>Name</u>	<u>Description</u>
airline_rcd	Airline IATA code.
flight_number	Flight number.
currency_rcd	Currency Code.
departure_date	Departure date.
passenger_type_rcd	Passenger type code:  ADULT  CHD  INF
charge_type	Charge type code.
charge_name	Full charge name.
charge_amount	Charge amount.
total_amount	Total amount including tax (i.e. VAT, GST etc.).
tax_amount	Tax amount (i.e. VAT, GST etc.).
passenger_count	Number of passengers.
sort_sequence	Quote sort sequence.



Name	Fees/Fee
Description	Shows detailed booking fee information.
Field name and description	
Name	Description
Booking_fee_id	Unique booking fee ID.
fee_amount_incl	Fee amount including VAT.
charge_amount	Charge amount.
charge_amount_incl	Charge amount including VAT.
payment_amount	Fee payment amount.
Booking_id	Booking ID.
passenger_id	Passenger ID.
currency_rcd	Fee currency code.
fee_id	Fee ID.
vat_percentage	Fee VAT percentage.
fee_rcd	Fee code.
display_name	Fee display name.
void_date_time	Fee void date time.
void_by	User ID of user that has voided a payment.
payment_amount	Fee payment amount.
Booking_segment_id	Flight segment ID.
agency_code	Code of agency that has made the booking.





Name	Tickets/Ticket
Description	Shows detailed ticket information.
Field name and description	
<u>Name</u>	<u>Description</u>
passenger_id	Passenger ID.
booking_segment_id	Flight segment ID.
origin_rcd	Origin airport code.
destination_rcd	Destination airport code.
Lastname	Passenger last name.
Firstname	Passenger first name.
gender_type_rcd	Passenger gender code:  M = Male  F = Female
title_rcd	Passenger title code.
passenger_type_rcd	Passenger type code:  ADULT  CHD  INF
flight_id	Flight ID.
airline_rcd	Airline IATA code.
flight_number	Flight number.
departure_date	Flight departure date.
booking_class_rcd	Booking class code.
boarding_class_rcd	Boarding class code.
seat_number	Seat number.
seat_row	Seat row.
seat_column	Seat column.



net_total	Total ticket price (including taxes, YQ and VAT).
tax_amount	Tax amount.
fare_amount	Fare amount.
fare_vat	Fare VAT amount.
yq_vat	YQ VAT amount.
yq_amount	YQ amount.
fare_amount_incl	Fare amount including VAT.
tax_amount_incl	Tax amount including VAT.
yq_amount_incl	YQ amount including VAT.
fare_id	Fare ID.
fare_code	Fare basis code.
e_ticket_flag	Identifies whether this ticket is an e-ticket.
standby_flag	Identifies whether this ticket is a standby ticket.
transferable_fare_flag	Identifies whether this ticket is a transferable ticket.
agency_code	Code of agency that has made the booking.
Ticket_number	Ticket number.
Ticketing_date_time	Ticketing date.
ticketing_by	Ticketing by.
Check_in_sequence	Flight check-in sequence.
Group_sequence	Flight check-in group sequence.
number_of_pieces	Number of bags.
baggage_weight	Baggage allowance.
excess_baggage_weight	Excess baggage weight.
payment_amount	Ticket payment amount.
Check_in_baggage_weight	Check-in baggage weight.



## 9. APPENDIX 5: SAVE BOOKING XML

The Save booking XML consists of two main sections, **BookingHeader**, **Passenger** and **Payment**.

<Booking>

<BookingHeader>

<booking\_id>

ac28d00a-d42d-11de-8873-00045a49d651

</booking\_id>

<phone\_mobile>1</phone\_mobile>

<phone\_home>1</phone\_home>

<phone\_business>1</phone\_business>

<phone\_search>1</phone\_search>

<lastname>Test</lastname>

<firstname>Ang</firstname>

<address\_line1>st</address\_line1>

<address\_line2>st</address\_line2>

<city>st</city>

<zip\_code>st</zip\_code>

<title\_rcd>MR</title\_rcd>

<country\_rcd>TH</country\_rcd>



<email>test@tik.com</email>

<apply\_client\_profile\_flag>1</apply\_client\_profile\_flag>

</BookingHeader>

<Passenger>

<passenger\_id>

ac28d00b-d42d-11de-8873-00045a49d651

</passenger\_id>

<passenger\_profile\_id>

ac28d00b-d42d-11de-8873-00045a49d651

</passenger\_profile\_id>

<booking\_id>

ac28d00a-d42d-11de-8873-00045a49d651

</booking\_id>

<passenger\_type\_rcd>ADULT</passenger\_type\_rcd>

<lastname>TEST</lastname>

<firstname>ANG</firstname>

<title\_rcd>MR</title\_rcd>

<gender\_type\_rcd>M</gender\_type\_rcd>

<nationality\_rcd>AL</nationality\_rcd>



<date\_of\_birth>19870422</date\_of\_birth>

<passport\_issue\_date>20100531</passport\_issue\_date>

<passport\_expiry\_date>20161231</passport\_expiry\_date>

<known\_traveler\_number>20099975</known\_traveler\_number>

</Passenger>

<!--Sample Credit Card Payment Xml Sample -->

<Payment>

<form\_of\_payment\_rcd>CC</form\_of\_payment\_rcd>

<form\_of\_payment\_subtype\_rcd>

VISA

</form\_of\_payment\_subtype\_rcd>

<NameOnCard>Ang Test</NameOnCard>

<CreditCardNumber>4444333322221111</CreditCardNumber>

<CVV>123</CVV>

<IssueMonth>1</IssueMonth>

<IssueYear>2008</IssueYear>

<IssueNumber></IssueNumber>

<ExpiryMonth>1</ExpiryMonth>



<ExpiryYear>2011</ExpiryYear>

<Addr1>st</Addr1>

<Addr2>st</Addr2>

<Street></Street>

<State></State>

<City>st</City>

<County></County>

<ZipCode>st</ZipCode>

<Country>TH</Country>

</Payment>

<!--Sample Credit Agency Payment Xml Sample -->

<Payment>

<form\_of\_payment\_rcd>CRAGT</form\_of\_payment\_rcd>

</Payment>

<!--Sample Single Voucher Payment Xml Sample -->

<Payment>

<form\_of\_payment\_rcd>VOUCHER</form\_of\_payment\_rcd>

<voucher\_number>3700002465</voucher\_number>

<password>4256</password>

</Payment>



<!--Sample Voucher Payment Xml for Multiple Voucher Sample -->

```
<Payment>
  <form_of_payment_rcd>VOUCHER</form_of_payment_rcd>
  <voucher_number>3700003865</voucher_number>
  <password>7356</password>
</Payment>
```

```
<Payment>
  <form_of_payment_rcd>VOUCHER</form_of_payment_rcd>
  <voucher_number>3700003866</voucher_number>
  <password>3423</password>
</Payment>
```

<!--Sample Voucher Payment Xml with Credit Card Payment Xml for MultipleFOP Sample -->

```
<Payment>
  <form_of_payment_rcd>VOUCHER</form_of_payment_rcd>
  <voucher_number>3700003865</voucher_number>
  <password>7356</password>
</Payment>

<Payment>

  <form_of_payment_rcd>CC</form_of_payment_rcd>

  <form_of_payment_subtype_rcd>VISA</form_of_payment_subtype_rcd>

  <NameOnCard>Ang Test</NameOnCard>

  <CreditCardNumber>4444333322221111</CreditCardNumber>

  <CVV>123</CVV>
```



```

<IssueMonth>1</IssueMonth>

<IssueYear>2008</IssueYear>

<IssueNumber></IssueNumber>

<ExpiryMonth>1</ExpiryMonth>

<ExpiryYear>2011</ExpiryYear>

<Addr1>st</Addr1>

<Addr2>st</Addr2>

<Street></Street>

<State></State>

<City>st</City>

<County></County>

<ZipCode>st</ZipCode>

<Country>TH</Country>

</Payment>

</Booking>

```

The Fields used in the Save Booking XML are described in the following tables:

Name	BookingHeader
------	---------------





<b>Description</b>	Shows booking information and booking contact details.
<b>Field name and description</b>	
<b><u>Name</u></b>	<b><u>Description</u></b>
booking_id	Unique booking ID.
phone_mobile	Mobile phone number of contact person.
phone_home	Home phone of contact person.
phone_business	Business phone of contact person.
lastname	Last name of contact person.
firstname	First name of contact person.
address_line1	Address line 1 of contact person.
address_line2	Address line 2 of contact person.
city	City of contact person.
zip_code	Zip code of contact person.
title_rcd	Title of contact person.
country_rcd	Country of contact person as ISO 3166-1-alpha-2 code.
email	Email address of contact person.



Name	Passengers/Passenger
Description	Shows detailed passenger information.
Field name and description	
<u>Name</u>	<u>Description</u>
passenger_id	Unique passenger ID.
booking_id	Unique booking ID.
passenger_type_rcd	Passenger type code:  ADULT  CHD  INF
Lastname	Passenger last name.
Firstname	Passenger first name.
title_rcd	Passenger title.
gender_type_rcd	Passenger gender code  M = Male  F = Female
nationality_rcd	Passenger nationality as ISO 3166-1-alpha-2 code.



Name	Payment
Description	Shows detailed payment information.
Field name and description	
<u>Name</u>	<u>Description</u>
NameOnCard	Name show on credit card.
CreditCardNumber	Credit Card number.
IssueMonth	Credit card issue month.
IssueYear	Credit card issue year.
IssueNumber	Credit card issue number.
ExpiryMonth	Credit card expiry month.
ExpiryYear	Credit card expiry year.
Addr1	Credit card address line 1.
Addr2	Credit card address line 2.
Street	Credit card street.
State	Credit card state.
City	Credit card city.
County	Credit card country.
ZipCode	Credit card zip code.
Country	Credit card country.
CVV	Credit card CVV code.
form_of_payment_subtype_rcd	Form of payment subtype code. The code is based on form of payment subtype settings in Avantik.



## 10. APPENDIX 6: AGENCY XML

<Agencies>

<Agency>

<agency\_code>TESTAGENT</agency\_code>

<b2b\_credit\_card\_payment\_flag>

1

</b2b\_credit\_card\_payment\_flag>

<b2b\_credit\_agency\_and\_invoice\_flag>

1

</b2b\_credit\_agency\_and\_invoice\_flag>

<booking\_payment>179279.69</booking\_payment>

<agency\_account>11063361</agency\_account>

</Agency>

</Agencies>

Name	Agencies/Agency
Description	Shows Agency information.
Field name and description	
Name	Description
agency_code	Code assigned to agency.
b2b_credit_card_payment_flag	Flag indicating whether the agency is allowed to make payment by credit card.
b2b_credit_agency_and_invoice_flag	Flag indicating whether the agency is allowed to make payment by credit agency account.
booking_payment	Amount of credit agency account payment for this PNR.



agency_account	Total credit agency account amount.
----------------	-------------------------------------

## 11. APPENDIX 7: ERROR XML

The Error XML consists of a single section.

