Crew Web Service Description

This document describes the web services implemented in the PDC FlightCrew system. The web service use the JSON protocol.

June 2017: Currently PDC FlightCrew supports two web services ‘PDCCrew’ and ‘PDCCrew2’ with separate methods.

The methods from PDCCrew will be replaced with methods in PDCCrew2, and PDCCrew will be deprecated. The methods in PDCCrew2 will not be identical to the methods in PDCCrew but will support the same functionality.

Contents

[Web service PDCCrew 4](#_Toc93046471)

[Data Types used PDCCrew and PDCCrew2 4](#_Toc93046472)

[String 4](#_Toc93046473)

[StringList 4](#_Toc93046474)

[DateOffset 4](#_Toc93046475)

[MinuteOffset 4](#_Toc93046476)

[Integer 4](#_Toc93046477)

[CCXReportKey 4](#_Toc93046478)

[CCXCrewAct 5](#_Toc93046479)

[CCXCrewAsg 5](#_Toc93046480)

[FlightLeg 6](#_Toc93046481)

[User 6](#_Toc93046482)

[Methods in PDCCrew 7](#_Toc93046483)

[version 7](#_Toc93046484)

[crewLogin 7](#_Toc93046485)

[getFlightLegKeysByCrewAndDate 7](#_Toc93046486)

[getLandingApproachList 7](#_Toc93046487)

[getPilotsLogCrewData 8](#_Toc93046488)

[setPilotsLogCrewData 8](#_Toc93046489)

[getDutyTimesByCrewDate 8](#_Toc93046490)

[setDeadheadReservation 9](#_Toc93046491)

[Web service PDCCrew2 10](#_Toc93046492)

[Data Types used PDCCrew2 10](#_Toc93046493)

[error 10](#_Toc93046494)

[notification 10](#_Toc93046495)

[noteRosterActivity 10](#_Toc93046496)

[crewAlert 11](#_Toc93046497)

[checkInRosterActivity 12](#_Toc93046498)

[otherCrew 13](#_Toc93046499)

[crewQualification 14](#_Toc93046500)

[crewCourse 14](#_Toc93046501)

[subCourse 15](#_Toc93046502)

[subCourseTraining 15](#_Toc93046503)

[FlightLegKey 15](#_Toc93046504)

[crewInfo 16](#_Toc93046505)

[fdpInfo 16](#_Toc93046506)

[crewSchedule 16](#_Toc93046507)

[hotelInfo 16](#_Toc93046508)

[rosterActivity 17](#_Toc93046509)

[otherCrew 19](#_Toc93046510)

[crewRequest 20](#_Toc93046511)

[qualificationDef 21](#_Toc93046512)

[trainingEvent 21](#_Toc93046513)

[trainer 21](#_Toc93046514)

[student 22](#_Toc93046515)

[activityTypeDef 22](#_Toc93046516)

[crewPositionDef 23](#_Toc93046517)

[Methods in PDCCrew2 24](#_Toc93046518)

[version 24](#_Toc93046519)

[login 24](#_Toc93046520)

[getNotification 25](#_Toc93046521)

[acknowledgeNotification 25](#_Toc93046522)

[getNextCheckIn 26](#_Toc93046523)

[doCheckIn 27](#_Toc93046524)

[getQualification 27](#_Toc93046525)

[updateQualification 28](#_Toc93046526)

[createCrewmember 29](#_Toc93046527)

[updateCrewmember 30](#_Toc93046528)

[updateEmployment 31](#_Toc93046529)

[getTripSheetInformation 31](#_Toc93046530)

[getRoster 32](#_Toc93046531)

[getCrewCourse 32](#_Toc93046532)

[getCrewRequest 33](#_Toc93046533)

[deleteCrewRequest 34](#_Toc93046534)

[createCrewRequest 35](#_Toc93046535)

[getAllQualificationDef 36](#_Toc93046536)

[getTrainingEvents 36](#_Toc93046537)

[getAllActivityTypeDef 36](#_Toc93046538)

[getAllCrewPositionDef 37](#_Toc93046539)

[Methods inherited from PDCCrew 38](#_Toc93046540)

[HTTP Error Codes 39](#_Toc93046541)

[Client Errors (4XX) 39](#_Toc93046542)

[Server Errors (5XX) 39](#_Toc93046543)

# Web service PDCCrew

This web service will eventually be replaced with PDCCrew2.

## Data Types used PDCCrew and PDCCrew2

This section contains a list of the data types used.

### String

Ordinary string type.

JSON example:  
 “arr”:”CPH”

### StringList

List of strings.

JSON example:  
“reports”:[“This is the first report”,””,”Third report”]

### DateOffset

Dates are represented as number of milliseconds from 1/1-1970. So 20/8-2012 is represented as :

Days since 1/1-1970: 734735 – 719163 = 15572

Milliseconds per day: 86400000

Result = 15572 \* 86400000 = 1345420800000

### MinuteOffset

Timestamps are represented by minute offsets, which has the same format as Dateoffset.

### Integer

Ordinary integer type.

### CCXReportKey

Complex key with the following definition:

|  |  |  |  |
| --- | --- | --- | --- |
| *CCXReportKey* | *Columns* | *Data Type* | *Description* |
|  | crewId | String | Crew ID |
|  | flightId | String | Flight ID |
|  | datop | Dateoffset | Date of operation |
|  | dep | String | Departure Station |
|  | arr | String | Arrival Station |
|  | legNo | Integer | Leg number (e.g. 100 for first leg) |

JSON example: "key":{"crewId":"1","flightId":"PDC1010","datop":1343865600000,"dep":"CPH","arr":"ARN","legNo":100}

### CCXCrewAct

|  |  |  |  |
| --- | --- | --- | --- |
| *CCXCrewAct* | *Columns* | *Data Type* | *Description* |
|  | takeoffNightTime | Boolean | True/False – Take off during night time |
|  | touchdownNightTime | Boolean | True/False – Touchdown during night time |
|  | manual | Boolean | True/False – Manual Landing approach |
|  | approach | String | Name of Approach type |
|  | crewList | Array of CCXCrewAsg |  |

### CCXCrewAsg

|  |  |  |  |
| --- | --- | --- | --- |
| *CCXCrewAsg* | *Columns* | *Data Type* | *Description* |
|  | crewCode | String | Crew ID |
|  | pic | Boolean | Pilot In Command |
|  | dutyCode | String | Duty Code |
|  | position | String | “FC”, “FO”, etc. |
|  | fullName | String | Full Name |
|  | lineCheck | String Array | Array of Names of training activities |
|  | pass | Boolean Array | Array of Passed/failed for the trainings |
|  | done | Boolean Array | Array of Done/not done for the trainings |
|  | takeoffFlying | Boolean | True/False |
|  | takeoffMonitoring | Boolean | True/False |
|  | touchdownFlying | Boolean | True/False |
|  | touchdownMonitoring | Boolean | True/False |
|  | approachFlying | Boolean | True/False |
|  | approachMonitoring | Boolean | True/False |

JSON example: "crewList":[ {"crewCode":"1","position":"FC","fullName":"CAPTAIN XXX", "pic":true, "takeoffFlying":true, "takeoffMonitoring":false,"touchdownFlying":true,"touchdownMonitoring":false,"done":[],"dutyCode":"", "lineCheck":[],"pass":[]}, {"crewCode":"2","position":"FO","fullName":"FO XXX","pic":false, "takeoffFlying":false, "takeoffMonitoring":true,"touchdownFlying":false,"touchdownMonitoring":true,"done":[],"dutyCode":"", "lineCheck":[],"pass":[]}

