

EFB Webservice

Date: Sep 7, 2023

Latest version: 1.8, revision 48

Endpoint address for production: <https://efb-1-8.ppsflightplanning.com/efbservice.asmx>

Endpoint address for test: <https://preprod-efb-1-8.ppsflightplanning.com/efbservice.asmx>

This document is a work in progress! Not all fields in the return objects are described yet, but revision and expansion of the documentaion happens regularly. Ask the Support staff, if you find something missing, that you specifically need some information about.

Table of contents

- Purpose of web service

 - Data availability

- Authentication

 - Partner and customer terms

 - GetSessionID method

 - GetUserFilteredSessionID

- FlightList methods

 - FlightListItem result object

 - GetFlightListSearch

 - GetSTDFlightList (deprecated)

 - GetChangedList (deprecated)

 - GetFlightList (deprecated)

 - GetTailList (deprecated)

 - GetUserList (deprecated)

 - GetFlightInfoList

- Single flight methods

 - Flight identification

 - GetFlight / GetFlightByExternalID result object

 - Field conventions

 - Flight

 - ATC

 - RoutePoint

 - Notam

 - Crew

 - HoldingFuel

 - GetFlight / GetFlightByExternalID

 - GetATC

 - GetFlightAirports

 - GetWX

 - GetNextFlight

 - GetArinc633FlightLog

 - GetArinc633WBACommon

 - GetArinc633FlightPlanAtclcao

 - GetArinc633UpperAirWeather

 - GetEFF_FullPackage

 - RecalculateFlight

 - Prerequisites for using the recalculation method

 - Parameters

 - Warning: Recalc Vanishing Parameters

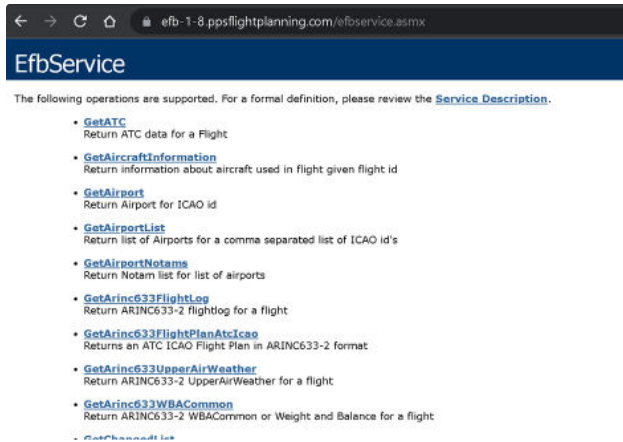
 - RefreshFlight

- Document methods
 - [GetFlightDocumentsMeta](#)
 - [GetFlightDocument](#)
 - [GetPDF_ATC](#)
 - [GetPDF_ShortATC](#)
 - [GetPDF_Logstring](#)
 - [GetPDF_Messages](#)
 - [GetPDF_NOTAMs](#)
 - [GetPDF_WX](#)
 - [GetRSChart](#)
- Additional methods
 - [GetAirport](#)
 - [GetAirportList](#)
 - [GetAirportNotams](#)
 - [GetFirNotams](#)
 - [GetSIGMETs](#)
- Technical support

Purpose of web service

The intention by the EFB web service is to make PPS flight specific data available for partners or customers in order for the consumer to make EFB functionality. The service provides flight data and documents that are equal (or almost equal) to the ones available from the CrewBriefing website. The service is meant to be accessed from a Soap webservice client and cannot be used without a client program that is developed for this purpose. It is the consumer's responsibility to make the necessary arrangement for developing such a client.

By opening the endpoint address in a browser the complete technical description of each of the service methods will be available:



Data availability

The EFB webservice is based on the same data as those used on the CrewBriefing website. Flights on CrewBriefing **are available for up to one week after scheduled time of departure**. After that, it will no longer be possible to view them on CrewBriefing or reading their data by the EFB service.

Authentication

Partner and customer terms

Normally the service is not used and consumed by a PPS customer but by a partner who reads data on behalf of a PPS customer. Thus both a partner and a customer login must be used to authenticate to the service. The partner login is created by us and handed from us to the partner. The customer login is a valid Crewbriefing login from the PPS customer which data should be read. The PPS customer should provide the login to the partner. The service can only be used to access data from customers who have a specific license for the EFB service. If no such agreement, no partner will be able to read data for the customer.