<error>

<code>000</code>

<message>Successful transaction request</message>

<referenceId>ReferenceId</referenceId>

</error>

The Fields used in the Error XML are described in the following table:

Name	Error
Description	Shows error and success information.
Field name and description	
<u>Name</u>	<u>Description</u>
Code	Error code.
Message	Error message.
referenceId	Transaction reference ID.





The following Error Numbers are used by the Avantik WebAPI:

ERROR NUMBER	ERROR DESCRIPTION
000	Successful transaction request.
100	Fail To Initialize API Service.
101	Login Information Required.
102	Service needs To Be Initialized.
103	General error.
104	This Agency is not allowed to create booking.
105	Agency information not found.
106	This function call is not allowed.
107	Infant over limit.
200	Origin and destination required.
201	Travel date range required – Departure from yyyyymmdd to yyyyymmdd.
202	Passenger type required - At least one adult is required in the booking.
207	Invalid Voucher.
300	Flight XML not found.
301	Invalid flightAdd - FlightAvailability needs to be called before flightAdd.
302	Your Selected Flight is full.
303	The selected class is not open.
304	Your selected flight is closed.
305	One of your selected flights can not be added.
400	No booking information found.
401	Find US segment.
500	Payment XML only needs one node.
501	The booking could not be saved.
502	Form of payment not found.
503	Not Enough Credit Balance.
504	Agency Information not found.
505	Payment by credit card is not allowed.
506	Payment by credit agency account is not allowed.



507	Internal inventory lock is timeout.
508	Duplicate seat.
509	Data access failed.
510	Booking in-use.
511	Error read booking
512	Payment request timeout.
513	Payment by voucher is not allowed.
514	Voucher not found.
515	Credit card XML must has only one section.
516	Multiple form of payments not allowed.
601	Can't return booking Url.
602	No response from server.
701	SSR code not found.
702	Flight information not found.
800	Please supply origin.
801	Please supply destination.
802	Please supply flight Id.
803	No seat information found.
900	Agency code is required.
901	Booking segment Id is required.
902	Passenger Id is required.
903	No Mapping information found.
904	Baggage fee not found.
1000	Invalid input paeameter.
1001	Can't add more than two items of baggage information.
1002	GetBaggageFee have to be successfully call before process this function call.
1200	One of the request supply an invalid BookingSegmentId.
1201	One of the request supply an invalid PassengerId.
1203	Seat assignment failed.
1204	No request information found.
1300	One of the request supply an invalid PassengerSegmentServiceId.
1302	One of the request supply an invalid PassengerId.





1303	One of the request supply an invalid BookingSegmentId.
1304	One of the request supply an invalid OriginRcd.
1305	One of the request supply an invalid DestinationRcd.
1306	One of the request supply an invalid SpecialServiceRcd.
1307	One of the request supply an invalid ServiceText.
1308	Duplicate special service selected.
1309	No request information found.
1400	No PassengerSegmentServiceId match to remove.
1401	No Passenger Segment Service found.
1500	Client number is required.
1501	Password is required.
1503	Login Failed.
1600	Client Logon is required.
1601	Login Failed.
1700	No Booking information Found.
1800	No Booking Information found.
1900	PassengerId is required.
1901	BookingSegmentId is required.
1902	FeeId is required.
1903	FeeCategoryRcd is required.
1904	FeeRcd is required.
1905	CurrencyRcd is required.
1906	ChargeCurrencyRcd is required.
2000	Remove Fee request parameter is required.
3000	Remark Type Rcd is required.
3001	Remark Text is required.
3002	Booking remark id is required.
3003	Booking id is required.
3004	Failed adding remark.
4000	Remove Remark request parameter is required.
4001	Delete remark failed.
5000	Agency Code Required.
5001	User Code Required



5002	Password Required.
5003	Agency Type Required.
5004	Agency name Required.
5005	Legal name Required.
5006	IATA number Required.
5007	Tax Id Required.
5008	Phone Required.
5009	Fax Required.
5010	Email Required.
5011	Website Address Required.
5012	Currency Code Required.
5013	Language Code Required.
5014	Contact Person Required.
5015	Address line1 Required.
5016	Address line2 Required.
5017	Street Required.
5018	State Required.
5019	Province Required.
5020	City Required.
5021	Zip Code Required.
5022	Country code Required.
5023	Lastname Required.
5024	Firstname Required.
5025	Title Required.
5026	Agency Password Required.
5027	Comment Required.
5028	Register Failed.
5029	Authentication Failed.
5030	Agency not found.
6000	Voucher Invalid.
6101	Object ClientRequest is null.
6102	TitleRcd required.
6103	Firstname required.



6104	Lastname required
6105	DateOfBirth required
6106	Invalid date of birth
6107	PassengerTypeRcd required
6108	ContactEmail required
6109	ContactPassword required
6110	GenderTypeRcd required
6111	Contact email is already in use.
6302	TitleRcd required.
6303	Firstname required.
6304	Lastname required
6305	PassengerTypeRcd required.
6307	PassengerRole required.
6308	DateOfBirthRequired.
6309	Invalid DateOfBirth.
6310	GenderTypeRcd Required.
6310	Object PassengerRequest is null.
7000	Save client profile failed.
7001	Invalid Voucher.
7002	Voucher password not matches.
7003	Voucher currency not matches.
7004	Voucher status is not open.
7005	Duplicate vouchers.
7006	Voucher amount is not enough.
7009	Number of vouchers over limit.
7010	Multiple voucher payment not allowed.
8000	Save passenger profile failed.
9000	Client profile not found.
9001	Passenger profile not found.
9101	Invalid Flight.
9102	Invalid Agency information.
9103	Passengers are required.
9104	Number of infants is greater than number of adults.



9105	AgencyCode is required.
9106	Language is required.
9107	CurrencyCode is required.
9108	Password is required.
9109	FlightSegment is required.
9110	FlightId is required.
9111	FareId is required
9112	OriginRcd is required
9113	DestinationRcd is required
9114	BoardingClassRcd is required
9115	BookingClassRcd is required
9116	TransitAirportRcd is required.
9117	TransitBoardingClassRcd is required.
9118	TransitBookingClassRcd is required.
9119	TransitDepartureDate is required.
XX	Internal connection timeout.
0	Card type does not support CVV.



---

## 12. APPENDIX 8: SPECIAL SERVICE FEE XML

```
<service>

  <service_fee>

    <special_service_rcd>WCHC</special_service_rcd>

    <display_name>

      Wheelchair (Carry) (SSR) (Language Mapping - EN)

    </display_name>

    <origin_rcd>FUK</origin_rcd>

    <destination_rcd>KIX</destination_rcd>

    <od_origin_rcd>FUK</od_origin_rcd>

    <od_destination_rcd>KIX</od_destination_rcd>

    <booking_class_rcd>T</booking_class_rcd>

    <fare_code>TCOOL</fare_code>

    <airline_rcd>T1</airline_rcd>

    <flight_number>152</flight_number>

    <departure_date>2012-12-05T00:00:00</departure_date>

    <currency_rcd>JPY</currency_rcd>

    <fee_amount>7000</fee_amount>

    <fee_amount_incl>7000</fee_amount_incl>

    <total_fee_amount>7000</total_fee_amount>

    <total_fee_amount_incl>7000</total_fee_amount_incl>

    <service_on_request>True</service_on_request>
```



```

</service_fee>

<service_fee>

    <special_service_rcd>BSTF</special_service_rcd>

    <display_name>Bus transfer-EN</display_name>

    <origin_rcd>FUK</origin_rcd>

    <destination_rcd>KIX</destination_rcd>

    <od_origin_rcd>FUK</od_origin_rcd>

    <od_destination_rcd>KIX</od_destination_rcd>

    <booking_class_rcd>T</booking_class_rcd>

    <fare_code>TCOOL</fare_code>

    <airline_rcd>T1</airline_rcd>

    <flight_number>152</flight_number>

    <departure_date>2012-12-05T00:00:00</departure_date>

    <currency_rcd>JPY</currency_rcd>

    <fee_amount>6000</fee_amount>

    <fee_amount_incl>6000</fee_amount_incl>

    <total_fee_amount>6000</total_fee_amount>

    <total_fee_amount_incl>6000</total_fee_amount_incl>

    <service_on_request>True</service_on_request>

</service_fee>

</service>

```



Name	service/service_fee
Description	Show a list of Special Service with fee amount.
Field name and description	
<u>Name</u>	<u>Description</u>
special_service_rcd	Special service code
display_name	Display name of the special service
origin_rcd	Origin Airport.
destination_rcd	Destination Airport.
od_origin_rcd	First destination airport
od_destination_rcd	Final destination airport
booking_class_rcd	Booking Class
fare_code	Fare Code
airline_rcd	Airline Code
flight_number	Flight Number
departure_date	Departure Date
currency_rcd	Currency Code
fee_amount	Fee amount
fee_amount_incl	Fee Amount Include Vat
total_fee_amount	Total Fee Amount per number of passenger
total_fee_amount_incl	Total Fee Amount per number of passenger include Var
service_on_request	To set this SSR to be service on request.