### FlightLeg

FlightLeg data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *FlightLeg* | *Columns* | *Data Type* | *Description* |
|  | fltId | String | Flight ID |
|  | datop | Dateoffset | Date of operation |
|  | acReg | String | Aircraft Registration |
|  | depStn | String | Departure Station |
|  | arrStn | String | Arrival Station |
|  | status | String | Status string (“scheduled” etc.) |
|  | STD | MinuteOffset | Time of Departure |
|  | STA | MinuteOffset | Time of Arrival |
|  | acVersion | String | Aircraft version |
|  | STC | String | (“J” etc.) |

JSON example: leg":{"fltId":"6YT5454","datop":1378425600000,"acReg":"40V","status":"scheduled","depStn":"CPH","arrStn":"ARN","STD":1378461600000,"STA":1378472400000,"acVersion":"","STC":"J"}

### User

User data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *User* | *Columns* | *Data Type* | *Description* |
|  | username | String | User id |

"login":{"username":"pdcguest"}

## Methods in PDCCrew

All request parameters are in a “params” object, and all response fields are in a “result” object.

This section contains the web services with their request and responses. Fields that are not mandatory in the request can be omitted and default values are used. Note that all fields in the JSON request and response are case sensitive.

### version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| version | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | methods | Integer | Yes | DeadheadReservation Seqno |
| Response | result | methods |  | Array of method descriptions |

JSON example:

{"jsonrpc":"2.0","id":2,"method":"version","params":{"methods":[]}}

### crewLogin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| crewLogin | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | username | String | Yes | User id |
|  | password | String | Yes | Password |
| Response | Result | String |  | “Login succeeded” / “Login failed” |
|  | crewCategory | String |  | “Flight Deck” /”Cabin Crew” /”Non Flying” |

JSON example:  
"method":"crewLogin","params":{"username":"userX","password":"test"}

### getFlightLegKeysByCrewAndDate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| getFlightLegKeysByCrewAndDate | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | crewId | String | Yes | Crew ID |
|  | datop | Dateoffset | Yes | Date of operation |
| Response | key | CCXReportKey |  | Described in Data Type section |

JSON example:  
"method":"getFlightLegKeysByCrewAndDate","params":{"crewId":"1","datop":1398895200000}

### getLandingApproachList

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| getLandingApproachList | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | Key | CCXReportKey | Yes | Described in Data Type section |
| Response |  | StringList |  | Lists of strings |

JSON example:  
"method":"getLandingApproachList","params":{"key":{"arr":"ARN","crewId":"1","datop":1398902400000, "dep":"CPH","flightId":"PDC1010","legNo":100}

### getPilotsLogCrewData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| getPilotsLogCrewData | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | Key | CCXReportKey | Yes | Described in Data Type section |
| Response | approach | String |  | Approach Type |
|  | Manual | Boolean |  | True/False – Manual approach |
|  | crewList | Array of CCXCrewAsg |  | Described in Data Type section |
|  | takeoffNightTime | Boolean |  | True/False |
|  | touchdownNightTime | Boolean |  | True/False |

JSON example:  
"method":"getPilotsLogCrewData","params":{"key":{"arr":"ARN","crewId":"1","datop":1398902400000, "dep":"CPH","flightId":"PDC1010","legNo":100}

### setPilotsLogCrewData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| setPilotsLogCrewData | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | Key | CCXReportKey | Yes | Described in Data Type section |
|  | Value | CCXCrewAct | Yes | Described in Data Type section |
| Response |  | String |  | Update Succeeded / Update Failed |

JSON example:  
"method":"setPilotsLogCrewData","params":{"key":{"arr":"ARN","crewId":"1","datop":1398902400000, "dep":"CPH","flightId":" PDC1010","legNo":100}, "value":{"approach":"CAT","crewList":[ {"crewCode":"1","position":"FC","fullName":"Captain XXX", "pic":true, "takeoffFlying":true, "takeoffMonitoring":false,"touchdownFlying":true,"touchdownMonitoring":false,"done":[],"dutyCode":"", "lineCheck":[],"pass":[]}, {"crewCode":"2","position":"FO","fullName":"FO XXX","pic":false, "takeoffFlying":false, "takeoffMonitoring":true,"touchdownFlying":false,"touchdownMonitoring":true,"done":[],"dutyCode":"", "lineCheck":[],"pass":[]}],"takeoffNightTime":false,"touchdownNightTime":true,"manual":false, "approachFlying":false, "approachMonitoring":false}

### getDutyTimesByCrewDate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| getDutyTimesByCrewDate | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | crewCode | String | Yes | Crew Code |
|  | Date | Dateoffset | Yes | Date of duty |
| Response | biologicalNightDuty | String |  | Bilogical Night Duty |
|  | checkInBaseOffset | Integer |  |  |
|  | checkInTime | Dateoffset |  | Check In Time |
|  | checkOutBaseOffset | Integer |  |  |
|  | checkOutTime | Dateoffset |  | Check Out Time |
|  | crewCode | String |  | Crew Code |
|  | fdpStart | Dateoffset |  | FDP Start |
|  | fdpTime | Integer |  | FDP Time |
|  | inFlightControlTime | Integer |  | Control time in flight |
|  | inFlightReliefTime | Integer |  | Relief time in flight |
|  | isSplitDuty | Boolean |  | Returns true or false |
|  | Name | String |  | Crew Name |
|  | nightBlockHours | Integer |  | Night Block Hours |
|  | Position | String |  | Crew Position |
|  | splitDutyFromBaseOffset | Integer |  |  |
|  | splitDutyFromTime | Dateoffset |  | Split Duty From Time |
|  | splitDutyToBaseOffset | Integer |  |  |
|  | splitDutyToTime | Dateoffset |  | Split Duty To Time |

JSON example:

"method":"getDutyTimesByCrewDate","params":{"crewCode":"kav","date":1431122400000}

### setDeadheadReservation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| setDeadheadReservation | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | seqno | Integer | Yes | DeadheadReservation Seqno |
|  | status | String | Yes | Booked/Cancelled |
|  | bookingID | String |  |  |
| Response | result | null |  | When update successful |
|  | error | error |  | Error information |

JSON example:

"method":"setDeadheadReservation","params":{"seqno":12345,"status":"Booked","bookingID":"ABC123"}

NB: If anything goes wrong or update is not successful the result will be replaced by error

# Web service PDCCrew2

## Data Types used PDCCrew2

### error

Prolog JSON error-object.

|  |  |  |  |
| --- | --- | --- | --- |
| *Error* | *Columns* | *Data Type* | *Description* |
|  | code | Integer | Error code |
|  | message | String | Error message |

"error":{"code":-32098,"message":"Very bad error happened"}

### notification

Notification data type:

|  |  |  |  |
| --- | --- | --- | --- |
| ***notification*** | ***Columns*** | ***Data Type*** | ***Description*** |
|  | notificationSeqno | Integer | Identity seqno of the notification |
|  | date | Dateoffset | The date of the notification |
|  | revisionNum | String | Revision number of the notification |
|  | oldRoster | Array of noteRosterActivity | List of the activities assigned to the crewmember before the change |
|  | newRoster | Array of noteRosterActivity | List of the activities assigned to the crewmember after the change |

