

INTEGRACIJA NEZAVISNE DHTML KOMONENTE U APEX UNUTAR ORACLE BI SUSTAVA



Hrvoje Torbašinović
Davor Delač
IN2 d.o.o.
Zagreb, Marohničeva 1/1
hrvoje.torbasinovic@in2.hr
davor.delac@in2.hr
www.in2.hr

SAŽETAK

Oracle Application Express (APEX) je alat za brzi razvoj poslovnih aplikacija. Korisnik kroz internetski preglednik razvija aplikaciju koristeći predefinirane komponente. Ponekad te komponente ne zadovoljavaju sve potrebne zahtjeve, primjerice, za složenu manipulaciju podataka iz Excel datoteke. U tom slučaju funkcionalnost se može proširiti nezavisnom AJAX komponentom. U ovom radu opisana je integracija komponente dhtmlxGrid u APEX aplikaciju na primjeru unosa podataka iz Excel datoteke u skladište podataka, poštujući poslovna pravila filtriranja i mapiranja podataka prema zahtjevima korisnika. Navedena aplikacija razvijena je u svrhu proširenja funkcionalnosti Oracle BI sustava u sklopu stvarnog projekta.

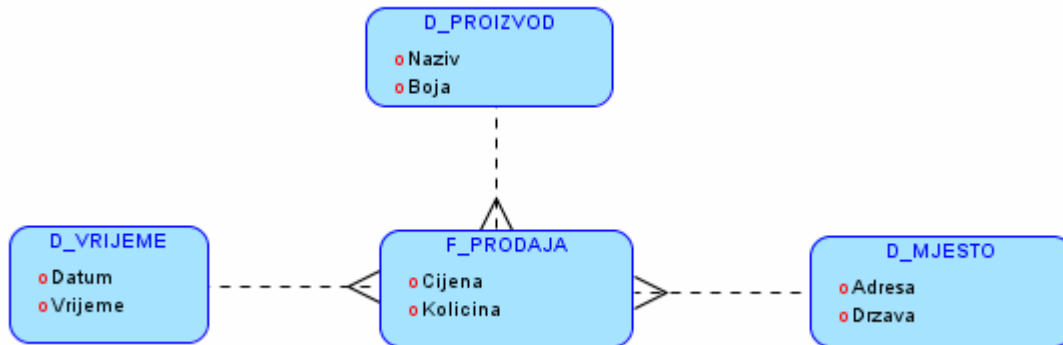
UVOD

Razvoj poslovnih aplikacija jedan je od najraširenijih oblika djelatnosti u IT industriji. U današnje doba, na tržištu postoji cijeli niz platformi koje olakšavaju i ubrzavaju izradu poslovnih aplikacija. Oracle već dugi niz godina nudi razne sustave za razvoj u web okruženju. Oracle Application Express (APEX) je jedna od takvih platformi, koja danas nudi najbrži i najjednostavniji razvoj manjih poslovnih sustava. Pored podrške za razvoj poslovnih aplikacija, Oracle nudi i niz poslovnih rješenja, poput sustava Oracle Business Intelligence (OBI). U takvoj „šumi“ poslovnih rješenja razvoj aplikacija iziskuje ne samo tehničko znanje, nego i fleksibilnost programera da se prilagodi ponudi te da postojeće sustave što efikasnije prilagodi zahtjevima klijenata.

Razvoj poslovnih aplikacija često se svodi na izradu formi za unos podataka u bazu podataka i izvještaja za prikaz podataka koji se u bazi nalaze. Sustavi za brzi razvoj aplikacija (engl. Rapid Application Development - RAD) većinom se baziraju oko ta dva zahtjeva. Takve je sustave moguće iskoristiti i kada su sami zahtjevi složeniji, ali se nerijetko u takvom slučaju mora pribjeći proširenju sustava vanjskim tehnologijama. U ovom radu opisan je Apex OBI, sustav za obradu Excel datoteka i unos podataka iz takvih datoteka u skladište podataka. Zahtjev za takvim sustavom proizašao je iz potrebe da se proširi funkcionalnost OBI-a kako bi se olakšala obrada velike količine podataka. U sljedećim poglavljima opisani su zahtjevi aplikacije, razmatrana rješenja te arhitektura aplikacije. Opisana je i integracija vanjske dhtmlx komponente s APEXom te veza aplikacije sa sustavom Oracle BI.