### 13. APPENDIX 9: BAGGAGE FEE XML

```
<?xml version="1.0" encoding="utf-16"?>

<BaggageFees>

  <Fee>

    <baggage_fee_option_id>

      00000000-0000-0000-0000-000000000000

    </baggage_fee_option_id>

    <passenger_id>7fa31e58-b333-4454-978a-bc00a1cc395e</passenger_id>

    <booking_segment_id>a604d6f5-2a5d-4abd-bc35-85f0f1509e01</booking_segment_id>

    <fee_id>02859c97-03ba-42f4-a539-eb7687057073</fee_id>

    <fee_rcd>BAGF</fee_rcd>

    <fee_category_rcd>BAGSALES</fee_category_rcd>

    <currency_rcd>JPY</currency_rcd>

    <display_name>Baggage Fee-EN</display_name>

    <number_of_units>2</number_of_units>

    <fee_amount>3000</fee_amount>

    <fee_amount_incl>3000</fee_amount_incl>

    <total_amount>3000</total_amount>

    <total_amount_incl>3000</total_amount_incl>
```





<vat\_percentage>0</vat\_percentage>

</Fee>

<Fee>

<baggage\_fee\_option\_id>

00000000-0000-0000-0000-000000000000

</baggage\_fee\_option\_id>

<passenger\_id>7fa31e58-b333-4454-978a-bc00a1cc395e</passenger\_id>

<booking\_segment\_id>a604d6f5-2a5d-4abd-bc35-85f0f1509e01</booking\_segment\_id>

<fee\_id>54c4aa1a-a790-48cf-8cbc-442a0e109f38</fee\_id>

<fee\_rcd>BAGF</fee\_rcd>

<fee\_category\_rcd>BAGSALES</fee\_category\_rcd>

<currency\_rcd>JPY</currency\_rcd>

<display\_name>Baggage Fee-EN</display\_name>

<number\_of\_units>3</number\_of\_units>

<fee\_amount>4000</fee\_amount>

<fee\_amount\_incl>4000</fee\_amount\_incl>

<total\_amount>7000</total\_amount>

<total\_amount\_incl>7000</total\_amount\_incl>

<vat\_percentage>0</vat\_percentage>



</Fee>

<Fee>

<baggage\_fee\_option\_id>

00000000-0000-0000-0000-000000000000

</baggage\_fee\_option\_id>

<passenger\_id>7fa31e58-b333-4454-978a-bc00a1cc395e</passenger\_id>

<booking\_segment\_id>a604d6f5-2a5d-4abd-bc35-85f0f1509e01</booking\_segment\_id>

<fee\_id>a1c766a0-5b44-49ce-84d4-7478b795568f</fee\_id>

<fee\_rcd>BAGF</fee\_rcd>

<fee\_category\_rcd>BAGSALES</fee\_category\_rcd>

<currency\_rcd>JPY</currency\_rcd>

<display\_name>Baggage Fee-EN</display\_name>

<number\_of\_units>4</number\_of\_units>

<fee\_amount>5000</fee\_amount>

<fee\_amount\_incl>5000</fee\_amount\_incl>

<total\_amount>12000</total\_amount>

<total\_amount\_incl>12000</total\_amount\_incl>

<vat\_percentage>0</vat\_percentage>

</Fee>



</BaggageFees>



Name	BaggageFees/Fee
Description	Show a list of available baggage that can be select.
Field name and description	
Name	Description
passenger_id	ID of the passenger that want select the baggage.
booking_segment_id	Id of segment that want to display a baggage list.
fee_id	Fee Id
fee_rcd	Fee Code
fee_category_rcd	Fee Category Code
currency_rcd	Currency Code
display_name	Fee Display name
number_of_units	Number of baggage unit
fee_amount	Baggage Fee based on the number of unit.
fee_amount_incl	Baggage Fee include vat based on the number of unit.
total_amount	
total_amount_incl	
vat_percentage	Vat percentage.



## 14. APPENDIX 10: PASSENGER PROFILE READ

<MainProfile>

<status\_code>A</status\_code>

<client\_password></client\_password>

<company\_flag>0</company\_flag>

<profile\_on\_hold\_date\_time>0001-01-01T00:00:00</profile\_on\_hold\_date\_time>

<profile\_on\_hold\_comment></profile\_on\_hold\_comment>

<profile\_on\_hold\_by>00000000-0000-0000-0000-000000000000</profile\_on\_hold\_by>

<company\_client\_profile\_id>00000000-0000-0000-0000-000000000000</company\_client\_profile\_id>

<ffp\_total>1950</ffp\_total>

<ffp\_period>0</ffp\_period>

<ffp\_balance>435</ffp\_balance>

<client\_type\_rcd>IND</client\_type\_rcd>

<member\_since\_date>0001-01-01T00:00:00</member\_since\_date>

<member\_level\_display\_name></member\_level\_display\_name>

<keep\_point>0</keep\_point>

<origin\_rcd />

<destination\_rcd />

<lock\_on\_to\_book\_flag>0</lock\_on\_to\_book\_flag>



```

<lock_on_to_report_flag>0</lock_on_to_report_flag>

<allow_ffp_redemption_flag>>false</allow_ffp_redemption_flag>

<passenger_id>00000000-0000-0000-0000-000000000000</passenger_id>

<booking_id>00000000-0000-0000-0000-000000000000</booking_id>

<passenger_profile_id>2e907d41-0535-4df9-b19e-1fbd12d81515</passenger_profile_id>

<passenger_status_rcd />

<employee_number></employee_number>

<wheelchair_flag>0</wheelchair_flag>

<vip_flag>0</vip_flag>

<passenger_role_rcd />

<member_level_rcd>PRM</member_level_rcd>

<member_number></member_number>

<window_seat_flag>0</window_seat_flag>

<company_phone_home />

<company_phone_business />

<company_phone_mobile />

<redress_number></redress_number>

<pnr_name />

<nationality_display_name />

```



```

<passport_issue_country_display_name />

<title_rcd>MISS</title_rcd>

<lastname>PROMRUKSA</lastname>

<firstname>RATCHANEE</firstname>

<middlename></middlename>

<language_rcd>EN</language_rcd>

<nationality_rcd></nationality_rcd>

<passenger_weight>0</passenger_weight>

<gender_type_rcd>M</gender_type_rcd>

<passenger_type_rcd>ADULT</passenger_type_rcd>

<client_profile_id>f2c8e43b-b4a8-4f92-88c5-01bce0202d30</client_profile_id>

<client_number>60000058</client_number>

<address_line1>ddd</address_line1>

<address_line2>dddd</address_line2>

<state>gggg</state>

<district>BKK</district>

<province>BKK</province>

<zip_code>12345</zip_code>

<po_box>12345</po_box>

```



```

<country_rcd>TH</country_rcd>

<street>BKK</street>

<city>bkk</city>

<document_type_rcd />

<document_number></document_number>

<residence_country_rcd></residence_country_rcd>

<passport_number></passport_number>

<passport_issue_date>0001-01-01T00:00:00</passport_issue_date>

<passport_expiry_date>0001-01-01T00:00:00</passport_expiry_date>

<passport_issue_place></passport_issue_place>

<passport_birth_place />

<date_of_birth>0001-01-01T00:00:00</date_of_birth>

<passport_issue_country_rcd></passport_issue_country_rcd>

<contact_name>Paew</contact_name>

<contact_email>ratchanee.promruksa@bravo.aero</contact_email>

<mobile_email></mobile_email>

<phone_mobile>55555555</phone_mobile>

<phone_home></phone_home>

<phone_fax></phone_fax>

```





```

<phone_business></phone_business>

<received_from></received_from>

<create_by>00000000-0000-0000-0000-000000000000</create_by>

<create_date_time>0001-01-01T00:00:00</create_date_time>

<update_by>00000000-0000-0000-0000-000000000000</update_by>

<update_date_time>0001-01-01T00:00:00</update_date_time>

</MainProfile>

<Client>

<status_code>A</status_code>

<client_password></client_password>

<company_flag>0</company_flag>

<profile_on_hold_date_time>0001-01-01T00:00:00</profile_on_hold_date_time>

<profile_on_hold_comment></profile_on_hold_comment>

<profile_on_hold_by>00000000-0000-0000-0000-000000000000</profile_on_hold_by>

<company_client_profile_id>00000000-0000-0000-0000-000000000000</company_client_profile_id>

<ffp_total>0</ffp_total>

<ffp_period>0</ffp_period>

<ffp_balance>0</ffp_balance>

<client_type_rcd></client_type_rcd>

```



```

<member_since_date>0001-01-01T00:00:00</member_since_date>

<member_level_display_name></member_level_display_name>

<keep_point>0</keep_point>

<origin_rcd />

<destination_rcd />

<lock_on_to_book_flag>0</lock_on_to_book_flag>

<lock_on_to_report_flag>0</lock_on_to_report_flag>

<allow_ffp_redemption_flag>>false</allow_ffp_redemption_flag>

<passenger_id>8cf081ff-0129-4bde-8f3b-d918ad59842c</passenger_id>

<booking_id>00000000-0000-0000-0000-000000000000</booking_id>

<passenger_profile_id>689f52a9-a4fc-4033-8dc1-163884d2a709</passenger_profile_id>

<passenger_status_rcd />

<employee_number></employee_number>

<wheelchair_flag>0</wheelchair_flag>

<vip_flag>0</vip_flag>

<passenger_role_rcd>FATHER</passenger_role_rcd>

<member_level_rcd>PRM</member_level_rcd>

<member_number></member_number>

<window_seat_flag>0</window_seat_flag>

```



```

<company_phone_home />

<company_phone_business />

<company_phone_mobile />

<redress_number></redress_number>

<pnr_name />

<nationality_display_name />

<passport_issue_country_display_name />

<title_rcd>MISS</title_rcd>

<lastname>RUANGPAISAN</lastname>

<firstname>SAMLEE</firstname>

<middlename>LULEE</middlename>

<language_rcd />

<nationality_rcd>AQ</nationality_rcd>

<passenger_weight>0</passenger_weight>

<gender_type_rcd>F</gender_type_rcd>

<passenger_type_rcd>YOUTH</passenger_type_rcd>

<client_profile_id>f2c8e43b-b4a8-4f92-88c5-01bce0202d30</client_profile_id>

<client_number>60000058</client_number>

<address_line1></address_line1>

```



```

<address_line2></address_line2>

<state></state>

<district></district>

<province></province>

<zip_code></zip_code>

<po_box></po_box>

<country_rcd>TH</country_rcd>

<street></street>

<city></city>

<document_type_rcd>I</document_type_rcd>

<document_number></document_number>

<residence_country_rcd></residence_country_rcd>

<passport_number>15421522</passport_number>

<passport_issue_date>2015-11-25T09:52:51+07:00</passport_issue_date>

<passport_expiry_date>2016-02-29T09:52:51+07:00</passport_expiry_date>

<passport_issue_place>Buriram</passport_issue_place>

<passport_birth_place>Buriram</passport_birth_place>

<date_of_birth>1982-11-01T00:00:00+07:00</date_of_birth>

<passport_issue_country_rcd>DZ</passport_issue_country_rcd>

```



```

<contact_name></contact_name>

<contact_email></contact_email>

<mobile_email></mobile_email>

<phone_mobile></phone_mobile>

<phone_home></phone_home>

<phone_fax></phone_fax>

<phone_business></phone_business>

<received_from></received_from>

<create_by>35e1292d-ccf0-4408-84a5-f0f4599cb95a</create_by>

<create_date_time>2015-11-25T09:54:16+07:00</create_date_time>

<update_by>35e1292d-ccf0-4408-84a5-f0f4599cb95a</update_by>

<update_date_time>2015-11-25T09:54:16+07:00</update_date_time>

</Client>

```

Name	ClientProfile/MainProfile
Description	Main client profile information.
Field name and description	
<u>Name</u>	<u>Description</u>
client_profile_id	Client profile id. Unique Id of the mail client profile. All the member will have the same ID.
client_password	Client password.
firstname	Member firstname.
middlename	Member middle name.
lastname	Member lastname.