In order to open a service session from a client and authenticate access to a certain account the first method to call on the service must be GetSessionID.

GetSessionID method

Used for authenticating partner and customer to the service

Parameters:

Field	Type	Description
PartnerUser	string	Username to identify as a specific integration partner. If you as a partner serves multiple customers, this username will always be the same across customers.
PartnerPassword	string	Integration partner's password. Both PartnerUser and PartnerPassword are obtained by contacting AIR SUPPORT.
CustomerUser	string	Valid CrewBriefing username - this will identify the customer and constrain which flightplans are available.
CustomerPassword	string	Corresponding CrewBriefing password

Return value: String that contains a session id.

On subsequent calls to other methods on the service the returned session id must be specified. The session and its id is kept alive up until 30 minutes after the latest method call. Thus the GetSessionID method should only be executed once for each session of method calls and not once for each single call.

Remark: In all webservice methods the sessionid must be passed to identify the client to the service. Thus, the sessionid parameter is implicit on all methods and is not mentioned in the description of each webservice method below.

Example request and response:

```
1 Request :
2
3 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:crew="http://crewbriefing.com/">
4   <soap:Header/>
5   <soap:Body>
6     <crew:GetSessionID>
7       <crew:PartnerUser>PARTNER1</crew:PartnerUser>
8       <crew:PartnerPassword>xyzyzy12kspw</crew:PartnerPassword>
9       <crew:CustomerUser>AS9intuser</crew:CustomerUser>
10      <crew:CustomerPassword>JKJ78dsd7xc!cq-e</crew:CustomerPassword>
11    </crew:GetSessionID>
12  </soap:Body>
13 </soap:Envelope>
14
15 Response :
```

```

16
17 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema
18   <soap:Body>
19     <GetSessionIDResponse xmlns="http://crewbriefing.com/">
20       <GetSessionIDResult>c9a235b8-881e-4d61-be96-246687ff32f0</GetSessionIDResult>
21     </GetSessionIDResponse>
22   </soap:Body>
23 </soap:Envelope>

```

GetUserFilteredSessionID

Works exactly as GetSessionID but with the following additions:

If the following conditions are met, the flight lists read using the session are restricted:

- The Crewbriefing user, passed as parameter, has saved a default search criteria on CrewBriefing
- The user does not have 'User default settings modifications rights' on CrewBriefing
- The user does not have 'Search rights' on CrewBriefing

Under these circumstances, the flightlist on CrewBriefing is locked to flights, meeting the locked search criteria. When using GetUserFilteredSessionID flights returned from the session is locked to the same criteria.

If not all the mentioned conditions are met, the method works exactly as GetSessionID.

FlightList methods

FlightListItem result object

Flight list methods returns a list of all flights that meets the search criteria. The flight objects returned has the most basic information of the flights that are:

Field	Type	Description
ID	integer	Internal numeric FlightID. Used to uniquely identify the flight in subsequent calls to flight methods on the service.
FlightLogID	string	Identifiable representation of the flight, as it is shown in CrewBriefing. This representation consist of FLT NO, DEP, and DEST
GUFI	guid	Globally Unique Flight Identification
DEP	string	Departure airport (ICAO code)
DEST	string	Destination airport (ICAO code)
Alt1	string	Destination alternate 1 (ICAO code)
Alt2	string	Destination alternate 2 (ICAO code)
STD	DateTime	Scheduled time of departure
STA	DateTime	Scheduled time of arrival
ETA	DateTime	Estimated time of arrival
ACFTAIL	string	Tail number
ATCTime	string	ATC Scheduled Time of departure (string format is HHMM, eg. 1809)

