pears.NetClientCreateBooking

Generated schema reference. Regenerate this page from the SQL unload; keep hand-maintained business notes in the narrative namespace.

Original SQL

```
CREATE PROCEDURE "pears"."NetClientCreateBooking"( IN "pWebUserID"
CHAR(20),IN "pPersonID" CHAR(20),IN "pLineID" CHAR(20),IN "pIsShift" CHAR(1)
)
RESULT( "pResult" CHAR(250) )
// IQXWeb
BEGIN
  DECLARE "sid" CHAR(20);
  DECLARE "swebusername" CHAR(50);
  DECLARE "availability" CHAR(20);
  DECLARE "spopup" CHAR(100);
  DECLARE LOCAL TEMPORARY TABLE "IDs"(
    "ID" CHAR(20) NULL,
    ) NOT transactional;
  INSERT INTO "IDs"( "ID" )
    SELECT DISTINCT "e"."companyid" FROM "employment" AS "e" KEY JOIN
"iqxnetuserlink" AS "i" WHERE "i"."iqxnetuserid" = "pWebUserID";
  IF "pIsShift" = '0' THEN
    IF NOT "pLineID" = any(SELECT "v"."vacancyid" FROM "vacancy" AS "v" KEY
JOIN "employment" AS "e" JOIN "IDs" ON "e"."CompanyID" = "IDs"."ID") THEN
      SELECT '99:~Permission denied';
      RETURN
    END IF
  ELSE IF NOT "pLineID" = any(SELECT "p"."tempshiftplanid" FROM
"tempshiftplan" AS "p" KEY JOIN "vacancy" AS "v" KEY JOIN "employment" AS
"e" JOIN "IDs" ON "e"."CompanyID" = "IDs"."ID") THEN
      SELECT '99:~Permission denied';
      RETURN
    END IF END IF;
  IF "IQXNetHasPermission"("pWebUserID",'FILLJOBS') = 0 THEN
    SELECT '99:~Permission denied';
    RETURN
  END IF;
  SET "sid" = "uniquekey"("ppersonid");
  SET "swebusername" = (SELECT "name" FROM "iqxnetuser" WHERE "iqxnetuserid"
= "pwebuserid");
```

```
  IF "pIsShift" = '0' THEN
    IF(SELECT
"EmployeeAcceptable"("ppersonid","e"."companyid",1,"v"."vacancyid") FROM
"vacancy" AS "v" KEY JOIN "employment" AS "e" WHERE "v"."vacancyid" =
"pLineID") IS NOT NULL THEN
      SELECT '110:~Employee is not suitable for this vacancy';
      RETURN
    END IF;
    INSERT INTO "employment"(
"temp","employmentid","personid","companyid","startdate","leavedate","positi
on","department" )
      SELECT
"v"."temp","sid","ppersonid","e"."companyid","v"."startdate","v"."finishdate
","v"."position","v"."clientdepartment"
        FROM "vacancy" AS "v" KEY JOIN "employment" AS "e" WHERE
"v"."vacancyid" = "pLineID";
    INSERT INTO "placement"(
"placementid","placedate","employmentid","vacancyid","departmentid","staffid
","temp","daysperweek","salary",
"TheirRef","ContractRef","Currency","WorkMonday","WorkTuesday","WorkWednesda
y","WorkThursday","WorkFriday","WorkSaturday","WorkSunday",
      "WorkNormalHours","WorkStartTime" )
      SELECT "sid",CURRENT
DATE,"sid","pLineID","v"."departmentid","userstaffid","v"."temp",5,"v"."sala
ry","v"."theirref","v"."contractref","v"."currency",
"v"."WorkMonday","v"."WorkTuesday","v"."WorkWednesday","v"."WorkThursday","v
"."WorkFriday","v"."WorkSaturday","v"."WorkSunday",
        "v"."WorkNormalHours","v"."WorkStartTime"
        FROM "vacancy" AS "v" WHERE "v"."vacancyid" = "pLineID";
    CALL
"IQXNetPopup"("pWebUserID",NULL,NULL,"pLineID","sid",NULL,'CLIENTBOOKING','C
lient made placement',NULL)
  ELSE
    IF(SELECT
"EmployeeAcceptable"("ppersonid","e"."companyid",1,"v"."vacancyid") FROM
"tempshiftplan" AS "p" KEY JOIN "vacancy" AS "v" KEY JOIN "employment" AS
"e" WHERE "p"."tempshiftplanid" = "pLineID") IS NOT NULL THEN
      SELECT '110:~Employee is not suitable for this shift';
      RETURN
    END IF;
    /*    if (select
isnull(TempShiftAllowed(p.tempshiftplanid,ppersonid,v.vacancyid),'') from
tempshiftplan as p key join vacancy v key join employment e where
p.tempshiftplanid = pLineID) != '' then
select '111:~Employee is not suitable for this shift';
return;
end if;
// NetClientRequirementVerifiedCandidates does NOT call TempShiftAllowed so
```

```
this shouldn't either by common sense. May need review.