date_of_birth	Birth date.
title_rcd	Member Title.
gender_type_rcd	Gender Code.  <b>F</b> = Female  <b>M</b> = Male
passenger_type_rcd	Passenger type code.  <b>ADULT</b> = Adult  <b>CHD</b> = Child  <b>INF</b> = Infant
nationality_rcd	Nationality Code.
passport_number	Passport Number.
passport_issue_date	Passport issue date.
passport_expiry_date	Passport expiry date.
passport_issue_place	Place of issuing the passport.
passport_birth_place	Profile member birth place.
passport_issue_country_rcd	Country code of issuing country.
wheelchair_flag	Identify whether the member required wheelchair.  <b>0</b> = not required.  <b>1</b> = required.
window_seat_flag	Identify that the member is prefer to set close to the window.  <b>0</b> = Can be any place.  <b>1</b> = Prefer.
vip_flag	Identify that the member of the profile is a VIP member.  <b>0</b> = No



	<b>1</b> = Yes
passenger_weight	Passenger weight.
document_type_rcd	Document Type.
passenger_role_rcd	Passenger role code
member_level_rcd	Member level code
member_number	Member code.
country_rcd	Country code.
client_number	Client profile number.
Language_rcd	Language code.
address_line1	Address line 1.
address_line2	Address line 2
state	State
zip_code	Post code.
country_rcd	Country code.
street	Stree.
City	City.
contact_email	Main client profile contact email address.
phone_mobile	Mobile phone number.
phone_home	Home phone number.
phone_business	Business phone number.
redress_number	Department of Homeland Security's Traveler Redress Inquiry Program (TRIP).
status_code	Client profile status code.  <b>A</b> = Active  <b>I</b> = Inactive  <b>D</b> = Delete.
profile_on_hold_date_time	Date and time that client account is on hold.
client_type_rcd	Client type code.  <b>COR</b> = Corporate profile. User of B2E.



	<b>IND</b> = Individual profile. For passenger client profile.  <b>STF</b> = Staff passenger.
member_since_date	Member since date.
company_flag	Identify if this client profile is a B2E profile.
ffp_total	Total point that client have
ffp_balance	Balance point that client can use to redemption





Name	ClientProfile/Client
Description	Show a list of client profile member.
Field name and description	
<u>Name</u>	<u>Description</u>
passenger_id	Passenger id
passenger_profile_id	Passenger profile id. This ID is required if you want to book the passenger profile as a passenger in the booking.
client_profile_id	Client profile id. Unique Id of the mail client profile. All the member will have the same ID.
firstname	Member firstname.
lastname	Member lastname.
date_of_birth	Birth date.
title_rcd	Member Title.
gender_type_rcd	Gender Code.  <b>F</b> = Female  <b>M</b> = Male
passenger_type_rcd	Passenger type code.
nationality_rcd	Nationality Code.
passport_number	Passport Number.
passport_issue_date	Passport issue date.
passport_expiry_date	Passport expiry date.
passport_issue_place	Place of issuing the passport.
passport_birth_place	Profile member birth place.
wheelchair_flag	Identify whether the member required wheelchair.  <b>0</b> = not required.  <b>1</b> = required.
window_seat_flag	Identify that the member is prefer to set close to the



	<p>window.</p> <p><b>0</b> = Can be any place.</p> <p><b>1</b> = Prefer.</p>
<b>vip_flag</b>	<p>Identify that the member of the profile is a VIP member.</p> <p><b>0</b> = No</p> <p><b>1</b> = Yes</p>
<b>passenger_weight</b>	Passenger weight.
<b>document_type_rcd</b>	Document Type.
<b>passenger_role_rcd</b>	Passenger role code
<b>member_level_rcd</b>	Member level code
<b>member_number</b>	Member code.
<b>country_rcd</b>	Country code.
<b>client_number</b>	Client profile number.



## 15. APPENDIX 11: GET SEATMAP

<SeatMaps>

<Seat>

<flight\_id>52329ada-4f14-49f7-8eb6-10e8cc8a25e9</flight\_id>

<free\_seating\_flag>0</free\_seating\_flag>

<origin\_rcd>KIX</origin\_rcd>

<destination\_rcd>FUK</destination\_rcd>

<aircraft\_configuration\_code>Y146TEST</aircraft\_configuration\_code>

<number\_of\_bays>33</number\_of\_bays>

<boarding\_class\_rcd>Y</boarding\_class\_rcd>

<number\_of\_rows>30</number\_of\_rows>

<number\_of\_columns>11</number\_of\_columns>

<layout\_row>0</layout\_row>

<layout\_column>0</layout\_column>

<seat\_row>0</seat\_row>

<seat\_column>D</seat\_column>

<stretcher\_flag>0</stretcher\_flag>

<handicapped\_flag>0</handicapped\_flag>

<no\_child\_flag>0</no\_child\_flag>

<bassinet\_flag>0</bassinet\_flag>

<no\_infant\_flag>0</no\_infant\_flag>

<infant\_flag>0</infant\_flag>

<emergency\_exit\_flag>0</emergency\_exit\_flag>



```

<unaccompanied_minors_flag>0</unaccompanied_minors_flag>

<window_flag>0</window_flag>

<aisle_flag>0</aisle_flag>

<block_b2c_flag>0</block_b2c_flag>

<block_b2b_flag>0</block_b2b_flag>

<blocked_flag>0</blocked_flag>

<low_comfort_flag>0</low_comfort_flag>

<passenger_count>0</passenger_count>

<passenger_id>e3293c3c-cc2c-4900-beb7-3f0e6b94ce33</passenger_id>

<firstname>ATHINYA</firstname>

<lastname>CHUAJEDTONE</lastname>

<title_rcd>MISS</title_rcd>

<gender_type_rcd>F</gender_type_rcd>

<fee_rcd>SEAT</ fee_rcd >

</Seat>

</SeatMaps>

```



Name	ClientProfile/Client
Description	Show a list of client profile member.
Field name and description	
<u>Name</u>	<u>Description</u>
flight_id	Flight Id
free_seating_flag	Flag to identify if the flight is the free seating.  <b>0</b> = Not free seating.  <b>1</b> = Free seating.
origin_rcd	Origin Airport
destination_rcd	Destination Airport
aircraft_configuration_code	Aircraft configuration code.
number_of_bays	Total number of Bay
boarding_class_rcd	Boarding class of this seat.
number_of_rows	Total number of layout row.
number_of_columns	Total number of layout column.
layout_row	Row position of the seatmap layout.
layout_column	Column position of the seatmap layout.
seat_row	Seat Row number.
seat_column	Seat Column alphabet.
stretcher_flag	Tell that this seat is a stretcher seat.  <b>0</b> = Not stretcher.  <b>1</b> = stretcher.
handicapped_flag	Tell that this seat is for handicapped.  <b>0</b> = Not for handicapped.  <b>1</b> = For handicapped.
no_child_flag	



<b>bassinet_flag</b>	<p>Identify that this seat has bassinet.</p> <p><b>0</b> = Not include.</p> <p><b>1</b> = Include.</p>
<b>no_infant_flag</b>	
<b>infant_flag</b>	
<b>emergency_exit_flag</b>	<p>Tell that this seat is close to the emergency exit.</p> <p><b>0</b> = Not close to emergency exit</p> <p><b>1</b> = Close to emergency exit.</p>
<b>unaccompanied_minors_flag</b>	
<b>window_flag</b>	<p>Tell that this seat is close to window.</p> <p><b>0</b> = Not close to window</p> <p><b>1</b> = Close to window.</p>
<b>aisle_flag</b>	<p>Tell that this seat is close to aisle.</p> <p><b>0</b> = Not close to aisle</p> <p><b>1</b> = Close to aisle.</p>
<b>block_b2c_flag</b>	<p>Tell that this seat is block to B2C booking.</p> <p><b>0</b> = Can select.</p> <p><b>1</b> = Block.</p>
<b>block_b2b_flag</b>	<p>Tell that this seat is block to B2B booking.</p> <p><b>0</b> = Can select.</p> <p><b>1</b> = Block.</p>
<b>blocked_flag</b>	<p>Tell that this seat is block.</p> <p><b>0</b> = Can select.</p>



	<b>1=</b> Block.
low_comfort_flag	
passenger_count	Number of passenger that occupied the seat. Maximun is 2 ADULT + INFANT.
passenger_id	Id of the passenger that occupied seat.
firstname	Firstname of the passenger that occupied seat.
lastname	Lastname of the passenger that occupied seat.
title_rcd	Title of the passenger that occupied seat.
gender_type_rcd	Gender type code of the passenger that occupied seat.
fee_rcd	Fee Code. You need to pass this code to function SelectSeat() if this field present in the seatmap xml.



## 16. APPENDIX 12: GETFEEDEFINITION XML

```

<Fees>

<Fee>

  <fee_id>ba4be872-f8c5-4248-8039-3f2210aeae5d</fee_id>

  <fee_rcd>INSU</fee_rcd>

  <currency_rcd>JPY</currency_rcd>

  <display_name>ACE Insurance</display_name>

  <fee_amount>800.00</fee_amount>

  <fee_amount_incl>840.00</fee_amount_incl>

  <fee_percentage>0</fee_percentage>

  <vat_percentage>5.00</vat_percentage>

  <minimum_fee_amount_flag>0</minimum_fee_amount_flag>

  <od_flag>0</od_flag>

  <fee_category_rcd>INSURANCE</fee_category_rcd>

  <charge_amount>0</charge_amount>

  <charge_amount_incl>0</charge_amount_incl>

</Fee>

</Fees>

```

Name	ClientProfile/Client
Description	Show a list of fees.
Field name and description	
Name	Description
fee_id	GUID for fee.
fee_rcd	Fee Code.





currency_rcd	Currency Code.
display_name	Fee display name.
fee_amount	Fee Amount
fee_amount_incl	Fee Amount + VAT
fee_percentage	Fee percentage. This file is only use with payment fee.
vat_percentage	Vat percentage calculate form Fee amount.
minimum_fee_amount_flag	Minimum fee charge if fee amount is less than fee percentage. If this flag is set to on(1), and fee_amount is less than fee percentage then take fee percentage as an amount.
od_flag	Incase of Married flight, if marked O&D, fee will be applied to first origin and final destination.
fee_category_rcd	Category fee code.
charge_amount	Charge amount in other currency that is not sale currency.
charge_amount_incl	Charge_amount + VAT.