ATCCtot	string	ATC calculated takeoff time (string format is HHMM, eg. 0730)
ATCDOF	string	ATC date of flight (string format is YYMMDD, eg. 190731)
ATCEET	string	ATC estimated enroute time / flight time (string format is HHMM, eg. 0138)
ReleasedForDispatch	boolean	
LastEdit	DateTime	Timestamp for the last time anything was changed concerning the flight (CrewBriefing upload, recalculation, new documents, new weather etc.)
LatestFlightPlanDate	DateTime	Timestamp for last change to the actual flightplan (CB upload, recalculation)
LatestDocumentUploadDate	DateTime	Timestamp for last uploaded document (including weather charts)
Crew	Array<Crew>	List of crew-members. See the Crew table for specific content
AirportIATACodes	Object	Object containing dep, dest, alt1, alt2 represented by their 3-letter IATA codes
CommercialFlightNumber	string	Commercial FlightNumber
IsRecalc	boolean	Flagging to show if flight has been recalculated
CustomReferences	object	Object containing custom RefID and MilID
AMEXSYStatus	string	(Deprecated - use CFMUStatus)
CFMUStatus	string	Filing response status. Possible values: <ul style="list-style-type: none"> • (blank) Not yet filed • MAN • REJ • PENDING • ACK • NotSent • Sent • DES • SRM • CNL • FLS • SAM • SLC

GetFlightListSearch

Returns a list of flights that conforms to the specified search criteria. This method is a combination of GetChangedList, GetSTDFlightList, GetTailList, GetFlightList, and GetUserList, **and those methods should be considered deprecated.**

All search-parameters are optional, and can be combined in any way as you deem fit. The parameters work as an increasing constraint on the matched flights. This means that a flight needs to match every search-criteria that you specify.

An example could be to find all flights departing from Copenhagen Airport within the next 48 hours. The method should then be called with:

```
1 ToSTD = (today + 48 hours)
2 Departure = EKBI
```

Note about the DateTime parameters: Although all search parameters are optional, the DateTime parameters always need to be specified, due to a technical constraint. To avoid using one of these, just add a date before the year 2000, eg. **1900-01-01 00:00:00**.

Available search parameters:

Field	Type	Description
SessionID	guid	Authorization token. This token is obtained by first calling the method GetSessionID .
FromSTD	DateTime	Choose all flights with scheduled time of departure on this date/time or later
ToSTD	DateTime	Choose all flights with scheduled time of departure on this date/time or sooner
ChangedAfter	DateTime	Choose all flights that have been uploaded to CrewBriefing or changed (recalculation, ctot, etc.) since this date/time
Crew	array<string >	Choose all flights containing the specified list of crewmembers. This parameter is used together with the next one called CrewAnded .
CrewAnded	boolean	When true , all crewmembers in the list must be present in the flight. When false , just one or more crewmembers need to be present in the flight.
FlightNumber	array<string >	Choose all flights that have one of the flightnumbers in the list.
TailNumber	array<string >	Choose all flights that have one of the aircraft-registrations (tail-numbers) in the list
Departure	string	Choose all flights departing from this airport (ICAO and/or IATA can be specified)
Destination	string	Choose all flights arriving to this airport (ICAO and/or IATA can be specified)

GetSTDFlightList (deprecated)

Please use GetFlightListSearch instead and fill in FromSTD and ToSTD.

GetChangedList (deprecated)

Please use GetFlightListSearch instead and fill in ChangedAfter.

GetFlightList (deprecated)

Please use GetFlightListSearch instead and fill in FlightNumber.

GetTailList (deprecated)

Please use GetFlightListSearch instead and fill in TailNumber.

GetUserList (deprecated)

Please use GetFlightListSearch instead and fill in Crew and CrewAanded.

GetFlightInfoList

Returns a list of CrewBriefing flights containing only the most basic information like:

- Id
- Callsign
- Scheduled time of departure
- Day of flight
- Departure airport
- Destination airport

The list will only contain flights, that are uploaded to the Amexsy system.