### noteRosterActivity

noteRosterActivity data type:

|  |  |  |  |
| --- | --- | --- | --- |
| ***noteRosterActivity*** | ***Columns*** | ***Data Type*** | ***Description*** |
|  | refDate | Dateoffset | Reference date depending on the date mode Z, L (local to start of ftp) or B (crew base) selected for CrewConnex |
|  | pairingId | String | If the activity is part of a pairing then this tag will include the pairing id |
|  | activityIsChanged | Boolean | Only used if part of a notification in the old and new activities. Indicates if the activity is changed as part of the notification. |
|  | dutyCode | String | Duty code this will be SN (supernumerary), DH (deadhead), R\* (recency training), C\* (course training), T\* (other training), I\* (Instructor), Position code if the position requirement is specialized or if the crewmember work low or high rank, else blank. |
|  | checkInTime | MinuteOffset | If the check-in time should be shown on CrewConnex then this field will contain the check-in date and time in UTC. |
|  | checkOutTime | MinuteOffset | If the check-out time should be shown on CrewConnex then this field will contain the check-out date and time in UTC. |
|  | activityName | String | The name of the activity. For Flights it is the FltId, for activities it is the activity type code |
|  | stnFrom | String | Station code for the departure/start station |
|  | startTime | MinuteOffset | Start time for the activity in UTC. For check-in it is the STD of the flight. If not for check-in it is the start time of the flight corresponding to the MVT status of the flight leg. |
|  | stnTo | String | Station code for the arrival/end station |
|  | finishTime | MinuteOffset | Finish time for the activity in UTC. For check-in it is the STA of the flight. If not for check-in it is the end time of the flight corresponding to the MVT status of the flight leg. |

### crewAlert

crewAlert data type:

|  |  |  |  |
| --- | --- | --- | --- |
| ***crewAlert*** | ***Columns*** | ***Data Type*** | ***Description*** |
|  | crewCode | String | Crew code of the crewmember |
|  | seqno | Integer | Identity of the alert |
|  | timeOfAlert | MinuteOffset | The time that the alert is related to |
|  | text | String | The alert text |

### checkInRosterActivity

checkInRosterActivity data type:

|  |  |  |  |
| --- | --- | --- | --- |
| ***checkInRosterActivity*** | ***Columns*** | ***Data Type*** | ***Description*** |
|  | crewCode | String |  |
|  | activitySeqno | Integer | Identification of the roster activity |
|  | trainingType | String | If this is a training activity that can be updated using the groundTraining methods then the type is returned: SIM: training on a simulator duty, GND: training on a ground duty. Else blank. |
|  | pairingSeqno | Integer | If this activity is part of a Pairing then the pairing identification is returned |
|  | refDate | Dateoffset | Reference date depending on the date mode Z, L (local to start of ftp) or B (crew base) selected for CrewConnex |
|  | dayRosterType | String | You are not currently using day roster types, so leave it. |
|  | pairingId | String | If the activity is part of a pairing then this tag will include the pairing id |
|  | revisionNum | String | If notifications has been released on this day the revisionNum will contain the revision number |
|  | activityIsChanged | Boolean | Only used if part of a notification in the old and new activities. Indicates if the activity is changed as part of the notification. |
|  | dutyCode | String | Duty code this will be SN (supernumerary), DH (deadhead), R\* (recency training), C\* (course training), T\* (other training), I\* (Instructor), Position code if the position requirement is specialized or if the crewmember work low or high rank, else blank. |
|  | checkInTime | MinuteOffset | If the check-in time should be shown on CrewConnex then this field will contain the check-in date and time in UTC. |
|  | checkOutTime | MinuteOffset | If the check-out time should be shown on CrewConnex then this field will contain the check-out date and time in UTC. |
|  | activityName | String | The name of the activity. For Flights it is the FltId, for activities it is the activity type code |
|  | activityTypeDesc | String | Description of the activity type if not a flight leg |
|  | stnFrom | String | Station code for the departure/start station |
|  | startTime | MinuteOffset | Start time for the activity in UTC. For check-in it is the STD of the flight. If not for check-in it is the start time of the flight corresponding to the MVT status of the flight leg. |
|  | stnTo | String | Station code for the arrival/end station |
|  | finishTime | MinuteOffset | Finish time for the activity in UTC. For check-in it is the STA of the flight. If not for check-in it is the end time of the flight corresponding to the MVT status of the flight leg. |
|  | acType | String | The aircraft type for flight legs, else blank |
|  | absence | Boolean | True/False if the activity is defined as absence |
|  | turnTime | String | Duration in H:MM of the turn time to the next flight leg, if this activity is a flight leg else blank. |
|  | blockHours | String | Blockhours of the duty in H:MM shown on the last activity of the duty |
|  | dutyHours | String | DutyHours of the duty in H:MM shown on the last activity of the duty |
|  | fdpTime | String | The FDP time of the duty in H:MM shown on the last activity of the duty |
|  | maxFdpTime | String | The Max FDP time of the duty in H:MM shown on the last activity of the duty. If Max fdp time is used in the rules, blank if rolling 24-hours are used to determine the max fdp time. |
|  | Reservation | String | For a deadhead activity this is the reservation code, for a hotel activity this is the hotel name |
|  | acReg | String | If this is a flight leg then it is the Aircraft registration else it is blank |
|  | gate | String | Departure and arrival gate if they are defined Flight |
|  | bookedPax | String | Number of booked pax |
|  | otherCrew | Array of otherCrew | An array of information for the other crewmembers assigned to the activity |
|  | fltNotes | String | The flight notes from the flight leg shown as ../…/../… |
|  | dutyStart | MinuteOffset | If the activity is part a duty, an FDP or a rest period then this is the start time of the duty/fdp/rest period in UTC. |
|  | dutyEnd | MinuteOffset | If the activity is part a duty, an FDP or a rest period then this is the end time of the duty/fdp/rest period in UTC. |
|  | isRest | Boolean | True if the activity is part of a duty or an FDP else False. |

### otherCrew

otherCrew data type:

|  |  |  |  |
| --- | --- | --- | --- |
| ***otherCrew*** | ***Columns*** | ***Data Type*** | ***Description*** |
|  | crewCode | String | Crew code of the crewmember |
|  | employmentNum | String | Employment number of the crewmember |
|  | fullName | String | Full name of the crewmember |
|  | isDeadhead | Boolean | True/False is this crewmember assigned to the flight as deadhead. Remaining fields are only filled out if False. |
|  | position | String | Position name of the assigned crewmember |
|  | languageQualifications | Array of String | List of the language qualifications that this crewmember has |
|  | otherQualifications | Array of String | List of other qualifications that this crewmember has and that is selected to be shown on CrewConnex |
|  | previousDuty | String | Name of the previous duty that this crewmember was assigned to |

### crewQualification

crewQualification data type:

|  |  |  |  |
| --- | --- | --- | --- |
| ***crewQualification*** | ***Columns*** | ***Data Type*** | ***Description*** |
|  | crewCode | String |  |
|  | crewQualificationSeqno | Integer | Identification of the crew qualification |
|  | qualificationName | String | Name of qualification |
|  | parameter | String | Optional parameter for the qualification. Will be the AC-type for operating qualifications |
|  | position | String | Optional position for the operating qualifications |
|  | grade | String | Optional grade for the qualification |
|  | validityStart | Dateoffset | The first date that the crewmember has this qualification |
|  | validityEnd | Dateoffset | The last date that the crewmember has this qualification |
|  | lastTrainingDate | Dateoffset | The last training date where the crewmember passed the training |
|  | lastTraining | String | Activity type of last training |
|  | expiryDate | Dateoffset | The current expiry date of the qualification |
|  | lastExpiryDate | Dateoffset | The last expiry date of the qualification |
|  | passportNumber | String | For the passport qualification the passport number is returned |
|  | passportCountry | String | For the passport qualification the passport issue country is returned |
|  | passportNumber2 | String | For the 2nd passport qualification the passport number is returned |
|  | passportCountry2 | String | For the 2nd passport qualification the passport issue country is returned |
|  | licenseNumber | String | For the license qualification the license number is returned |
|  | licenseCountry | String | For the license qualification the license issue country is returned |
|  | licenseType | Sting | For the license qualification the license type is returned |