## 17. Appendix 13: VOUCHER TEMPLATE XML

<Template>

<Voucher>

<voucher\_template\_id>8c109add-f0b3-4e9f-9218-aeff2edfc297</voucher\_template\_id>

<form\_of\_payment\_subtype\_rcd>VOUCHER</form\_of\_payment\_subtype\_rcd>

<form\_of\_payment\_rcd>VOUCHER</form\_of\_payment\_rcd>

<currency\_rcd>JPY</currency\_rcd>

<voucher\_value>100000</voucher\_value>

<charge\_amount>100</charge\_amount>

<multiple\_payment\_flag>1</multiple\_payment\_flag>

<valid\_to\_date>2014-02-28T00:00:00+07:00</valid\_to\_date>

<valid\_days>365</valid\_days>

<valid\_from\_date>2012-04-25T00:00:00+07:00</valid\_from\_date>

<display\_name>Free voucher All</display\_name>

<status\_code>A</status\_code>

<create\_by>871ae70b-f731-488e-9429-6260677c622c</create\_by>

<create\_date\_time>2012-04-26T13:12:11+07:00</create\_date\_time>

<update\_by>ef3e720e-a518-11df-a0ea-00014a18390d</update\_by>

<update\_date\_time>2013-11-13T18:43:58+07:00</update\_date\_time>

<origins>KIX</origins>

<destinations>FUK</destinations>

<passenger\_segments>2</passenger\_segments>

<recipient\_only\_flag>1</recipient\_only\_flag>

<fare\_only\_flag>1</fare\_only\_flag>



```

<valid_for_class>FCY</valid_for_class>

<ticket_flag>1</ticket_flag>

<seat_fee_flag>1</seat_fee_flag>

<other_fee_flag>1</other_fee_flag>

<create_name>panjaroen-soiadmin supannee</create_name>

<update_name>Charoenthongtrakul Ornuma</update_name>

</Voucher>

</Template>

```



Name	Template/Voucher
Description	Show a list of voucher template.
Field name and description	
<u>Name</u>	<u>Description</u>
voucher_template_id	ID use to identify individual template.
form_of_payment_subtype_rcd	Form of payment subtype code.
form_of_payment_rcd	Form of payment code. Usually set as "VOUCHER"
currency_rcd	Currency code.
voucher_value	Value of voucher
charge_amount	Voucher charge amount. Use a payment amount.
multiple_payment_flag	Flag identify that this voucher can use multiple time.  0 = One time use.  1 = Can use multiple time.
valid_to_date	Voucher valid until.
valid_days	Number of day that the voucher will valid when first issue.
valid_from_date	Voucher valid from date.
display_name	Template display name.
status_code	Voucher status.  A = Active  I = Inactive  D = Delete.
create_by	ID of use that create template.
create_date_time	Template creates date time.
update_by	ID of user update template information.
update_date_time	Template updates date time.
origins	The airport origin that this template can use.
destinations	The airport destination that this template can use.
passenger_segments	The maximum number of passenger that can be use by this



	voucher.
recipient_only_flag	<p>Flag use to identify that field recipient_name have to match with passenger name.</p> <p>1 = have to match.</p> <p>0 = Can be any name.</p>
fare_only_flag	<p>Flag use to identify that this voucher can use for fare payment only.</p> <p>1 = Can use for fare payment only.</p> <p>0 = Can use with any charge item.</p>
valid_for_class	Identify the booking class that this voucher allowed to use.
ticket_flag	This flag is use to identify that this voucher can use for ticket payment.
seat_fee_flag	This flag is use to identify that this voucher can use for seat fee payment.
other_fee_flag	This flag is use to identify that this voucher can use for other fee payment.
create_name	Display name of user create voucher template.
update_name	Display name of user update voucher template information.



---

## 18. Appendix 14: VOUCHER XML

<Voucher>

<Details>

<voucher\_id>39e640ef-bd35-4d1d-b024-932d9bb8cfb8</voucher\_id>

<voucher\_number>3700002498</voucher\_number>

<agency\_code>NPAC</agency\_code>

<form\_of\_payment\_subtype\_rcd>VOUCHER</form\_of\_payment\_subtype\_rcd>

<form\_of\_payment\_rcd>VOUCHER</form\_of\_payment\_rcd>

<currency\_rcd>JPY</currency\_rcd>

<create\_by>06981bf3-c4e0-4337-b761-012d9bfa68d8</create\_by>

<create\_date\_time>2013-11-14T09:14:56.713+07:00</create\_date\_time>

<update\_by>06981bf3-c4e0-4337-b761-012d9bfa68d8</update\_by>

<update\_date\_time>2013-11-14T09:14:56.713+07:00</update\_date\_time>

<void\_by>06981bf3-c4e0-4337-b761-012d9bfa68d8</void\_by>

<void\_date\_time>2013-11-14T09:15:23.57+07:00</void\_date\_time>

<expiry\_date\_time>2014-11-14T16:14:52+07:00</expiry\_date\_time>

<fare\_only\_flag>1</fare\_only\_flag>

<ticket\_flag>1</ticket\_flag>

<seat\_fee\_flag>1</seat\_fee\_flag>

<other\_fee\_flag>1</other\_fee\_flag>

<recipient\_only\_flag>1</recipient\_only\_flag>

<multiple\_payment\_flag>1</multiple\_payment\_flag>

<valid\_from\_date>2012-04-25T00:00:00+07:00</valid\_from\_date>

<valid\_to\_date>2014-02-28T00:00:00+07:00</valid\_to\_date>



```

<valid_for_class>FCY</valid_for_class>

<origins>KIX</origins>

<destinations>FUK</destinations>

<voucher_value>100000</voucher_value>

<passenger_segments>2</passenger_segments>

<recipient_name>Name Lastname</recipient_name>

<voucher_status_rcd>CANCELLED</voucher_status_rcd>

<charge_amount>100</charge_amount>

<voucher_password>7479</voucher_password>

<acct_charge_amount>82</acct_charge_amount>

<acct_voucher_value>82000</acct_voucher_value>

<voucher_template_id>8c109add-f0b3-4e9f-9218-aeff2edfc297</voucher_template_id>

<booking_payment_id>3bba9c87-b957-409f-9fbb-76737870a1a6</booking_payment_id>

<create_name>Name Lastname</create_name>

<update_name>Name Lastname</update_name>

<void_name>Name Lastname</void_name>

</Details>

</Voucher>

```



Name	Voucher/Details
Description	Show a list of voucher template.
Field name and description	
<u>Name</u>	<u>Description</u>
voucher_id	ID GUID use to identify individual Voucher.
form_of_payment_subtype_rcd	Form of payment subtype code.
voucher_number	Voucher number.
agency_code	Agency code that create voucher.
voucher_password	Voucher password.
form_of_payment_rcd	Form of payment code. Usually set as "VOUCHER"
currency_rcd	Currency code.
voucher_value	Value of voucher
charge_amount	Voucher charge amount. Use a payment amount.
multiple_payment_flag	Flag identify that this voucher can use multiple time.  0 = One time use.  1 = Can use multiple time.
valid_to_date	Voucher valid until.
expiry_date_time	Expiry date. This field have to be calculate from valid_days field of Voucher template XML.
valid_from_date	Voucher valid from date.
display_name	Template display name.
status_code	Voucher status.  A = Active  I = Inactive  D = Delete.
create_by	ID of use that create template.
create_date_time	Template creates date time.
update_by	ID of user update template information.





update_date_time	Template updates date time.
void_by	ID of user that void voucher.
void_date_time	Void voucher date time.
origins	The airport origin that this template can use.
destinations	The airport destination that this template can use.
passenger_segments	The maximum number of passenger that can be use by this voucher.
recipient_only_flag	Flag use to identify that field recipient_name have to match with passenger name.  1 = have to match.  0 = Can be any name.
fare_only_flag	Flag use to identify that this voucher can use for fare payment only.  1 = Can use for fare payment only.  0 = Can use with any charge item.
valid_for_class	Identify the booking class that this voucher allowed to use.
ticket_flag	This flag is use to identify that this voucher can use for ticket payment.
seat_fee_flag	This flag is use to identify that this voucher can use for seat fee payment.
other_fee_flag	This flag is use to identify that this voucher can use for other fee payment.
create_name	Display name of user create voucher template.
update_name	Display name of user update voucher template information.
recipient_name	The value of this field will be the name of passenger. If recipient_only_flag is "1" passenger name have to match with this field.
voucher_status_rcd	Staus code of voucher.



	<p>CANCELLED – Cancel voucher</p> <p>OPEN – Active voucher.</p> <p>EXPIRED – Expire voucher.</p> <p>USED – Identify that this voucher is already use.</p>
acct_charge_amount	Charge in accounting currency.
acct_voucher_value	Voucher value in accounting currency.
voucher_template_id	Reference Id to voucher template.



## 19. Appendix 15: PASSENGERSMANIFEST XML

<PassengersManifest>

<PassengerManifest>

<booking\_id>f14e0f44-38cb-4d0a-aa95-ae31a2aa9afc</booking\_id>

<record\_locator>LVN4GZ</record\_locator>

<language\_rcd>EN</language\_rcd>

<agency\_code>KAGENCY</agency\_code>

<contact\_name>Mr. Krueger</contact\_name>

<contact\_email>kitiya@mercator.asia</contact\_email>

<mobile\_email>kitiya@mercator.asia</mobile\_email>

<phone\_business>00</phone\_business>

<received\_from>c</received\_from>

<group\_booking\_flag>0</group\_booking\_flag>

<passenger\_id>86dea22e-a5a1-4c1e-a2ac-bd9188650029</passenger\_id>

<booking\_segment\_id>d8db6bf5-9c8e-4dc1-906f-cba8376b0cab</booking\_segment\_id>

<passenger\_type\_rcd>ADULT</passenger\_type\_rcd>

<lastname>YAMAMURA</lastname>

<firstname>SADAKO</firstname>

<date\_of\_birth>1988-11-14T00:00:00+07:00</date\_of\_birth>

<title\_rcd>MRS</title\_rcd>

<gender\_type\_rcd>F</gender\_type\_rcd>

<client\_number>60004900</client\_number>

<member\_number>6666</member\_number>

<member\_level\_rcd>PLT</member\_level\_rcd>



```

<create_date_time>2014-01-30T09:40:23.663+07:00</create_date_time>

<create_by>d4e0103d-284e-4c2c-abaa-74c0809a098f</create_by>

<update_date_time>2014-01-30T16:44:37+07:00</update_date_time>

<update_by>d4e0103d-284e-4c2c-abaa-74c0809a098f</update_by>

<airline_rcd>EK</airline_rcd>

<flight_number>6662</flight_number>

<departure_date>2014-08-01T00:00:00+07:00</departure_date>

<departure_time>2000</departure_time>

<origin_rcd>GPS</origin_rcd>

<destination_rcd>ATL</destination_rcd>

<boarding_class_rcd>C</boarding_class_rcd>

<booking_class_rcd>J</booking_class_rcd>

<inventory_class_rcd>J</inventory_class_rcd>

<segment_status_rcd>MM</segment_status_rcd>

<fare_id>375dabe6-a396-4b1f-89bc-17be92714193</fare_id>

<net_total>103.5</net_total>

<fare_amount>103.5</fare_amount>

<fare_date_time>2014-01-30T16:40:00+07:00</fare_date_time>

<fare_type_rcd>FARE</fare_type_rcd>

<currency_rcd>USD</currency_rcd>

<ticket_number>9242101338818</ticket_number>

<e_ticket_flag>1</e_ticket_flag>

<seat_number>1E</seat_number>

<baggage_weight>0</baggage_weight>