Single flight methods

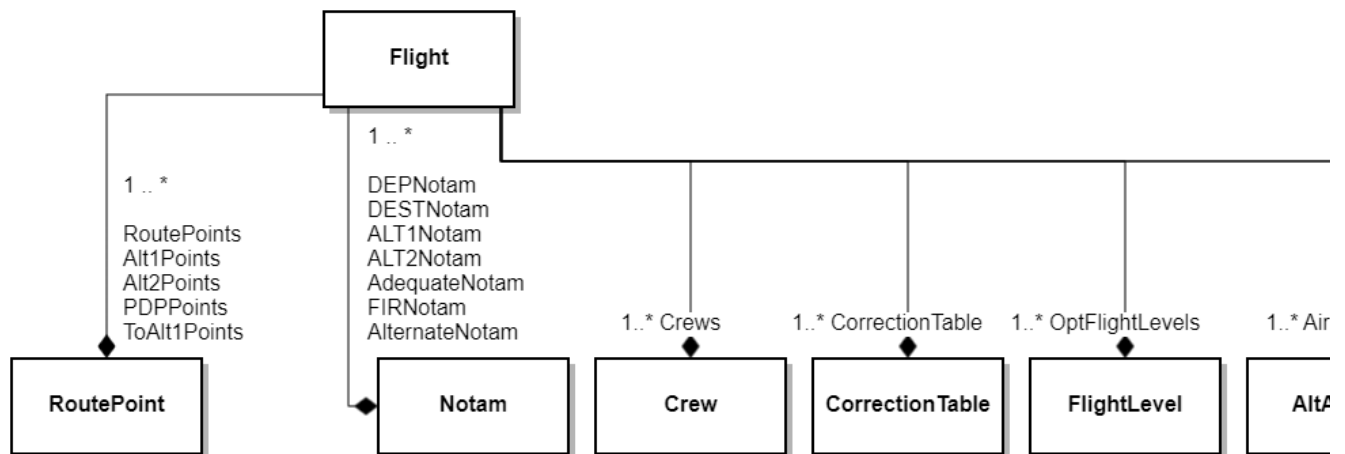
This section covers methods, that retrieves detailed data about one single flight at a time. The methods differ in the data, that they return, and for some also the format.

Flight identification

For all methods, save one, the identifying input-parameter is **FlightID**. This ID is obtained as part of the resultset from **GetFlightListSearch**.

GetFlight / GetFlightByExternalID result object

The returned flight data is a complex object with related data, described in the following diagram



Not all fields have yet been described. Refer to the method/field overview in the webservice for more details.

Field conventions

All weight fields (fuel, load, mass) are default in **pounds**. It is also possible to access the values in both **kilo** and **ton**.

All numerical "time" fields are always in **minutes**.

All distance fields are in **nautical miles**.

All speed fields are in **knots**.

All temperature fields are in **degree celcius**.

All direction/angle fields are in **degrees**.

Flight

Field	Type	Description
ID	integer	Internal numeric FlightID. Same FlightID as specified in the method call.
FlightLogID	string	Identifiable representation of the flight, as it is shown in CrewBriefing. This representation consist of FLT NO, DEP, and DEST
PPSName	string	Customer identification
GUFI	guid	Globally Unique Flight Identification
DEP	string	Departure airport (ICAO code)
DEST	string	Destination airport (ICAO code)
Alt1	string	Destination alternate 1 (ICAO code)
Alt2	string	Destination alternate 2 (ICAO code)
STD	DateTime	Scheduled time of departure
STA	DateTime	Scheduled time of arrival
ETA	DateTime	Estimated time of arrival
ACFTAIL	string	Tail number
ATCTime	string	ATC Scheduled Time of departure (string format is HHMM, eg. 1809)
ATCCtot	string	ATC calculated takeoff time (string format is HHMM, eg. 0730)
ATCDOF	string	ATC date of flight (string format is YYMMDD, eg. 190731)
ATCEET	string	ATC estimated enroute time / flight time (string format is HHMM, eg. 0138)
ReleasedForDispatch	boolean	
LastEditDate	DateTime	Timestamp for the last time anything was changed concerning the flight (CrewBriefing upload, recalculation, new documents, new weather, ATC CTOT etc.)
LatestFlightPlanDate	DateTime	Timestamp for last change to the actual flightplan (CB upload, recalculation, ATC CTOT)
LatestDocumentUploadDate	DateTime	Timestamp for last uploaded document (including weather charts)
Crew	Array<Crew>	List of crew-members. See the Crew table for specific content
AirportIATACodes	Object	Object containing dep, dest, alt1, alt2 represented by their 3-letter IATA codes
CommercialFlightNumber	string	Commerciel FlightNumber