### crewCourse

crewCourse data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *crewCourse* | *Columns* | *Data Type* | *Description* |
|  | crewCode | String |  |
|  | crewCourseSeqno | Integer | Identification of the crew course |
|  | courseName | String | Name of the Course |
|  | startDate | Date | Start of the course |
|  | endDate | Date | End of the Course |
|  | isClosed | Boolean | Is the Course closed |
|  | isExclusive | Boolean | Is the related flying allowed during the course |
|  | isRecovery | Boolean | Is this a recovery course |
|  | subCourses | Array of subCourse |  |

### subCourse

subCourse data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *subCourse* | *Columns* | *Data Type* | *Description* |
|  | subCourseSeqno | Integer | Identification of the sub course |
|  | id | String | Id of the sub course |
|  | minOccurrences | Integer | Minimum number of occurrences required |
|  | minDuration | Integer | Minimum duration if required. Duration in minutes. 0 if no requirement |
|  | performedOccurrences | Integer | Currently Performed occurrences |
|  | performedDuration | Integer | Currently performed duration |
|  | plannedOccurences | Integer | Number of planned occurrences |
|  | plannedDuration | Integer | Duration of planned training in the course |
|  | courseTrainings | Array of subCourseTraining |  |

### subCourseTraining

subCourseTraining data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *subCourseTraining* | *Columns* | *Data Type* | *Description* |
|  | trainingSeqno | Integer | Identification of the training |
|  | status | String | P: Passed F: Failed H: Historic (old training that is not passed)  <blank>: Unset |
|  | plannedDate | Date | When is the training planned |
|  | trainerCrewCode | String | CrewCode of the trainer |
|  | activityId | String | FlightId if it is a flight else the activity code |
|  | duration | Integer | Duration in minutes for the course duration |
|  | accOccurrence | Integer | The occurrence of this training in the course |
|  | accDuration | Integer | The accumulated duration of training within the course |

### FlightLegKey

Complex key with the following definition:

|  |  |  |  |
| --- | --- | --- | --- |
| *FlightLegKey* | *Columns* | *Data Type* | *Description* |
|  | flightId | String | Flight ID |
|  | datop | Dateoffset | Date of operation |
|  | legNo | Integer | Leg number (e.g. 100 for first leg) |
|  | number | Integer | The number that the Crew information in the report is referring to |

### crewInfo

|  |  |  |  |
| --- | --- | --- | --- |
| *crewInfo* | *Columns* | *Data Type* | *Description* |
|  | category | String | Captain, First Officer, Senior Flight Attendant or Flight Attendant |
|  | crew | String | A string containing CrewCode, Position, Name and Legs |
|  | sortNo | Integer | Sorting number within category |

### fdpInfo

|  |  |  |  |
| --- | --- | --- | --- |
| *fdpInfo* | *Columns* | *Data Type* | *Description* |
|  | posHeader | String | Column header for the positions |
|  | fdp | String | FDP time (Check-in to Check-out) |
|  | split | String | Split Time (Off to On) |
|  | maxFDP | String | H:MM |
|  | duty | String | Duty time ( Check-in to Check-out) |
|  | reqRest | String | H:MM |
|  | nextCheckIn | String | H:MM |

### crewSchedule

|  |  |  |  |
| --- | --- | --- | --- |
| *crewSchedule* | *Columns* | *Data Type* | *Description* |
|  | fltId | String | Flight Id or Split or DH id |
|  | acReg | String | Aircraft registration or blank |
|  | datOp | Date | Date of operation for flights else start date in z |
|  | depTime | String | Dep time in z and LT |
|  | fromStn | String | Departure station |
|  | toStn | String | Arrival Station |
|  | arrTime | String | Arrival time in z and LT |
|  | blockTime | String | H:MM |
|  | Pax | String |  |
|  | icaoType | String | ??? |
|  | legNo | Integer |  |

### hotelInfo

|  |  |  |  |
| --- | --- | --- | --- |
| *hotelInfo* | *Columns* | *Data Type* | *Description* |
|  | hotelName | String | Hotel name and other information for the crewmember |
|  | pickupNote | String | Pickup note for the hotel |
|  | legNo | Integer | Reference to the legno in the crewSchedule |

### rosterActivity

For the methods that returns roster information we use the type rosterActivity in the web service calls. This type is described in this section.

The rosterActivity type includes all the values for the columns that you can select to see at CrewConnex (check-in page and roster page).