*/
    SET "availability" = (SELECT
"personavailableforplan"("ppersonid","p"."ShiftDate","p"."TimeFrom","p"."Tim
eTo","p"."RecoveryHours","p"."minutes",0,"v"."departmentid","p"."essentialsk
ill","p"."essentialskillgradeid","p"."essentialskillchoicelist",NULL)
        FROM "tempshiftplan" AS "p" KEY JOIN "vacancy" AS "v" WHERE
"p"."tempshiftplanid" = "pLineID");
    IF "isnull"("availability",'A') NOT LIKE '[AM]%' THEN
      SELECT '112:~Candidate not suitable or not available';
      RETURN
    END IF;
    IF EXISTS(SELECT * FROM "tempshift" WHERE "state" IN( 'P','B' ) AND
"tempshiftplanid" = "pLineID") THEN
      SELECT '113:~Shift already filled';
      RETURN
    END IF;
    INSERT INTO "tempshift"(
"TempShiftID","VacancyID","PersonID","ShiftDate","TimeFrom","TimeTo","State"
,"TempShiftPlanID",
"BreakMinutes","EssentialSkillGradeID","ReferenceCode","ClientConfirmed","Te
mpConfirmed","StaffID",
      "ConfirmedWith","TempShiftTypeID","AnalysisCode","RecoveryHours" )
      SELECT
"sid","v"."VacancyID","pPersonID","p"."ShiftDate","p"."TimeFrom","p"."TimeTo
","P","pLineID",
"p"."BreakMinutes","p"."EssentialSkillGradeID","p"."ReferenceCode",1,0,"user
StaffID",
"swebusername","p"."TempShiftTypeID","p"."AnalysisCode","p"."RecoveryHours"
        FROM "tempshiftplan" AS "p" KEY JOIN "vacancy" AS "v" KEY JOIN
"employment" AS "e" WHERE "p"."tempshiftplanid" = "pLineID";
    SET "spopup" = (SELECT "dateformat"("p"."shiftdate",'Ddd dd/mm/yyyy')+'
'+"dateformat"("p"."timefrom",'hh:nn')+' with '+"person"."name" FROM
"tempshiftplan" AS "p","person"
        WHERE "p"."tempshiftplanid" = "pLineID" AND "person"."personid" =
"ppersonid");
    CALL
"IQXNetPopup"("pWebUserID",NULL,NULL,"pLineID",NULL,NULL,'CLIENTBOOKING','Cl
ient filled shift',"spopup")
  END IF;
  SELECT '0:~Success'
END /* DOC

2019-01-18 PC IW-788 allow client to create booking, based on NetVacancyFill
2019-02-18 PC/VHS Modify to prevent multiple bookings
2019-03-07 PC IW-884 EmployeeAcceptable and TempshiftAllowed moved ahead of
main functions
2019-06-19 MHS tidied up and fixed tests
```

```
*/
/* TEST
call nettestsetup('');
delete from withholds where personid='test';
delete from tempshift where personid='test' and tempshiftid<>'test';
delete from tempshift where tempshiftplanid = 'TEST1';
def test select * from NetClientCreateBooking('test.client','test',
'test',0);
do test;
expect 0*;
insert into withholds (withholdsid,personid,vacancyid) values
(uniquekey(''),'test','test');
do test;
expect 110*;
update employment set startdate=current date - 2, leavedate=current date - 1
where personid='TEST';
insert into tempshiftplan
(tempshiftplanid,vacancyid,description,shiftdate,timefrom,timeto) on
existing update defaults off values ('TEST','TEST','t1',current
date,'09:00','17:00');
insert into tempshift
(tempshiftid,vacancyid,personid,shiftdate,timefrom,timeto,state,tempshiftpla
nid) on existing update defaults off values ('TEST','TEST','TEST',current
date,'09:00','17:00','B','TEST');
insert into tempshiftplan
(tempshiftplanid,vacancyid,description,shiftdate,timefrom,timeto) on
existing update defaults off values ('TEST1','TEST','t2',current
date,'09:00','17:00');
def test select * from NetClientCreateBooking('test.client','test',
'test1',1);
do test;
expect 110*;
delete from withholds where personid='test';
do test;
expect 112*;
update tempshiftplan set shiftdate=current date + 1 where
tempshiftplanid='TEST1';
do test;