IsRecalc	boolean	Flagging to show if flight has been recalculated
CustomReferences	object	Object containing custom RefID and MillID
PAX	int	Total number of passengers
FUEL	double	
ValidHrs	string	
MinFL	int	Minimum enroute flight level
MaxFL	int	Maximum enroute flight level
EROPSAItApts	string	
AdequateApt	array<string>	
FIR	array<string>	List of all FIR (Flight Information Region) that the flight-route crosses over.
AltApts	array<string>	List of all airport alternates chosen in the flightplan.
TOA	string	Type of Aircraft
FMDID	string	
DESTSTDALT	string	
FUELCOMP	string	Company fuel
TIMECOMP	string	Company time
FUELCONT	string	Contingency fuel (deprecated, use FuelContDef instead)
TIMECONT	string	Contingency time
PCTCONT	string	Contingency percentage (deprecated, use FuelContDef instead)
FUELMIN	string	Minimum ramp fuel
TIMEMIN	string	Minimum ramp time
FUELTAI	string	Taxi fuel
TIMETA	string	Taxi time
FUELEXTA	string	Extra fuel
TIMEEXTA	string	Extra time
FUELLDG	string	Landing fuel
TIMELDG	string	Landing time
ZFM	string	Zero fuel mass
GCD	string	Great circle distance
ESAD	string	Equivalent still air distance (air distance)

GL	string	Gain/Loss
FUELBIAS	string	Fuel bias
LocalTime <ul style="list-style-type: none"> • Departure <ul style="list-style-type: none"> ◦ STD ◦ ETD ◦ Sunrise ◦ Sunset • Destination <ul style="list-style-type: none"> ◦ STA ◦ ETA ◦ Sunrise ◦ Sunset 	DateTime <p>STD = Scheduled Time of Departure</p> <p>ETD = Estimated Time of Departure</p> <p>STA = Scheduled Time of Arrival</p> <p>ETA = Estimated Time of Arrival</p>	
SCHBLOCKTIME	int	Scheduled block time (Minutes from scheduled departure to scheduled destination)
DISP	string	
FUELMINTO	string	Minimum takeoff fuel
TIMEMINTO	string	Minimum takeoff time
ARAMP	string	Ramp mass
TIMEACT	string	
FUELACT	string	Actual fuel endurance
DestERA	string	Enroute alternate
TrafficLoad	string	
WeightUnit	string	Weight unit used in the printed OFP - note that all mass in EFB are always presented in pounds . So this field is only used to specify what to convert to, if the crew wants to see same units as on the OFP.
WindComponent	string	Average wind component for the routing. A negative value depicts a headwind.
CustomerDataPPS	string	(deprecated)
CustomerDataScheduled	string	(deprecated)
FI	int	Enroute flight level
RouteDistNM	int	(deprecated)
RouteName	string	
RouteType	string	
RouteRemark	string	
EmptyWeight	int	

TotalDistance	int	Total distance from departure to destination in NM (nautical miles).
AltDist	int	
DestTime	int	
AltTime	int	Traveling time to Alternate 1
AltFuel	int	Required fuel to Alternate 1
HoldTime	int	Planned holding time
ReserveTime	int	Planned reservice time
Cargo	int	Total cargo
ActTOW	double	Actual take-off weight
TripFuel	double	Fuel to destination
HoldFuel	double	(deprecated, use Holding instead)
Holding	HoldingFuel	
Elw	double	Elevation
FuelPolicy	string	
Alt2Time	int	Traveling time to Alternate 2
Alt2Fuel	int	Required fuel to Alternate 2
MaxTOM	double	Maximum take-off mass
MaxLM	double	Maximum landing mass
MaxZFM	double	Maximum zero-fuel mass
WeatherObsTime	DateTime	Weather observation time
WeatherPlanTime	DateTime	
MFCI	string	Male/Female/Children/Infants
CruiseProfile	string	Cruise profile
TempTopOfClimb	int	Temperature at top-of-climb
Climb	string	Climb data
Descend	string	Descend data
FuelPL	string	Fuel planning
DescendWind	string	Descend wind
ClimbProfile	string	Climb profile
DescendProfile	string	Descend profile
HoldProfile	string	HoldProfile