|  |  |  |  |
| --- | --- | --- | --- |
| *rosterActivity* | *Columns* | *Data Type* | *Description* |
|  | crewCode | String |  |
|  | employmentNum | String | Employment number |
|  | activitySeqno | Integer | Identification of the roster activity |
|  | trainingType | String | If this is a training activity that can be updated using the groundTraining methods then the type is returned: SIM: training on a simulator duty, GND: training on a ground duty. Else blank. |
|  | pairingSeqno | Integer | If this activity is part of a Pairing then the pairing identification is returned |
|  | kind | String | DH for deadheading else blank |
|  | refDate | Dateoffset | Reference date depending on the date mode Z, L (local to start of ftp) or B (crew base) selected for CrewConnex |
|  | dayRosterType | String | Customer specific column |
|  | pairingId | String | If the activity is part of a pairing then this tag will include the pairing id |
|  | revisionNum | String | If notifications have been released on this day the revisionNum will contain the revision number |
|  | dutyCode | String | Duty code this will be SN (supernumerary), DH (deadhead), R\* (recency training), C\* (course training), T\* (other training), I\* (Instructor), Position code if the position requirement is specialized or if the crewmember work low or high rank, else blank. |
|  | crewPosition | String | The required position that the crewmember is working in |
|  | specializedPosition | String | If the required position is specialized then this position is returned |
|  | checkInTime | Datetimeoffset | If the check-in time should be shown on CrewConnex then this field will contain the check-in date and time in UTC. |
|  | checkInTimeB | Datetimeoffset | The checkInTime in the crewmembers base time |
|  | checkInTimeL | Datetimeoffset | The checkInTime in the local time of the station where the checkIn takes place |
|  | checkOutTime | Datetimeoffset | If the check-out time should be shown on CrewConnex then this field will contain the check-out date and time in UTC. |
|  | checkOutTimeB | Datetimeoffset | The checkOutTime in the crewmbmers base time |
|  | checkOutTimeL | Datetimeoffset | The checkOutTime in the local time of the station where the checkout takes place |
|  | activityName | String | The name of the activity. For Flights it is the FltId, for activities it is the activity type code |
|  | activityRemark | String | The remark field from the activity. |
|  | activityTypeDesc | String | Description of the activity type if not a flight leg |
|  | stnFrom | String | Station code for the departure/start station |
|  | startTime | Datetimeoffset | Start time for the activity in UTC. For check-in it is the STD of the flight. If not for check-in it is the start time of the flight corresponding to the MVT status of the flight leg. |
|  | startTimeB | Datetimeoffset | The startTime in the crewmembers base time |
|  | startTimeL | Datetimeoffset | The startTime in the stnFrom time zone |
|  | stnTo | String | Station code for the arrival/end station |
|  | finishTime | Datetimeoffset | Finish time for the activity in UTC. For check-in it is the STA of the flight. If not for check-in it is the end time of the flight corresponding to the MVT status of the flight leg. |
|  | finishTimeB | Datetimeoffset | The finishTime in the crewmembers base time |
|  | finishTimeL | Datetimeoffset | The finishTime in the stnTo time zone |
|  | acType | String | The aircraft type for flight legs, else blank |
|  | absence | Boolean | True/False if the activity is defined as absence |
|  | turnTime | String | Duration in H:MM of the turn time to the next flight leg, if this activity is a flight leg else blank. |
|  | blockHours | String | Blockhours of the duty in H:MM shown on the last activity of the duty |
|  | dutyHours | String | DutyHours of the duty in H:MM shown on the last activity of the duty |
|  | fdpTime | String | The FDP time of the duty in H:MM shown on the last activity of the duty |
|  | maxFdpTime | String | The Max FDP time of the duty in H:MM shown on the last activity of the duty. If Max fdp time is used in the rules, blank if rolling 24-hours are used to determine the max fdp time. |
|  | restCompletedTime | String | The time when the required rest after the activity ends format HH:MM |
|  | assignedTakeoffLanding | String | Information of planned takeoff or landing |
|  | hotelName | String | If this is a hotel activity the this is the Hotel Name else the string is blank |
|  | reservation | String | For a deadhead activity this is the reservation code else the string is blank |
|  | acReg | String | If this is a flight leg then it is the Aircraft registration else it is blank |
|  | gate | String | Departure and arrival gate if they are defined Flight |
|  | crewMeal | String | Number of crew meals to be delivered on the flight for this crew. |
|  | bookedPax | String | Number of booked pax |
|  | otherActiveCrew | Array of otherCrew | An array of information for the other crewmembers assigned to the activity |
|  | otherDHCrew | Array of otherCrew | An array of information for the other crewmembers who are deadheading on the flight. |
|  | fltNotes | String | The flight notes from FlightOps |
|  | fltNote1 | String | The flight notes from FlightOps |
|  | fltNote2 | String | The flight notes from FlightOps |
|  | fltNote3 | String | The flight notes from FlightOps |
|  | fltNote4 | String | The flight notes from FlightOps |
|  | trainingInfo | String | If the activity is part of a training this will be like the tooltip in FlightCrew. E.g.:  Training in <Qualification name> by <Trainer code>  Trainer for trainee <Trainee code> in <Qualification name>  If the activity is part of a course this will be like the tooltip in FlightCrew. E.g.:  Course in <Course Name> by <Trainer code> (<progress>/<min req>) |
|  | dutyStart | Datetimeoffset | If the activity is part a duty, an FDP or a rest period then this is the start time of the duty/fdp/rest period in UTC. |
|  | dutyStartB | Datetimeoffset | The dutyStart in the crewmembers base time |
|  | dutyStartL | Datetimeoffset | The dutyStart in the timezone where the duty starts |
|  | dutyEnd | Datetimeoffset | If the activity is part a duty, an FDP or a rest period then this is the end time of the duty/fdp/rest period in UTC. |
|  | dutyEndB | Datetimeoffset | The dutyEnd in the crewmembers base time |
|  | dutyEndL | Datetimeoffset | The dutyEnd in the timezone where the duty ends |
|  | isRest | Boolean | True if the activity is part of a duty or an FDP else False. |

### otherCrew

|  |  |  |  |
| --- | --- | --- | --- |
| *otherCrew* | *Columns* | *Data Type* | *Description* |
|  | crewCode | String | Crew code of the crewmember |
|  | employmentNum | String | Employment number of the crewmember |
|  | fullName | String | Full name of the crewmember |
|  | isDeadhead | Boolean | True/False is this crewmember assigned to the flight as deadhead. Remaining fields are only filled out if False. |
|  | Position | String | Position name of the assigned crewmember |
|  | languageQualifications | Array of String | List of the language qualifications that this crewmember has |
|  | otherQualifications | Array of String | List of other qualifications that this crewmember has and that is selected to be shown on CrewConnex |
|  | previousDuty | String | Name of the previous duty that this crewmember was assigned to |
|  | seniorityNum | String | Seniority number |
|  | seating | String | The seating information |
|  | takeoffLanding | String | Planned takeoff and landing information |

### crewRequest

For the method that returns request information we use the type CrewRequest in the web service calls. This type is described in this section.

|  |  |  |  |
| --- | --- | --- | --- |
| *crewRequest* | *Columns* | *Data Type* | *Description* |
|  | crewCode | String |  |
|  | crewRequestSeqno | Integer | Identification of the request |
|  | priority | Integer | The priority of the request |
|  | comment | String | The Comment on the request |
|  | fromDate | Date | First date of the request |
|  | toDate | Date | Last date of the request |
|  | weekday | Integer | 1, 2, 3, 4, 5, 6, or 7 for Monday, Tuesday, …, or Sunday  0 for all |
|  | recurrence | String | O: Once  M: Multiple  A: All |
|  | requestType | String | F: free text request  T: time off request  P: pairing request  N: night stop request  A: absence request  V: vacation request |
|  | pairingId | String | Only for pairing requests |
|  | nightStopStn | String | Only for night stop requests |
|  | absenceType | String | Only for absences and vacation requests |
|  | timeOffStart | Time | Only for time off requests |
|  | timeOffEnd | Time | Only for time off requests |
|  | timeOffType | String | Only for time off requests |
|  | vacationBiddingType | Integer | Only if the request is a request for a vacation season |
|  | status | String | G: Granted  D: Denied  U: Unset |

### qualificationDef

qualificationDef data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *qualificationDef* | *Columns* | *Data Type* | *Description* |
|  | qualificationSeqno | Integer | Identification of the qualification |
|  | qualificationName | String | Name of qualification |
|  | description | String | Description of the qualification |
|  | type | String | The type can be no\_parameter, type\_aircraft, station, country, free\_text, route\_flightid, route\_citypair, or route\_countrypair. Normally only the first 3 is used. |
|  | validityStart | Dateoffset | The first date that the qualification is valid |
|  | validityEnd | Dateoffset | The last date that the qualification is valid |
|  | invisibleTraining | Boolean | It true the training will not bee seen in PDC FlightCrew. This could e.g. be for training received when landing (3 landings in 90 days or Familiarization at certain stations) |

### trainingEvent

trainingEvent data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *trainingEvent* | *Columns* | *Data Type* | *Description* |
|  | activitySeqno | Integer | Identification of the training activity |
|  | activityTypeSeqno | Integer | Identification of the activity type |
|  | activityType | String | Id of the activity Type |
|  | description | String | Description of the activity type |
|  | from | Datetimeoffset | Start time and date of the training activity in UTC |
|  | to | Datetimeoffset | End time and date of the training activity in UTC |
|  | trainers | Array of trainer | A list of the trainers who are assigned to the activity |
|  | students | Array of students | A list of the students who receive training on the activity |

### trainer

trainer data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *trainer* | *Columns* | *Data Type* | *Description* |
|  | crewCode | String | Identification of the crewmember |
|  | position | String | The name of the Position that the trainer is working in |