```



<piece\_allowance>0</piece\_allowance>

<check\_in\_baggage\_weight>0</check\_in\_baggage\_weight>

<hand\_baggage\_weight>0</hand\_baggage\_weight>

</PassengerManifest>

</PassengersManifest>



Name	PassengersManifest/PassengerManifest
Description	Show a list of passenger manifest.
Field name and description	
<u>Name</u>	<u>Description</u>
booking_id	ID GUID use to identify individual booking.
record_locator	Booking PNR number.
language_rcd	Language two letter code.
agency_code	Agency code that create voucher.
contact_name	Contact name of booking.
contact_email	Contact email of booking.
mobile_email	Contact mobile email of booking.
phone_business	Contact business phone of booking.
received_from	Booking receive from.
group_booking_flag	Identify if the booking is a group booking.  0 = Not Group Booking.  1 = Group Booking.
passenger_id	ID GUID use to identify individual passenger.
booking_segment_id	ID GUID use to identify individual flight segment.
passenger_type_rcd	Passenger type code.  ADULT = Adult  CHD = Child  INF = Infant
lastname	Passenger last name.
firstname	Passenger first name.
date_of_birth	Passenger date of birth.
title_rcd	Passenger title code. Base on Avantik setting.
gender_type_rcd	Passenger gender type code. Base on Avantik setting.



client_number	Client profile number of passenger in the booking.
member_number	Member number of the client profile in the booking.
member_level_rcd	FFP member level. Base on Avantik setting.
airline_rcd	Two letter code of airline.
flight_number	Flight number.
departure_date	Flight departure date.
departure_time	Flight departure time.
origin_rcd	Three letter code of airport origin.
destination_rcd	Three letter code of airport destination.
boarding_class_rcd	Boarding class code. Base on Avantik setting.
booking_class_rcd	Booking class code. Base on Avantik setting.
segment_status_rcd	Flight segment status. Base on Avantik Setting.
fare_id	ID GUID use to identify individual booking.
net_total	Total charge which include Fare, VAT, and tax.
fare_amount	Ticket fare amount.
fare_type_rcd	Type of fare as FARE or POINT.  FARE = Normal ticket fare amount.  POINT = FFP point redumption.
currency_rcd	Currency three letter code.
ticket_number	Ticket number.
e_ticket_flag	Identify if the booking is an e-ticket.  0 = Not e-ticket.  1 = E-ticket.
seat_number	Passenger seat number.
baggage_weight	Baggage weight.
piece_allowance	Baggage piece allowance.
check_in_baggage_weight	Check in baggage weight.
hand_baggage_weight	Hand baggage weight.



---

## 20. Appendix 16: Activities

<Activities>

<Activity>

<booking\_id>ab1ba032-2932-4a4c-a8cb-85a44d767a0d</booking\_id>

<record\_locator>LVM9K2</record\_locator>

<booking\_number>667558465</booking\_number>

<language\_rcd>EN</language\_rcd>

<booking\_agency\_code>NPAC</booking\_agency\_code>

<contact\_name>ashish ashish</contact\_name>

<phone\_mobile>9876543210</phone\_mobile>

<booking\_source\_rcd>B2B</booking\_source\_rcd>

<lastname>ashish</lastname>

<firstname>ashish</firstname>

<address\_line1>Dubai</address\_line1>

<city>Dubai</city>

<title\_rcd>MR.</title\_rcd>

<country\_rcd>GB</country\_rcd>

<agency\_code>NPAC</agency\_code>

<booking\_remark\_id>4e500d1d-9b3e-48e2-b223-5651ad42f65b</booking\_remark\_id>

<remark\_type\_rcd>TKTL</remark\_type\_rcd>

<timelimit\_date\_time>2012-08-27T17:52:06+07:00</timelimit\_date\_time>

<remark\_text>LOCAL TIME AT POS NPAC</remark\_text>

<added\_by>SYS</added\_by>

<complete\_flag>0</complete\_flag>





<remark\_name>Ticket Timelimit</remark\_name>

<passenger\_name>ASHISH/ASHISH MR.</passenger\_name>

<flight\_segment>T1 152 27 Sep 12 FUK - KIX</flight\_segment>

</Activity>

</Activities>

Name	Activities/Activity
Description	Show a list of activity.
Field name and description	
<u>Name</u>	<u>Description</u>
booking_id	ID GUID use to identify individual booking.
record_locator	Booking PNR number.
booking_number	Booking number.
booking_agency_code	The agency code that create booking.
contact_name	Booking contact person name.
phone_mobile	Contact mobile phone number.
Lastname	Passenger last name.
Firstname	Passenger first name.
title_rcd	Passenger title.
address_line1	Contact address.
City	Contact city.
country_rcd	Contact country code.
agency_code	Code of agency that create a remark.
booking_remark_id	Id use for identify individual remark.
timelimit_date_time	Payment timelimit for TKTL remark type.
remark_text	Remark description.
added_by	User name who add the remark.
complete_flag	Identify if the activity is complete.
remark_name	Remark display name.
passenger_name	Name of passenger. The passenger name will be in below



	format.  [Lastname]/[Firstname] [TitleRcd]
flight_segment	Flight segment information. Flight information will show as below format.  [Airline Code] [Flight number] [Dept. Day] [Dept.Month] [Dept.Year] [origin code] [destination code]



## 21. Appendix 17: RemarkQueue

<RemarksQueue>

<RemarkQueue>

<sort\_sequence>0</sort\_sequence>

<is\_event>0</is\_event>

<within\_48>0</within\_48>

<remark\_type\_rcd>B2BREQ</remark\_type\_rcd>

<display\_name>B2B Agency Request</display\_name>

<queue\_count>1</queue\_count>

</RemarkQueue>

</RemarksQueue>

Name	RemarksQueue/RemarkQueue
Description	Count the pending remark by category and agency.
Field name and description	
<u>Name</u>	<u>Description</u>
sort_sequence	Sort sequence of the queue.
is_event	Identify whether it is fall into below category.  0 = Remark & Activity.  1 = System event.
within_48	Identify whenter the remark is on queue within 48 hrs compare to departure date.  0 = More than 48 hrs.  1 = Within 48 hrs.



remark_type_rcd	Remark type code
display_name	Remark display name.
queue_count	Number of pending queue.



## 22. Appendix 18: GetBookings

<Bookings>

<Booking>

<booking\_id>83a85dbb-2489-4eff-86c3-7a9e3d437f6d</booking\_id>

<record\_locator>LVN9JE</record\_locator>

<create\_date\_time>2014-05-01T09:23:11.63+07:00</create\_date\_time>

<update\_date\_time>2014-05-01T09:23:11.63+07:00</update\_date\_time>

<create\_by>Webagent B2C</create\_by>

<update\_by>Webagent B2C</update\_by>

<language\_rcd>EN</language\_rcd>

<currency\_rcd>JPY</currency\_rcd>

<agency\_code>B2C</agency\_code>

<contact\_name>One Test</contact\_name>

<contact\_email>ang@test.com</contact\_email>

<phone\_mobile>02</phone\_mobile>

<phone\_home>02</phone\_home>

<phone\_business>02</phone\_business>

<received\_from>WEB</received\_from>

<phone\_search>02</phone\_search>

<group\_booking\_flag>0</group\_booking\_flag>

<booking\_source\_rcd>B2C</booking\_source\_rcd>

<passenger\_name>TEST/ONE MR</passenger\_name>

<flight\_segment>MM 158 07 May 14 KIX - FUK</flight\_segment>



```

<flight_segment_xx>MM 158 07 May 14 KIX - FUK</flight_segment_xx>

<passenger_count>1</passenger_count>

<infant_count>0</infant_count>

<ticket_total>1586</ticket_total>

<fee_total>550</fee_total>

<ticket_payment_total>1586</ticket_payment_total>

<fee_payment_total>550</fee_payment_total>

</Booking>

</Bookings>

```

Name	Bookings/Booking
Description	Show a list of activity.
Field name and description	
<u>Name</u>	<u>Description</u>
booking_id	ID GUID use to identify individual booking.
record_locator	Booking PNR number.
create_date_time	Date time that the booking is created.
update_date_time	Date time that the booking is update.
create_by	User that create the booking.
update_by	User that update the booking.
language_rcd	Language code that make the booking.
currency_rcd	Booking currency code.
agency_code	Agency that create the booking.
contact_name	Booking contact person name.
contact_email	Booking contact email.
phone_mobile	Booking contact mobile email.
phone_home	Booking contact home phone.
phone_business	Booking contact business phone.
received_from	Booking free text



phone_search	Phone number exclude alphanumeric.
group_booking_flag	Identify whether the booking is a group or individual booking.  0 = Individual.  1 = Group booking.
booking_source_rcd	Show the booking is create from one of below source:  INT = booking from Avantik  B2C = booking from B2C  B2B = booking from B2B, B2E
passenger_name	First passenger name in the booking.
flight_segment	Show flight segment information  It there is departure date then show airline code, flight number, departure date, origin, destination.  If there is no departure date then show 'OPEN' + origin, destination.  For segment status are 'HK', 'MM', 'HL', 'QQ', 'PO'
flight_segment_xx	Show flight segment information  It there is departure date then show airline code, flight number, departure date, origin, destination.  If there is no departure date then show 'OPEN' + origin, destination. For all segment status.
passenger_count	Total passenger count in the booking.
infant_count	Total infant count in the booking.
ticket_total	Total ticket amount.
fee_total	Show total of booking fee amount?
ticket_payment_total	Show total payment that paid for total ticket amount.



fee_payment_total	Show fee payment that paid for total booking fee amount.
-------------------	----------------------------------------------------------





---

## 23. Appendix 19: Active Booking XML

```
<Bookings>

<ActiveBooking>

  <booking_id>f5d55985-4076-40f4-b635-c626ee2c3dae</booking_id>

  <client_profile_id>62dc9c26-efb7-4955-8b2b-6a0944da49b7</client_profile_id>

  <record_locator>LVN9MR</record_locator>

  <create_date_time>2014-05-06T12:07:35.64+07:00</create_date_time>

  <update_date_time>2014-05-06T12:07:35.64+07:00</update_date_time>

  <airline_rcd>AB</airline_rcd>

  <flight_number>220</flight_number>

  <departure_date>2014-05-07T00:00:00+07:00</departure_date>

  <departure_time>1500</departure_time>

  <arrival_time>1600</arrival_time>

  <origin_rcd>XCN</origin_rcd>

  <destination_rcd>LWY</destination_rcd>

  <booking_class_rcd>C</booking_class_rcd>

  <boarding_class_rcd>C</boarding_class_rcd>

  <segment_status_rcd>HK</segment_status_rcd>

  <number_of_units>1</number_of_units>

  <flight_segment>AB 220 07 May 14 XCN - LWY</flight_segment>

  <passenger_name>PANBOY/BOYB MISS</passenger_name>

</ActiveBooking>

</Bookings>
```



Name	Bookings/ActiveBooking
Description	Show a list of active booking.
Field name and description	
<u>Name</u>	<u>Description</u>
booking_id	Unique booking ID.
client_profile_id	Unique ID of client profile.
record_locator	Booking reference.
create_date_time	Date time that the booking is created.
update_date_time	Date time that the booking is update.
airline_rcd	Airline IATA code.
flight_number	Flight number.
departure_date	Departure date.
departure_time	Departure time.
arrival_time	Arrival time.
origin_rcd	Airport origin code.
destination_rcd	Airport destination code.
booking_class_rcd	Flight booking class.
boarding_class_rcd	Flight boarding class.
segment_status_rcd	Flight segment status:  HK = Confirmed  HL = Waitlisted  XX = Cancelled
number_of_units	Number of passengers on each segment booked.
flight_segment	Show flight information.
passenger_name	Show the name of the first passenger.