StepClimbProfile	string	Step climb profile
FuelContDef	string	Contingency fuel definition
FuelAltDef	string	Diversion fuel definition
AmexsyStatus	string	(Deprecated - use CFMUStatus)
AvgTrack	int	Average track (Great circle course)
DEPTAF	Taf	TAF (Terminal Aerodrome Forecast) at departure airport
DEPMetar	string	METAR (METeorological Aerodrome Report) at departure airport
DEPNotam	array<Notam>	NOTAMs (Notice To AirMen) for departure airport
DESTTAF	Taf	TAF (Terminal Aerodrome Forecast) at destination airport
DESTMetar	string	METAR (METeorological Aerodrome Report) at destination airport
DESTNotam	array<Notam>	NOTAMs (Notice To AirMen) for destination airport
ALT1TAF	Taf	TAF at First destination alternate airport
ALT2TAF	Taf	TAF at Second destination alternate airport
ALT1Metar	string	METAR at First destination alternate airport
ALT2Metar	string	METAR at Second destination alternate airport
ALT1Notam	array<Notam>	NOTAMs for first destination alternate airport
ALT2Notam	array<Notam>	NOTAMs for second destination alternate airport
RoutePoints	array<RoutePoint>	Primary route (departure to destination)
Crews	array<Crew>	Crew
Response	complex	(Sorry for the typo - cannot fix it without breaking backwards compatibility) Values describing the technical result from calling the method) <ul style="list-style-type: none"> • Succeed : bool - did the method return a correct result • Message: string - if Succeed is false, this field contains the error message
ATCData	ATC	Values sent to aviation authorities.
NextLeg	complex	Contains DEP, DEST, STD and minimum fuel requirement for aircraft's immediate next flight
OptFlightLevels	array<FlightLevel>	Contains fuel and time profiles for optimal flight levels
AdequateNotam	array<Notam>	
FirNotam	array<Notam>	NOTAMs for the FIR that the route crosses

	m>	
AlternateNotam	array<Notam>	NOTAMs for the alternate airports
Airports	array<AltAirport>	Information about all airports in the flight (dep, dest, alternates, takeoff alternates)
EnrouteAlternates	array<string>	
Alt1Points	array<RoutePoint>	Route to Alternate 1
Alt2Points	array<RoutePoint>	Route to Alternate 2
FW1, FW2, FW3...FW9	string	
TOTALPAXWEIGHT	double	Total passenger weight
LEGFUEL	int	Displays the fuel burn off for each route point. Please note that the user would have to upgrade to PPS8 1.9 SP2.107 or later to get access to LEGFUEL.

ATC

RoutePoint

Notam

Crew

HoldingFuel

GetFlight / GetFlightByExternalID

These two methods are identical - except on the identifying parameter.

GetFlight expects the FlightID as described above.

GetFlightByExternalID can only be used in combination with AIR SUPPORT's **ScheduledFlight Webservice**. It looks for a flight containing the external reference, that can be uploaded as part of the scheduled flight. This should be a unique reference to the flight.

Apart from this, the input parameters are the same:

Field	Type	Description
SessionID	guid	Authorization token. This token is obtained by first calling the method GetSessionID .
FlightID or ExternalFlightId	int string	Unique identification of flight

Taf	boolean	When true, all TAF (Terminal Aerodrome Forecast) data for airports in this flight are added to the result.
Metar	boolean	When true, all METAR (METeorological Aerodrome Report) data for airports in this flight are added to the result.
Notams	boolean	When true, all NOTAM (NOtice To AirMen) data for airports and route-points/waypoints in this flight are added to the result.
Messages	boolean	When true, all messages for this flight are added to the result.
MassUnit	String	It is possible to have load, mass and fuel in either lb (pound), kg (kilo) or t (ton).

Note about optional data: Taf, metar, notam, and messages all add considerable amount of data to the payload returned by the GetFlight method. Therefore it is suggested, that you only choose to include them when needed.

GetATC

Reads ATC data of a flight.

Parameters:

- FlightId

Refer to the webservice for detailed information about the ATC data properties.

Note: All the data in this method are available in the GetFlight method as well.

GetFlightAirports

Reads detailed airport information for departure, destination, alternate 1 and alternate 2 airports of a flight.

Parameters:

- FlightId

Each airport object contains:

- Name
- ICAO code
- IATA code
- Country
- TAF
- Metar
- Elevation
- Magnetic variation
- Latitude
- Longitude
- FIR
- List of runways

GetWX

Returns Taf, Metar and Notam information of a flight.