### student

student data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *student* | *Columns* | *Data Type* | *Description* |
|  | crewCode | String | Identification of the crewmember |
|  | position | String | The name of the Position that the student is working in |
|  | trainingSeqno | Integer | Identification of the training |
|  | qualificationSeqno | Integer | Identification of the Qualification that the crewmember receives training in |
|  | crewQualificationSeqno | Integer | Identification of the Crewmembers Qualification that the crewmember receives training in. In some cases there is no crew qualification related to the training. This could be if the training is part of a course or is extra training. In this case the label will not exist. |
|  | qualificationName | String | Name of qualification |
|  | qualificationParameter | String | Optional parameter for the crew qualification. |
|  | courseSeqno | Integer | Identification of a course if this training is part of a course. If the training is not part of a course then this label do not exist. |
|  | courseId | String | Name of the course. |
|  | status | String | P: Passed F: Failed H: Historic (old training that is not passed) <blank>: Unset |
|  | trainerCrewCode | String | CrewCode of the trainer |

### activityTypeDef

activityTypeDef data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *activityTypeDef* | *Columns* | *Data Type* | *Description* |
|  | activityTypeSeqno | Integer | Identification of the ActivityType |
|  | activityCategory | String | Name of Activity Category |
|  | activityType | String | Id of the activity type |
|  | description | String | Description of the activity type |
|  | validityStart | Dateoffset | The first date that the qualification is valid |
|  | validityEnd | Dateoffset | The last date that the qualification is valid |

### crewPositionDef

crewPositionDef data type:

|  |  |  |  |
| --- | --- | --- | --- |
| *crewPositionDef* | *Columns* | *Data Type* | *Description* |
|  | crewPositionSeqno | Integer | Identification of the Crew Position |
|  | crewCategory | String | Name of Crew Category |
|  | crewPosition | String | Id of the crew position |
|  | description | String | Description of the activity type |
|  | trainer | Boolean | Is trainer position |
|  | trainee | Boolean | Is trainee position |
|  | validityStart | Dateoffset | The first date that the qualification is valid |
|  | validityEnd | Dateoffset | The last date that the qualification is valid |

## Methods in PDCCrew2

The web service PDCCrew2 introduces a new authentication system that requires that you login using the login method before accesing the other methods. When calling login you receive an authentication string that you should use subsequently in the other methods. This authentication string expires after a set time, and you will need to re-login. The users that can login are the PDC FlightCrew users and the Crewmembers who has access to CrewConnex.

This section will describe the methods supported in PDCCrew2.

### version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| version | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | methods | Integer | Yes | DeadheadReservation Seqno |
| Response | result | methods |  | Array of method descriptions |

JSON example:

{"jsonrpc":"2.0","id":2,"method":"version","params":{"methods":[]}}

### login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **login** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | username | String | Yes | Username, CrewCode, or Employment Number |
|  | password | String | Yes | Password in plain text |
|  | crewmemberLogin | Boolean | Yes | No if the user is a PDC Crew user. |
| **Result** | authentication | String |  | Authentication string that will work for access to the below web service methods for a fixed time. Afterwards you should login again |
|  | limit | MinuteOffset |  | Expiry time of the authentication |
| **Error** | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

JSON example:

"method":"login","params":{"username":"User","password":"Userpass","crewmemberLogin":false}

"method":"login","params":{"username":"crewcode","password":"Crewpass","crewmemberLogin":true}

"method":"login","params":{"username":"employNum","password":"Crewpass","crewmemberLogin":true}

### getNotification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **getNotification** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user. Crewmembers may only ask for their own notifications. |
|  | crewCode | String | Yes | Crew code for crewmember |
| **Result** | Notifications | Array of notification |  | Return the list of un-acknowledged notifications for the crewmember |
| **Error** | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

JSON example:

"method":"getNotification","params":{"authentication":"1p470038037","crewCode":"min"}

### acknowledgeNotification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **acknowledgeNotification** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user. Crewmembers can only confirm their own notifications. |
|  | crewCode | String | Yes | Crew code for crewmember that wants to acknowledge a notification |
|  | notificationSeqno | Integer | Yes | Identity seqno of the notification |
| **Result** | updateOk | Boolean |  | True/False |
|  | Status | String |  | Explanation if updateOk = false |
| **Error** | Code | Integer |  | Not included if no error |
|  | Message | String |  | Not included if no error |

JSON example:

"method":"acknowledgeNotification","params":{"authentication":"1p439883914","crewCode":"min","notificationSeqno":10867955}

### getNextCheckIn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **getNextCheckIn** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user. Crewmember logins can only access their own. |
|  | crewCode | String | Yes | Crew code for crewmember that wants to check in |
| **Result** | checkInStatus | String |  | The status will be one of the following strings:  "Check-in is not yet possible" "Checked in at %s"  "Can check in"  "Too late for check-in"  "Check-in not required"  "Roster needs to be released, contact OPS" "Must acknowledge notifications first"  "Check-in is not possible" |
|  | checkInAvailable | Boolean |  | True/False – If the check-in will be available in CrewConnex |
|  | activityList | Array of checkInRosterActivities |  | An array of the rosterActivities assigned to the crewmember included in the first fdp that the crewmember can check in for, and all following fdp until the crewmember is back on the base. |
| **Error** | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

JSON example:

"method":"getNextCheckIn","params":{"authentication":"2p3118419325","crewCode":"adm"}

### doCheckIn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **doCheckIn** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user. Crewmembers can only check themselves in. |
|  | crewCode | String | Yes | Crew code for crewmember that wants to check in |
|  | checkInTime | MinuteOffset | Yes | The check-in time in UTC that you want to register |
| **Result** | updateOk | Boolean |  | True/False |
|  | status | String |  | The status will be one of the following strings:  "Checked in at %s"  "Too late for check-in"  "Check-in not required"  "Roster needs to be released, contact OPS"  If updateOk is True then the status is "Checked in at %s" |
| **Error** | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

JSON example:

"method":"doCheckIn","params":{"authentication":"2p3118419325","crewCode":"hge","checkInTime":45120000}

### getQualification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **getQualification** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user. Crewmembers may only see their own qualifications. |
|  | crewCode | String | Yes | Crew code for crewmember |
| **Result** | qualifications | Array of crewQualification |  | Return the list of qualifications that this crewmember has |
| **Error** | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

JSON example:

"method":"getQualification","params":{"authentication":"1p439883914","crewCode":"min"}

### updateQualification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **updateQualification** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user  PDCCrew users only |
|  | crewCode | String | Yes | Crew code for crewmember |
|  | crewQualificationSeqno | Integer | No | Identification of the crew qualification.  If crewQualificationSeqno is not present the qualifications is found using the qualificationName, parameter and validityStart/ validityEnd.  If no matching qualification is found, a new qualification is created for the crewmewmber |
|  | qualificationName | String | No | Only for new crew qualifications |
|  | parameter | String | No | Only for new crew qualifications |
|  | position | String | No | Only for new crew qualifications |
|  | Grade | String | No | Only for new crew qualifications |
|  | validityStart | Dateoffset | No | The validity Start will be set to this value  Mandatory for new crew qualifiations |
|  | validityEnd | Dateoffset | No | The validity End will be set to this value Mandatory for new crew qualifiations |
|  | lastTrainingDate | Dateoffset | No | The last training date will be set to this value |
|  | lastTraining | String | No | The last training will be set to this value |
|  | expiryDate | Dateoffset | No | The expiry date will be set to this value |
|  | passportNumber | String | No | For the passport qualification only |
|  | passportCountry | String | No | For the passport qualification only |
|  | passportNumber2 | String | No | For the 2nd passport qualification only |
|  | passportCountry2 | String | No | For the 2nd passport qualification only |
|  | licenseNumber | String | No | For the license qualification only |
|  | licenseCountry | String | No | For the license qualification only |
|  | licenseType | Sting | No | For the license qualification only |
| **Result** | newCrewQualificationSeqno | Integer |  | If a new crew qualification is created then the identification for that new crew qualification is returned |
|  | Alerts | Array of crewAlert |  | Return the list of alerts that change raises |
| **Error** | Code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

This can be used to either update an existing qualification, or to create a new qualification for a given crewmember. 0 is returned for the seqno if the operation was an update; otherwise, the newly created crewQualification’s seqno is returned.