<Bookings>

<FlownBooking>

<booking\_id>f5d55985-4076-40f4-b635-c626ee2c3dae</booking\_id>



```

<client_profile_id>62dc9c26-efb7-4955-8b2b-6a0944da49b7</client_profile_id>

<record_locator>LVN9MR</record_locator>

<create_date_time>2014-05-06T12:07:35.64+07:00</create_date_time>

<update_date_time>2014-05-06T12:07:35.64+07:00</update_date_time>

<airline_rcd>AB</airline_rcd>

<flight_number>220</flight_number>

<departure_date>2014-05-07T00:00:00+07:00</departure_date>

<departure_time>1500</departure_time>

<arrival_time>1600</arrival_time>

<origin_rcd>XCN</origin_rcd>

<destination_rcd>LWY</destination_rcd>

<booking_class_rcd>C</booking_class_rcd>

<boarding_class_rcd>C</boarding_class_rcd>

<segment_status_rcd>HK</segment_status_rcd>

<number_of_units>1</number_of_units>

<flight_segment>AB 220 07 May 14 XCN - LWY</flight_segment>

<passenger_name>PANBOY/BOYB MISS</passenger_name>

</FlownBooking>

</Bookings>

```

Name	Bookings/FlownBooking
Description	Show a list of active booking.
Field name and description	
<u>Name</u>	<u>Description</u>
booking_id	Unique booking ID.



client_profile_id	Unique ID of client profile.
record_locator	Booking reference.
create_date_time	Date time that the booking is created.
update_date_time	Date time that the booking is update.
airline_rcd	Airline IATA code.
flight_number	Flight number.
departure_date	Departure date.
departure_time	Departure time.
arrival_time	Arrival time.
origin_rcd	Airport origin code.
destination_rcd	Airport destination code.
booking_class_rcd	Flight booking class.
boarding_class_rcd	Flight boarding class.
segment_status_rcd	Flight segment status:  HK = Confirmed  HL = Waitlisted  XX = Cancelled
number_of_units	Number of passengers on each segment booked.
flight_segment	Show flight information.
passenger_name	Show the name of the first passenger.

## 24. Appendix 20: GetPointHistory

<Transaction>

<ffp\_transaction\_id>3d4db055-9ba5-46b7-9b11-d5e40640bdc5</ffp\_transaction\_id>

<client\_profile\_id>f2c8e43b-b4a8-4f92-88c5-01bce0202d30</client\_profile\_id>

<passenger\_profile\_id>2e907d41-0535-4df9-b19e-1fbd12d81515</passenger\_profile\_id>

<booking\_id>da71d837-5fa7-49c0-8191-f0c43086fa66</booking\_id>



```

<passenger_id>53bbc3fc-f336-4a69-baa5-e1ef1ead6021</passenger_id>

<booking_segment_id>bd9819d3-1bf4-428a-abe7-759ded47267e</booking_segment_id>

<segment_type_rcd>AIR</segment_type_rcd>

<airline_rcd>WK</airline_rcd>

<flight_number>606</flight_number>

<origin_rcd>GCI</origin_rcd>

<destination_rcd>LGW</destination_rcd>

<departure_date>2016-12-25T00:00:00+07:00</departure_date>

<boarding_class_rcd>Y</boarding_class_rcd>

<booking_class_rcd>S</booking_class_rcd>

<ffp_amount>-1</ffp_amount>

<ffp_total>-1</ffp_total>

<record_locator>A051X4</record_locator>

<create_date_time>2016-12-15T09:48:47.4+07:00</create_date_time>

<expired_flag>0</expired_flag>

<passenger_lastname>PROMRUKSA</passenger_lastname>

<passenger_firstname>RATCHANEE</passenger_firstname>

</Transaction>

```



## 25. Appendix 21: Flight Redemption Availability response

AvailabilityFlight>

```

<airline_rcd>GR</airline_rcd>

<flight_number>604</flight_number>

<booking_class_rcd>Y</booking_class_rcd>

<boarding_class_rcd>Y</boarding_class_rcd>

<flight_id>{031CA90F-35A3-46F8-8AF0-CB675EE184D8}</flight_id>

<origin_rcd>GCI</origin_rcd>

<destination_rcd>LGW</destination_rcd>

<origin_name>Guernsey</origin_name>

<destination_name>London Gatwick</destination_name>

<flight_status_rcd>ACTIVE</flight_status_rcd>

<departure_date>2017-02-25T00:00:00</departure_date>

<planned_departure_time>1020</planned_departure_time>

<planned_arrival_time>1110</planned_arrival_time>

<fare_id>{A96933ED-8135-4D4A-BF0D-8BCC53615187}</fare_id>

<fare_code>SWEB</fare_code>

```



```

<fare_type_rcd>POINT</fare_type_rcd>

<redemption_points>2</redemption_points>

<transit_points />

<transit_points_name />

<transit_flight_id />

<transit_booking_class_rcd />

<transit_boarding_class_rcd />

<transit_airport_rcd />

<transit_departure_date />

<transit_planned_departure_time />

<transit_planned_arrival_time />

<transit_fare_id />

<transit_redemption_points />

<transit_name />

<transit_waitlist_open_flag />

<transit_airline_rcd />

<transit_flight_number />

<transit_flight_status_rcd />

<transit_flight_duration />

<transit_class_open_flag />

```



```

<transit_nesting_string />

<nesting_string>YBMUHQVWSTELK</nesting_string>

<nested_book_available>85</nested_book_available>

<full_flight_flag>0</full_flight_flag>

<class_open_flag>1</class_open_flag>

<close_web_sales>0</close_web_sales>

<total_adult_fare>14.59</total_adult_fare>

<total_child_fare>0.0000</total_child_fare>

<total_infant_fare>0.0000</total_infant_fare>

<other_fare>0.0000</other_fare>

<fare_column>1</fare_column>

<flight_comment />

<filter_logic_flag>1</filter_logic_flag>

<restriction_text />

<flight_duration>50</flight_duration>

<class_capacity>85</class_capacity>

<waitlist_open_flag>0</waitlist_open_flag>

<refundable_flag>0</refundable_flag>

<currency_rcd />

<aircraft_type_rcd>AT7</aircraft_type_rcd>

```





```

    <transit_aircraft_type_rcd />

    <arrival_date>2017-02-25T00:00:00</arrival_date>

    <transit_arrival_date />

    <number_of_stops>0</number_of_stops>

    <eticket_flag />

    <nest_seat_available />

    <endorsement_text />

  </AvailabilityFlight>

```

---

## 26. APPENDIX 22: PASSENGER MANIFEST

```

<root>

  <Flight>

    <flight_id>2cb01e3e-28be-4885-abe4-8e63efafc475</flight_id>

    <airline_rcd>3L</airline_rcd>

    <flight_number>333</flight_number>

    <origin_rcd>FDH</origin_rcd>

    <destination_rcd>HAM</destination_rcd>

    <departure_date_from>2015-02-27T00:00:00</departure_date_from>

    <departure_date_to>2015-02-27T00:00:00</departure_date_to>

    <flight_legs_count>1</flight_legs_count>

    <flight_leg>

      <flight_id>2cb01e3e-28be-4885-abe4-8e63efafc475</flight_id>

      <arrival_airport_rcd>HAM</arrival_airport_rcd>
    
```



```

<arrival_date>2015-02-27T00:00:00</arrival_date>

<utc_arrival_date_time>2015-02-27T12:00:00+07:00</utc_arrival_date_time>

<planned_arrival_time>0913</planned_arrival_time>

<departure_airport_rcd>FDH</departure_airport_rcd>

<departure_date>2015-02-27T00:00:00</departure_date>

<planned_departure_time>0825</planned_departure_time>

<utc_departure_date_time>2015-02-27T12:00:00+07:00</utc_departure_date_time>

</flight_leg>

<number_of_leg>

  <Adult>

    <Boarded>0</Boarded>

    <Checked>0</Checked>

    <Flown>0</Flown>

    <Noshow>1</Noshow>

    <No_check-in_status>0</No_check-in_status>

    <Offloaded>1</Offloaded>

  </Adult>

  <Child>

    <Boarded>0</Boarded>

    <Checked>0</Checked>

    <Flown>0</Flown>

    <Noshow>0</Noshow>

    <No_check-in_status>0</No_check-in_status>

    <Offloaded>0</Offloaded>

  </Child>

```



```

<Infant>

  <Boarded>0</Boarded>

  <Checked>0</Checked>

  <Flown>0</Flown>

  <Noshow>0</Noshow>

  <No_check-in_status>0</No_check-in_status>

  <Offloaded>1</Offloaded>

</Infant>

<Other>

  <Boarded>0</Boarded>

  <Checked>0</Checked>

  <Flown>0</Flown>

  <Noshow>0</Noshow>

  <No_check-in_status>0</No_check-in_status>

  <Offloaded>0</Offloaded>

</Other>

</number_of_leg>

<boarding_class>

  <Adult>

    <First_class>0</First_class>

    <Business_class>0</Business_class>

    <Economy_class>2</Economy_class>

  </Adult>

  <Child>

    <First_class>0</First_class>