Parameters:

- FlightId
- Filtered

If set to true Notams will be filtered according to the actual Notam filtering on CrewBriefing.

GetNextFlight

Returns the first flight by the specified tail number which have a scheduled time of departure later than current time.

Parameters:

- Tailnumber
- Taf
Specify by true or false if TAF data for this flight should be returned in result. To reduce load of the service do only select true if TAF is needed.
- Metar
Specify by true or false if METAR data for this flight should be returned in result. To reduce load of the service do only select true if METAR is needed.
- Notams
Specify by true or false if Notams for this flight should be returned in result. To reduce load of the service do only select true if Notams are needed.

Returns same object as GetFlight.

GetArinc633FlightLog

Returns an ARINC633-2 flightlog for a flight.

Parameters:

- FlightId
- massUnit (Optional, Default = "lb" (options: "lb", "kg" and "t"))

Returns a status and if the request succeeds the Arinc 633-2 FlightPlan element as result.

GetArinc633WBACommon

Returns an ARINC633-2 WIISUB (Weight & Balance Section) for a flight.

Parameters:

- FlightId
- massUnit (Optional, Default = "lb" (options: "lb", "kg" and "t"))

Returns a status and if the request succeeds the Arinc 633-2 WIISUB element as result.

GetArinc633FlightPlanAtclcao

Returns an ARINC633-2 Atclcao flightplan for a flight.

Parameters:

- FlightId

Returns a status and if the request succeeds the Arinc 633-2 FlightPlanAtclcao element as result

GetArinc633UpperAirWeather

Returns an ARINC633-2 f for a flight.

Parameters:

- FlightId
- StepSize
- NumberOfSamplesAboveFlightLevel
- NumberOfSamplesBelowFlightLevel

The granularity of Upper Air Weather calculations is specified by the StepSize, NumberOfSamplesAboveFlightLevel and NumberOfSamplesBelowFlightLevel parameters. They specify equidistant samples below and above the flightlevel of the route points. If StepSize=10, NumberOfSamplesBelowFlightLevel = 2, NumberOfSamplesAboveFlightLevel = 3 and the flightlevel of the point is 360, then upper air weather will be calculated for the route point in flightlevels:

- 340, 350 (2 samples below actual flightlevel with distance = 10)
- 360 (actual flight level of route point)
- 370, 380 and 390 (3 samples above actual flightlevel with distance = 10)

In general the upper air weather calculations supports flightlevels in the range from FL50 to FL530.

The method returns a status and if the request succeeds the Arinc 633-2 UpperAirWeather element as result.

GetEFF_FullPackage

Return an ARINC633-2 .eff package for a flight.

Parameters:

- FlightId
- weightUnit (Optional, Default = "lb" (options: "lb", "kg" and "t"))

Returns a status and if the request succeeds the Arinc 633-2 EFF package element as result.

Unlike the other ARINC633-2 methods (returning XML), this returns the result as a complete package (an .eff file with .lst and checksum). In a byte-array, similar to the GetPDF_X methods.

RecalculateFlight

Makes a recalculation of the flight. The recalculation results in a new instance of the flight and the id of this new instance is returned from the method.

Prerequisites for using the recalculation method

- The original flight (specified by the FlightId parameter) must have been uploaded by the PPS8 program (not PPS7)
- The customer must have a license for Recalculation (which not all customers have)
- The informations needed to do recalculation must be available on the recalculation server
This means for instance, that the aircraft file must have been uploaded to the recalculation server.

The way to test, whether all prerequisites are met for a certain flight, is to ensure the flight has been uploaded by PPS8 and that it can be recalculated on CrewBriefing.

Parameters

Required parameter:

- FlightId

In addition to the FlightId some optional parameters can be set, to override the corresponding value on the recalced flight. If a value is not set, the value will be copied from the original flight specified by the FlightId.