### createCrewmember

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **createCrewmember** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user  PDCCrew users only |
|  | crewCode | String | Yes | The crewCode for the new crewmember |
|  | empNumber | String | No | Employment number |
|  | empStart | DateOffset | Yes | Employment start date |
|  | empEnd | DateOffset | No | Employment end date |
|  | firstName | String | No |  |
|  | middleName | String | No |  |
|  | lastName | String | No |  |
|  | gender | String | Yes | Male or Female |
|  | dateOfBirth | DateOffset | Yes |  |
|  | crewCategory | String | Yes | Name of the Crew Category |
|  | nationality | String | Yes | Country code |
|  | countryOfBirth | String | Yes | Country code |
|  | cityOfBirth | String | No | City of Birth |
|  | provinceOfBirth | String | No | Province of Birth |
|  | passportNumber | String | No |  |
|  | passportCountry | String | No | Passport issue country |
|  | passportIssueDate | DateOffset | No | Passport issue date |
|  | passportExpiryDate | DateOffset | No | Passport expiry date |
|  | addrStreet | String | No | Address street |
|  | addrCity | String | No | Address City |
|  | addrState | String | No | Address State |
|  | addrZipCode | String | No | Address Zip Code |
|  | addrCountry | String | No | Address Country code |
|  | phoneMobile | String | No | Mobile phone number |
|  | phoneHome | String | No | Home phone number |
|  | Email | String | No | Email address |
|  | Base | String | Yes | Name of Base |
|  | tempBase | String | No | Name of Temp base |
|  | jobTitle | String | Yes | Job title |
|  | Rank | String | Yes | Name of Rank |
|  | ruleSet | String | No | Name of rule set setting |
|  | empKind | String | Yes | Permanent, Rehire, Temporary, Lease or Freelance |
|  | workPct | Integer | No | Work percentage |
| **Result** | crewSeqno | Integer |  | The seqno of the new crewmember |
| **Error** | Code | Integer |  | Not included if no error |
|  | Message | String |  | Not included if no error |

Json Example:

"method":"createCrewmember","params":{"authentication":"1p909747255","base":"AAR","countryOfBirth":"US","crewCategory":"FD","crewCode":"RJEJP","dateOfBirth":568944000000,"empKind":"Freelance","empStart":1480896000000,"gender":"male","jobTitle":"Captain","nationality":"Denmark","rank":"FC"}

### updateCrewmember

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **updateCrewmember** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user  PDCCrew users only |
|  | crewCode | String | Yes | The crewCode for the new crewmember This is the identification, it cannot be changed |
|  | empNumber | String | No | Employment number |
|  | empStart | DateOffset | No | Employment start date |
|  | empEnd | DateOffset | No | Employment end date |
|  | firstName | String | No |  |
|  | middleName | String | No |  |
|  | lastName | String | No |  |
|  | gender | String | No | Male or Female |
|  | dateOfBirth | DateOffset | No |  |
|  | crewCategory | String | No | Name of the Crew Category |
|  | nationality | String | No | Country code |
|  | countryOfBirth | String | No | Country code |
|  | cityOfBirth | String | No | City of Birth |
|  | provinceOfBirth | String | No | Province of Birth |
|  | passportNumber | String | No |  |
|  | passportCountry | String | No | Passport issue country |
|  | passportIssueDate | DateOffset | No | Passport issue date |
|  | passportExpiryDate | DateOffset | No | Passport expiry date |
|  | addrStreet | String | No | Address street |
|  | addrCity | String | No | Address City |
|  | addrState | String | No | Address State |
|  | addrZipCode | String | No | Address Zip Code |
|  | addrCountry | String | No | Address Country code |
|  | phoneMobile | String | No | Mobile phone number |
|  | phoneHome | String | No | Home phone number |
|  | Email | String | No | Email address |
| **Result** | updateOk | Boolean |  | True if ok, otherwise not included |
| **Error** | Code | Integer |  | Not included if no error |
|  | Message | String |  | Not included if no error |

### updateEmployment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **updateEmployment** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user  PDCCrew users only |
|  | crewCode | String | Yes | The crewCode for the new crewmember This is the identification, it cannot be changed |
|  | fromDate | DateOffset | Yes | Change Employment details from |
|  | Base | String | Yes | Name of base |
|  | tempBase | String | No | Name of Temp base |
|  | jobTitle | String | Yes | Job title |
|  | Rank | String | Yes | Name of Rank |
|  | ruleSet | String | No | Name of rule set setting |
|  | empKind | String | Yes | Permanent, Rehire, Temporary, Lease or Freelance |
|  | workPct | Integer | Yes | Work percentage |
|  | grounded | Boolean | Yes | Grounded |
| **Result** | updateOk | Boolean |  | True if ok, otherwise not included |
| **Error** | Code | Integer |  | Not included if no error |
|  | Message | String |  | Not included if no error |

### getTripSheetInformation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| getTripSheetInformation | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | fltId | String | Yes | The flight id |
|  | datOp | Date | Yes | Date of Operations |
|  | legNo | Int | Yes | Leg number |
|  | crewCode | String | Yes | CrewCode |
| Response | crewInformation | Array of crewInfo |  | JSON string containing the Captain/First office section, Flight Attendants section. |
|  | fdpInformation | Array of fdpInfo |  | JSON string containing the Flight Duty Period section |
|  | crewSchedule | Array of crewSchedule |  | JSON string containing the Schedule for the crewmember |
|  | hotelInformation | Array of hotelInfo |  | JSON string containing the Hotel information |
|  | pickupInformation | Array of pickupInfo |  | JSON string containing the transportation information |
|  | flightLegList | Array of FlightLegKey |  | The flight legs that should be included in the report. |

### getRoster

This method will return the activities, pairings and flights that are assigned to a crewmember’s roster in a specified period. Together with the roster you will get a date telling you what period is published to the crewmembers. If there are unpublished changes inside the roster control period, the method will return the roster that is published to the crewmember (same as the roster on CrewConnex).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| getRoster | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| Request | authentication | String | Yes | Authentication of the user |
|  | crewCode | String | Cond | Crew code for crewmember |
|  | employmentNum | String | Cond | Employment number for the crewmember. The crewCode or the employmentNo must be filled out. |
|  | fromDate | Dateoffset | Yes |  |
|  | toDate | Dateoffset | No |  |
| Result | roster | Array of rosterActivity |  | Return the list activities that are assigned to the crewmember in the selected period. |
|  | publishTo | Dateoffset |  | The data that the crewmembers rosters are published until |
| Error | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

Json example: {"jsonrpc":"2.0","id":21,"method":"getRoster","params":{"crewCode":"XXX213","fromDate":1561990000000,"toDate":1562140000000}}

### getCrewCourse

This method will return the crew courses including status and training of the crewmember