expect 0*;
update tempshift set personid=null where tempshiftplanid = 'TEST1';
do test;
expect 113*;
*/
GO

COMMENT TO PRESERVE FORMAT ON PROCEDURE "pears"."NetClientCreateBooking" IS
{CREATE PROCEDURE pears.NetClientCreateBooking( IN pWebUserID CHAR(20),IN
pPersonID CHAR(20), IN pLineID CHAR(20), IN pIsShift CHAR(1))
```

```
RESULT( pResult CHAR(250) )
// IQXWeb
BEGIN
  DECLARE sid CHAR(20);
  DECLARE swebusername CHAR(50);
  DECLARE availability CHAR(20);
  DECLARE spopup CHAR(100);
  DECLARE LOCAL TEMPORARY TABLE IDs(
    ID CHAR(20) NULL,
    ) NOT transactional;
  INSERT INTO IDs( ID )
    SELECT DISTINCT e.companyid FROM employment AS e KEY JOIN iqxnetuserlink
AS i WHERE i.iqxnetuserid = pWebUserID;
    IF pIsShift='0' THEN
        IF NOT pLineID = any(SELECT v.vacancyid FROM vacancy AS v KEY JOIN
employment AS e JOIN IDs ON e.CompanyID = IDs.ID) THEN
            SELECT '99:~Permission denied';
            RETURN
        END IF;
    ELSE
        IF NOT pLineID = any(SELECT p.tempshiftplanid FROM tempshiftplan AS
p KEY JOIN vacancy AS v KEY JOIN employment AS e JOIN IDs ON e.CompanyID =
IDs.ID) THEN
            SELECT '99:~Permission denied';
            RETURN
        END IF;
    END IF;
  IF IQXNetHasPermission(pWebUserID,'FILLJOBS') = 0 THEN
    SELECT '99:~Permission denied';
    RETURN
  END IF;
  SET sid = uniquekey(ppersonid);
  SET swebusername = (SELECT name FROM iqxnetuser WHERE iqxnetuserid =
pwebuserid);

  IF pIsShift = '0' THEN
    IF (SELECT EmployeeAcceptable(ppersonid,e.companyid,1,v.vacancyid) FROM
vacancy AS v KEY JOIN employment AS e WHERE v.vacancyid = pLineID) IS NOT
NULL THEN
        SELECT '110:~Employee is not suitable for this vacancy';
        RETURN;
    END IF;
    INSERT INTO employment(
temp,employmentid,personid,companyid,startdate,leavedate,POSITION,department
)
      SELECT
v.temp,sid,ppersonid,e.companyid,v.startdate,v.finishdate,v.position,v.clien
tdepartment
```

```
        FROM vacancy AS v KEY JOIN employment AS e WHERE v.vacancyid =
pLineID;
    INSERT INTO placement(
placementid,placedate,employmentid,vacancyid,departmentid,staffid,temp,daysp
erweek,salary,
TheirRef,ContractRef,Currency,WorkMonday,WorkTuesday,WorkWednesday,WorkThurs
day,WorkFriday,WorkSaturday,WorkSunday,
      WorkNormalHours,WorkStartTime )
      SELECT sid,CURRENT
DATE,sid,pLineID,v.departmentid,userstaffid,v.temp,5,v.salary,v.theirref,v.c
ontractref,v.currency,
v.WorkMonday,v.WorkTuesday,v.WorkWednesday,v.WorkThursday,v.WorkFriday,v.Wor
kSaturday,v.WorkSunday,
        v.WorkNormalHours,v.WorkStartTime
        FROM vacancy AS v WHERE v.vacancyid = pLineID;
    CALL
IQXNetPopup(pWebUserID,NULL,NULL,pLineID,sid,NULL,'CLIENTBOOKING','Client
made placement',NULL);
  ELSE
    IF (SELECT EmployeeAcceptable(ppersonid,e.companyid,1,v.vacancyid) FROM
tempshiftplan AS p KEY JOIN vacancy v KEY JOIN employment e WHERE
p.tempshiftplanid = pLineID) IS NOT NULL THEN
        SELECT '110:~Employee is not suitable for this shift';
        RETURN;
    END IF;
/*    if (select
isnull(TempShiftAllowed(p.tempshiftplanid,ppersonid,v.vacancyid),'') from
tempshiftplan as p key join vacancy v key join employment e where
p.tempshiftplanid = pLineID) != '' then
        select '111:~Employee is not suitable for this shift';
        return;
    end if;
   // NetClientRequirementVerifiedCandidates does NOT call TempShiftAllowed
so this shouldn't either by common sense. May need review.