Optional parameters:

- Pax
Set this value to override passenger count.
- Cargo
Set this value to override cargo (lbs).
- Fuel
Set this value to override fuel (lbs).
- Alt1
Set this value to override alternate 1 airport ICAO code.
- Alt2
Set this value to override alternate 2 airport ICAO code.
- CrewCMD
Set this value to override pilot in command.
- CrewCOP
Set this value to override co-pilot.
- CrewCA1
Set this value to override cabin attendant 1
- CrewCA2
Set this value to override cabin attendant 2
- CrewCA3
Set this value to override cabin attendant 3
- CrewCA4
Set this value to override cabin attendant 4
- CrewCA5
Set this value to override cabin attendant 5
- CrewACM1
Set this value to override additional crewmember 1
- CrewACM2
Set this value to override additional crewmember 2

Warning: Recalc Vanishing Parameters

Note that far too often there is a case where Recalc removes data like crew when due to lack of change it's not submitted. What it means is that whatever changes you make you must supply all data else the not supplied parameters will be considered removed.

RefreshFlight

Makes a recalculation of the flight based on the latest weather informations. The recalculation results in a new instance of the flight and the id of this new instance is returned from the method.

Prerequisites for using this method are the same as specified for the RecalculateFlight method.

Parameters:

- FlightId

Document methods

The document methods are all used for reading pdf documents equivalent to those available on CrewBriefing.

When a pdf is read it is returned as a bytearray which must be streamed to a file on the client to be opened.

GetFlightDocumentsMeta

Returns a list of available uploaded and weather chart documents for a specific flight. The list does not contain the pdf's itself but meta data related to each document. By using the GetFlightDocument method and passing the document identifier the actual pdf document can be read.

Parameters:

- FlightId

Each item in the list has the following document properties:

- Title
Name of the document
- Valid from and valid to
The period in which the document is active
- FlightId
Id of the flight to which the document belongs
- Identifier
Unique identifier that can be used for reading the pdf
- Type
The type of the document. Currently Uploaded, WeatherChart and RunwayAnalysis are available
- Category
Category of document
- Info1
Additional info about the document
- Info2
Additional info about the document

GetFlightDocument

Returns the specified pdf.

Parameters:

- DocumentIdentifier
Id of the document, which has been read by the GetFlightDocumentsMeta method.

GetPDF_ATC

Returns the full ATC pdf of a flight.

Parameters:

- FlightId

GetPDF_ShortATC

Returns the short ATC pdf of a flight.

Parameters:

- FlightId

GetPDF_Logstring

Returns the pdf flightlog.

Parameters:

- FlightId

GetPDF_Messages

Returns a pdf showing messages of the specified flight.

Parameters:

- FlightId
- ImportantOnly
If true only important messages are included in the document

GetPDF_NOTAMs

Returns a pdf showing notams of the specified flight.

Parameters:

- FlightId
- HalfSizePages
If true the document is generated in landscape format

GetPDF_WX

Returns a pdf showing WX of the specified flight.

Parameters:

- FlightId
- HalfSizePages
If true the document is generated in landscape format

GetRSChart

Returns a pdf showing weather charts of the specified flight.

Parameters:

- FlightId
- HalfSizePages
If true the document is generated in landscape format

Additional methods

GetAirport

Returns detailed information of an airport.

Parameters:

- ICAO
ICAO code of airport

The returned airport object has the following properties:

- Name

- ICAO code
- IATA code
- Country
- TAF
- Metar
- Elevation
- Latitude
- Longitude
- Runway lengths
- FIR

GetAirportList

Returns a list of airport objects like the GetAirport method.

Parameters:

- ICAOList
A comma separated list of ICAO codes

GetAirportNotams

Returns notams for the specified airports codes.

Parameters:

- ICAOList
A list of airport ICAO codes
- ValidTime
Read only notams that are active on the specified date
- Filtered
If set to true the CrewBriefing Notam filter will be used to filter the notams

GetFirNotams

Returns notams for the specified FIRs.

Parameters:

- ICAOList
A list of FIR ICAO codes
- FromFlightLevel and ToFlightLevel
The interval of flightlevels to read
- ValidTime
Read only notams that are active on the specified date
- Filtered
If set to true the CrewBriefing Notam filter will be used to filter the notams

GetSIGMETs

Returns SIGMETs for the specified FIRs.

Parameters:

- [ICAOList](#)
A list of FIR ICAO codes

Technical support

For technical help please add your questions to servicesupport@airsupport.dk. For domain specific questions please contact support@airsupport.dk.