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| getCrewCourse | | | | |
|  | *Label* | *Data Type* | *Mandatory* | *Description* |
| Request | authentication | String | Yes | Authentication of the user |
|  | crewCode | String | Cond | Crew code for crewmember |
|  | employmentNum | String | Cond | Employment number for the crewmember. The crewCode or the employmentNo must be filled out. |
|  | fromDate | Date | Yes | From date |
|  | toDate | Date | Yes | To date |
| Result | courses | Array of crewCourse |  | List of the crew courses intersecting the period. The training list is only training within the period. |
| Error | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

Json example: {"jsonrpc":"2.0","id":15,"method":"getCrewCourses","params":{"authentication":"hdlksdau38u924ujlsa9","crewCode":"ABC","fromDate":1565654400, "toDate":1565654400}}

### getCrewRequest

This method will return the requests that the crewmember has

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| getCrewRequest | | | | |
|  | *Label* | *Data Type* | *Mandatory* | *Description* |
| Request | authentication | String | Yes | Authentication of the user |
|  | crewCode | String | Cond | Crew code for crewmember |
|  | employmentNum | String | Cond | Employment number for the crewmember. The crewCode or the employmentNo must be filled out. |
|  | vacationSeasonSeqno | Integer | No | If present the method will only return the requests that belongs to this vacation season. |
| Result | requests | Array of crewRequest |  | The list of requests |
| Error | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

Json example: {"jsonrpc":"2.0","id":15,"method":"getCrewRequest","params":{"authentication":"hdlksdau38u924ujlsa9","crewCode":"ABC"}}

### deleteCrewRequest

This method will delete a request for a crewmember

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| deleteCrewRequest | | | | |
|  | *Label* | *Data Type* | *Mandatory* | *Description* |
| Request | authentication | String | Yes | Authentication of the user |
|  | crewCode | String | Cond | Crew code for crewmember |
|  | employmentNum | String | Cond | Employment number for the crewmember. The crewCode or the employmentNo must be filled out. |
|  | crewRequestSeqno | Integer | Yes | The Identification of the request |
| Result | result | String |  | Ok |
| Error | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

Json example: {"jsonrpc":"2.0","id":15,"method":"deleteCrewRequest","params":{"authentication":"hdlksdau38u924ujlsa9","crewCode":"ABC","requestSeqno":1234567}}

### createCrewRequest

This method will create a new request for a crewmember

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| createCrewRequest | | | | |
|  | *Label* | *Data Type* | *Mandatory* | *Description* |
| Request | Authentication | String | Yes | Authentication of the user |
|  | crewCode | String | Cond | Crew code for crewmember |
|  | employmentNum | String | Cond | Employment number for the crewmember. The crewCode or the employmentNo must be filled out. |
|  | Priority | Integer | Yes |  |
|  | comment | String | Yes |  |
|  | fromDate | Date | Yes | First date of the request |
|  | toDate | Date | No | Only needed if the request is for a period |
|  | weekday | Integer | No | 1, 2, 3, 4, 5, 6, or 7 for Monday, Tuesday, …, or Sunday |
|  | recurrence | String | Yes | O: Once  M: Multiple  A: All |
|  | requestType | String | Yes | F: free text request  T: time off request  P: pairing request  N: night stop request  A: absence request  V: vacation request |
|  | pairingId | String | Cond | Mandatory for pairing requests |
|  | nightStopStn | String | Cond | Mandatory fir night stop requests |
|  | absenceType | String | Cond | Mandatory for absences and vacation requests |
|  | timeOffStart | Time | Cond | Mandatory for time off requests |
|  | timeOffEnd | Time | Cond | Mandatory for time off requests |
|  | timeOffType | String | Cond | Mandatory for time off requests |
|  | vacationBiddingType | Integer | Cond | Mandatory if the request is a request for a vacation season |
| Result | newRequest | crewRequest |  | Returns the created request |
| Error | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

Request (TCreate a request for crewmember ABC for absence OFF on the 13Aug19):

{"jsonrpc":"2.0","id":15,"method":"createCrewRequest","params":{"authentication":"hdlksdau38u924ujlsa9","crewCode":"ABC","priority":1,"comment":"This is a new request","fromDate":156565440000, "recurrence":"O","requestType":"A","absenceType":"OFF"}}

### getAllQualificationDef

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **getAllQualificationDef** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user. |
| **Result** | qualifications | Array of qualificationDef |  | Return the list of the definitions of all qualification |
| **Error** | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

JSON example:

"method":"getAllQualificationDef","params":{"authentication":"1p439883914"}

### getTrainingEvents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **getTrainingEvents** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user. |
|  | from | Dateoffset | Yes |  |
|  | to | Dateoffset | Yes |  |
| **Result** | trainingEvents | Array of trainingEvent |  | Return the list training events within the period from – to. |
| **Error** | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

JSON example:

"method":"getTrainingEvents","params":{"authentication":"1p439883914","from":1620777600000,"to":1622419200000}

### getAllActivityTypeDef

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **getAllActivityTypeDef** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user. |
| **Result** | activityTypes | Array of activityTypeDef |  | Return the list of the definitions of all activity Types |
| **Error** | Code | Integer |  | Not included if no error |
|  | Message | String |  | Not included if no error |

JSON example:

"method":"getAllActivityTypeDef","params":{"authentication":"1p439883914"}

### getAllCrewPositionDef

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **getAllCrewPositionDef** | | | | |
|  | *Columns* | *Data Type* | *Mandatory* | *Description* |
| **Request** | authentication | String | Yes | Authentication of the user. |
| **Result** | crewPositions | Array of crewPostionDef |  | Return the list of the definitions of all activity Types |
| **Error** | Code | Integer |  | Not included if no error |
|  | Message | String |  | Not included if no error |

JSON example:

"method":"getAllCrewPositionDef","params":{"authentication":"1p439883914"}

## Methods inherited from PDCCrew

Many methods from PDCCrew can be called from PDCCrew2. They will require an additional authentication, like all PDCCrew2 methods.

## HTTP Error Codes

If a webservice call fails, an error message is sent back via JSON. The error message is made up of an HTTP error code and a descriptive message.

This section will describe the error codes used.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Error | code | Integer |  | Not included if no error |
|  | message | String |  | Not included if no error |

### Client Errors (4XX)

    Bad Request : 400.  
    Unauthorized : 401.   
    Forbidden : 403.  
    Not Found : 404.  
    Method Not Allowed : 405.  
    Not Acceptable : 406.  
    Proxy Authentication Required : 407.  
    Request Timeout : 408.  
    Conflict : 409.  
    Gone : 410.  
    Length Required : 411.  
    Precondition Failed : 412.  
    Payload Too Large : 413.  
    Uri Too Long : 414.  
    Unsupported Media Type : 415.  
    Range Not Satisfiable : 416.  
    Expectation Failed : 417.   
    Misdirected : 421.  
    Unprocessable Entity : 422.  
    Locked : 423.  
    Failed Dependency : 424.  
    Upgrade Required : 426.  
    Precondition Required : 428.  
    Too Many Requests : 429.  
    Request Header Fields Too Large : 431.  
    Unavailable For Legal Reasons : 451.

### Server Errors (5XX)

    Internal Server Error : 500.  
    Not Implemented : 501.  
    Bad Gateway : 502.  
    Service Unavailable : 503.  
    Gateway Timeout : 504.  
    HTTP Version Not Supported : 505.  
    Variant Also Negotiates : 506.  
    Insufficient Storage : 507.  
    Loop Detected : 508.  
    Not Extended : 510.  
    Network Authentication Required : 511.