```



```

    <Business_class>0</Business_class>

    <Economy_class>0</Economy_class>

</Child>

<Infant>

    <First_class>0</First_class>

    <Business_class>0</Business_class>

    <Economy_class>1</Economy_class>

</Infant>

<Other>

    <First_class>0</First_class>

    <Business_class>0</Business_class>

    <Economy_class>0</Economy_class>

</Other>

</boarding_class>

<Passenger>

    <flight_id>2cb01e3e-28be-4885-abe4-8e63efafc475</flight_id>

    <booking_id>27c6a9b6-0468-450c-8e08-ab8629a40864</booking_id>

    <language_rcd>EN</language_rcd>

    <contact_name>firstname lastname</contact_name>

    <contact_email>rachthawis.sakhorn@bravo.aero</contact_email>

    <phone_mobile>3452345234</phone_mobile>

    <phone_home>3245234523</phone_home>

    <passenger_id>79b07126-bff3-4917-b21d-2c0fd1a8b749</passenger_id>

    <booking_segment_id>8f5372cb-5bea-4fe6-a09d-860ad1a6f669</booking_segment_id>

    <booking_class_rcd>K</booking_class_rcd>

```



```

<boarding_class_rcd>Y</boarding_class_rcd>

<origin_rcd>FDH</origin_rcd>

<destination_rcd>HAM</destination_rcd>

<boarding_pass_number>1260870234</boarding_pass_number>

<baggage_weight>0</baggage_weight>

<piece_allowance>0</piece_allowance>

<passenger_check_in_status_rcd>OFFLOADED</passenger_check_in_status_rcd>

<passenger_status_rcd>OK</passenger_status_rcd>

<standby_flag>0</standby_flag>

<e_ticket_flag>1</e_ticket_flag>

<fare_code>K</fare_code>

<ticket_number>9242101358692</ticket_number>

<ticketing_date_time>2015-02-26T10:10:40+07:00</ticketing_date_time>

<fare_date_time>2015-02-26T10:04:05+07:00</fare_date_time>

<currency_rcd>EUR</currency_rcd>

<fare_type_rcd>FARE</fare_type_rcd>

<net_total>161</net_total>

<fare_amount>135</fare_amount>

<tax_amount>0</tax_amount>

<yq_amount>0</yq_amount>

<ticketing_fee_amount>0</ticketing_fee_amount>

<reservation_fee_amount>0</reservation_fee_amount>

<commission_amount>0</commission_amount>

<fare_amount_incl>161</fare_amount_incl>

<tax_amount_incl>0</tax_amount_incl>

```



```

<yq_amount_incl>0</yq_amount_incl>

<ticketing_fee_amount_incl>0</ticketing_fee_amount_incl>

<reservation_fee_amount_incl>0</reservation_fee_amount_incl>

<commission_amount_incl>0</commission_amount_incl>

<acct_net_total>805</acct_net_total>

<acct_fare_amount>675</acct_fare_amount>

<acct_tax_amount>0</acct_tax_amount>

<acct_yq_amount>0</acct_yq_amount>

<acct_ticketing_fee_amount>0</acct_ticketing_fee_amount>

<acct_reservation_fee_amount>0</acct_reservation_fee_amount>

<acct_commission_amount>0</acct_commission_amount>

<acct_fare_amount_incl>805</acct_fare_amount_incl>

<acct_tax_amount_incl>0</acct_tax_amount_incl>

<acct_yq_amount_incl>0</acct_yq_amount_incl>

<acct_ticketing_fee_amount_incl>0</acct_ticketing_fee_amount_incl>

<acct_reservation_fee_amount_incl>0</acct_reservation_fee_amount_incl>

<acct_commission_amount_incl>0</acct_commission_amount_incl>

<nationality_rcd>TH</nationality_rcd>

<document_type_rcd>B</document_type_rcd>

<client_number>0</client_number>

<vip_flag>0</vip_flag>

<record_locator>LVNWKS</record_locator>

<lastname>INFANTA</lastname>

<firstname>INFANT</firstname>

<date_of_birth>2015-01-01T00:00:00+07:00</date_of_birth>

```



```

<gender_type_rcd>M</gender_type_rcd>

<title_rcd>MR</title_rcd>

<passenger_type_rcd>INF</passenger_type_rcd>

<agency_code>B2C</agency_code>

</Passenger>

</Flight>

</root>

```

The Fields used in the Get Passenger Manifest XML are described in the following table:

Flight		
Name	Type	Description
flight_id	Field	Unique flight ID.
airline_rcd	Field	Airline IATA code.
flight_number	Field	Flight number.
origin_rcd	Field	Origin code.
destination_rcd	Field	Destination code.
departure_date_from	Field	Departure date from.
departure_date_to	Field	Departure date to.
flight_legs_count	Field	Flight legs count.
flight_leg	Element	Flight leg elements.
number_of_leg	Element	Status each of passenger type.
boarding_class	Element	Boarding class each of passenger type.
Passenger	Element	Passenger information.

Flight leg		
Name	Type	Description
flight_id	Field	Unique flight ID.
arrival_airport_rcd	Field	Arrival airport code.
arrival_date	Field	Arrival date.



utc_arrival_date_time	Field	Arrival date time in UTC format.
planned_arrival_time	Field	Planned arrival time.
departure_airport_rcd	Field	Departure airport code.
departure_date	Field	Departure date.
planned_departure_time	Field	Planned departure time.
utc_departure_date_time	Element	Departure date time in UTC format.

Number of leg (each of passenger type)		
<u>Name</u>	<u>Type</u>	<u>Description</u>
Boarded	Field	Passengers boarded count.
Checked	Field	Passengers checked-in count.
Flown	Field	Passengers flown count.
Noshow	Field	Passengers no show count
No_check-in_status	Field	Passengers not check-in count.
Offloaded	Field	Passengers offloaded count.

Boarding class (each of passenger type)		
<u>Name</u>	<u>Type</u>	<u>Description</u>
First class	Field	First class passengers count.
Business class	Field	Business class passengers count.
Economy class	Field	Economy class passengers count.

## 27. APPENDIX 23: GETCOPORATESESSIONPROFILE





```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <GetCoporateSessionProfileResponse xmlns="http://tempuri.org/">

      <GetCoporateSessionProfileResult>

        <![CDATA[<CorporateClient><Corporate>

<ClientProfile>

  <client_profile_id>e3f34e9f-136b-4be8-acdd-fe3d44fad807</client_profile_id>

  <client_type_rcd>COR</client_type_rcd>

  <client_number>60000079</client_number>

  <member_number>B2E00000001</member_number>

  <firstname>Enterprise</firstname>

  <lastname>Aurigney test coporations</lastname>

  <contact_name>samlee ruangpaisan</contact_name>

  <status_code>A</status_code>

  <address_line1>145/8</address_line1>

  <address_line2>Sansiri condominium</address_line2>

  <street>sukhumvit road</street>

  <state>thai</state>

  <district>huykwang</district>

  <province>bangkok</province>

  <city>bangkok</city>

  <zip_code>15847525</zip_code>

  <po_box>856/4</po_box>

  <country_rcd>TR</country_rcd>
```



```

<company_flag>1</company_flag>

<phone_business>045455656</phone_business>

<number_of_children>0</number_of_children>

<phone_fax>052552578</phone_fax>

<contact_email>samlee.r@bravo.aero</contact_email>

<emergency_contact>samlee</emergency_contact>

<language_rcd>EN</language_rcd>

<member_level_rcd>PLATNM</member_level_rcd>

<client_password>123123</client_password>

</ClientProfile>

</Corporate><Employees>

<Employee>

  <client_profile_id>85e6eab4-71e3-423c-baff-68381a95e948</client_profile_id>

  <client_number>60000241</client_number>

  <firstname>YAYA</firstname>

  <lastname>TOORE</lastname>

  <middlename>YAYA</middlename>

  <title_rcd>MR</title_rcd>

  <phone_business>045455656</phone_business>

  <contact_email>samlee.r@bravo.aero</contact_email>

  <country_rcd>TR</country_rcd>

  <user_account_id>85e6eab4-71e3-423c-baff-68381a95e948</user_account_id>

  <status_code>A</status_code>

  <system_admin_flag>0</system_admin_flag>

  <employee_number>0121521222</employee_number>

```



```

    <passenger_profile_id>342f4100-2d45-450c-9a54-e709fdf3027d</passenger_profile_id>

    <passenger_type_rcd>ADULT</passenger_type_rcd>

    <passenger_role_rcd>MYSELF</passenger_role_rcd>

    <vip_flag>0</vip_flag>

    <date_of_birth>1987-01-01T00:00:00+07:00</date_of_birth>

    <gender_type_rcd>M</gender_type_rcd>

    <passenger_weight>0</passenger_weight>

  </Employee>

</Employees></CorporateClient>]]>

  </GetCoporateSessionProfileResult>

  </GetCoporateSessionProfileResponse>

</soap:Body>

</soap:Envelope>

```

CorporateClient		
<u>Name</u>	<u>Type</u>	<u>Description</u>
ClientProfile	Element	Client information.
client_profile_id	Field	Unique Client id
client_type_rcd	Field	Client type
client_number	Field	Unique Client number
member_number	Field	Member number
firstname	Field	Firstname
lastname	Field	Lastmane
contact_name	Field	Contact name
status_code	Field	Status code
address_line1	Field	Address
address_line2	Field	Address



street	Field	Street
state	Field	State
district	Field	District

CorporateClient		
<u>Name</u>	<u>Type</u>	<u>Description</u>
province	Field	Province
city	Field	City
zip_code	Field	Zip code
po_box	Field	Po box
country_rcd	Field	Country code
company_flag	Field	Company flag
phone_business	Field	Phone business
number_of_children	Field	Number of children
phone_fax	Field	Phone fax
contact_email	Field	Contact email
emergency_contact	Field	Emergency contact
language_rcd	Field	Language code
member_level_rcd	Field	Member level code
client_password	Field	Client password



CorporateClient		
<u>Name</u>	<u>Type</u>	<u>Description</u>
Employee	Element	Employee
client_profile_id	Field	Unique Client id
client_number	Field	Unique Client number
firstname	Field	Firstname
lastname	Field	Lastname
middlename	Field	Middlename
title_rcd	Field	Title code
phone_business	Field	Phone business
contact_email	Field	Contact email
country_rcd	Field	Country code
user_account_id	Field	User account id

CorporateClient		
<u>Name</u>	<u>Type</u>	<u>Description</u>
status_code	Field	Status code
system_admin_flag	Field	System admin flag
employee_number	Field	Employee number
passenger_profile_id	Field	Passenger profile id
passenger_type_rcd	Field	Passenger type code



passenger_role_rcd	Field	Passenger role code
vip_flag	Field	Vip flag
date_of_birth	Field	Date of birth
gender_type_rcd	Field	Gender type code
passenger_weight	Field	Passenger weight