    */

    SET availability = (SELECT
personavailableforplan(ppersonid,p.ShiftDate,p.TimeFrom,p.TimeTo,p.RecoveryH
ours,p.minutes,0,v.departmentid,p.essentialskill,p.essentialskillgradeid,p.e
ssentialskillchoicelist,NULL)
        FROM tempshiftplan AS p KEY JOIN vacancy v WHERE p.tempshiftplanid =
pLineID);
    IF NOT isnull(availability,'A') LIKE '[AM]%' THEN
      SELECT '112:~Candidate not suitable or not available';
      RETURN
    END IF;
    IF EXISTS(SELECT * FROM tempshift WHERE state IN ('P','B') AND
tempshiftplanid = pLineID) THEN
```

```
      SELECT '113:~Shift already filled';
      RETURN
    END IF;
    INSERT INTO tempshift(
TempShiftID,VacancyID,PersonID,ShiftDate,TimeFrom,TimeTo,State,TempShiftPlan
ID,
BreakMinutes,EssentialSkillGradeID,ReferenceCode,ClientConfirmed,TempConfirm
ed,StaffID,
      ConfirmedWith,TempShiftTypeID,AnalysisCode,RecoveryHours )
      SELECT
sid,v.VacancyID,pPersonID,p.ShiftDate,p.TimeFrom,p.TimeTo,'P',pLineID,
p.BreakMinutes,p.EssentialSkillGradeID,p.ReferenceCode,1,0,userStaffID,
        swebusername,p.TempShiftTypeID,p.AnalysisCode,p.RecoveryHours
        FROM tempshiftplan AS p KEY JOIN vacancy v KEY JOIN employment e
WHERE p.tempshiftplanid = pLineID;
    SET spopup = (SELECT dateformat(p.shiftdate,'Ddd dd/mm/yyyy')+'
'+dateformat(p.timefrom,'hh:nn')+' with '+person.name FROM tempshiftplan AS
p,person
        WHERE p.tempshiftplanid = pLineID AND person.personid = ppersonid);
    CALL
IQXNetPopup(pWebUserID,NULL,NULL,pLineID,NULL,NULL,'CLIENTBOOKING','Client
filled shift',spopup);
  END IF;
  SELECT '0:~Success'
END

/* DOC

2019-01-18 PC IW-788 allow client to create booking, based on NetVacancyFill
2019-02-18 PC/VHS Modify to prevent multiple bookings
2019-03-07 PC IW-884 EmployeeAcceptable and TempshiftAllowed moved ahead of
main functions
2019-06-19 MHS tidied up and fixed tests
*/

/* TEST
call nettestsetup('');
delete from withholds where personid='test';
delete from tempshift where personid='test' and tempshiftid<>'test';
delete from tempshift where tempshiftplanid = 'TEST1';
def test select * from NetClientCreateBooking('test.client','test',
'test',0);
do test;
expect 0*;
insert into withholds (withholdsid,personid,vacancyid) values
(uniquekey(''),'test','test');
do test;
expect 110*;
```

```
update employment set startdate=current date - 2, leavedate=current date - 1
where personid='TEST';
insert into tempshiftplan
(tempshiftplanid,vacancyid,description,shiftdate,timefrom,timeto) on
existing update defaults off values ('TEST','TEST','t1',current
date,'09:00','17:00');
insert into tempshift
(tempshiftid,vacancyid,personid,shiftdate,timefrom,timeto,state,tempshiftpla
nid) on existing update defaults off values ('TEST','TEST','TEST',current
date,'09:00','17:00','B','TEST');
insert into tempshiftplan
(tempshiftplanid,vacancyid,description,shiftdate,timefrom,timeto) on
existing update defaults off values ('TEST1','TEST','t2',current
date,'09:00','17:00');
def test select * from NetClientCreateBooking('test.client','test',
'test1',1);
do test;
expect 110*;
delete from withholds where personid='test';
do test;
expect 112*;
update tempshiftplan set shiftdate=current date + 1 where
tempshiftplanid='TEST1';
do test;
expect 0*;
update tempshift set personid=null where tempshiftplanid = 'TEST1';
do test;
expect 113*;
*/
}
```

From:
https://iqxusers.co.uk/iqxhelp/ - **iqx**

Permanent link:
https://iqxusers.co.uk/iqxhelp/doku.php?id=database:procedures:pears_netclientcreatebooking

Last update: **2026/08/07 19:24**