1. ZAHTJEVI APLIKACIJE

Aplikacija Apex OBI služi unosu podataka iz Excel datoteke u skladište podataka. Zahtjev za takvom aplikacijom proizlazi iz potrebe da se brojni izvještaji, generirani kao Excel dokumenti, na što jednostavniji način predaju sustavu Oracle BI na obradu. Primjer organizacije podataka u skladištu podataka dan je Slika 1 - **Organizacija skladišta podataka**.



Slika 1 - Organizacija skladišta podataka

Skladište podataka prikazano Slika 1 - **Organizacija skladišta podataka** sastoji se od mjera i dimenzija. Mjere su tablice s najčešće numeričkim podacima poput prodaje, cijena, količina i sl. Dimenzije daju opis i kontekst podataka zapisanih u mjerama, poput proizvoda ili prodajnih mjesta za koje je podatak zapisan u mjeri vezan. Iz ovakve organizacije proizlaze slijedeći zahtjevi:

- korisnik mora odrediti podatke iz Excel dokumenta koji se obrađuju,
- korisnik mora definirati mapiranje svakog podatka iz skupa odabranih podataka na kolonu tablice dimenzije ili mjere,
- svi podaci koji se mapiraju na kolonu dimenzije moraju biti definirani u tablici dimenzije ili se unjeti u tablicu dimenzije prije unosa mjere kako bi se osigurao referencijski integritet.
- Unos novih podataka u mjere uz poštivanje referencijskog integriteta.

Pored ovako definiranih poslovnih zahtjeva, pri izradi aplikacije trebalo je uzeti u obzir i posebne zahtjeve za korisničkim sučeljem. Korisničko sučelje mora biti intuitivno i jednostavno te omogućiti što pregledniji i ugodniji rad. Aplikacija je zamišljena kao proširenje sustava Oracle BI, stoga mora raditi kao web aplikacija u browseru.

2. MOGUĆA RJEŠENJA PROBLEMA

Prilikom razmatranja problema, s tehničke strane porebno je razmotriti što više opcija i tehnologija kako bi bili sigurni da smo izabrali optimalno rješenje. Razmatrane tehnologije vezane su za Oracle i Javu. Korištenje JDevelopera i ADF-a možda bi bilo najpotpunije i najjednostavnije rješenje. Pošto se nebi koristila većina funkcionalnosti takve robusne platforme, nebi bilo opravdano pokriti troškove njenog korištenja za aplikaciju ove veličine. Apex, s druge strane, predstavlja jednostavno i besplatno rješenje, ali ne sadrži komponente poput tablice sa omogućenim označavanjem podataka za unos koja je potrebna za obradu podataka iz Excel datoteke. Ostale mogućnosti poput Microsoft tehnologija ili Java EE platforme odbačene jer bi izrada aplikacije u njima predugo trajala.

Unatoč nedostatku APEXa odlučili smo se da upravo njega koristimo pošto se njegova funkcionalnost grafičkog sučelja može proširiti JavaScriptom. U ovom slučaju odlučili smo se za gotovo rješenje dhtmlxGrid. DhtmlxGrid je JavaScript grid kontrola koja uključuje čitav niz funkcionalnosti poput odabira podataka po redcima ili regijama, punjenja zapisima iz XMLa, JSONa, JS arraya ili HTML tablice te iznimnim performansama pri radu sa velikom količinom podataka. Dodatno omogućava i filtriranje, grupiranje u uređivanje podataka. Napredne tehnike poput *Smart Rendering-a* omogućavaju prikaz Excel datoteke sa nekoliko desetaka tisuća zapisa bez potrebe za djeljenjem podataka na više stranica. Naposljetku, dhtmlxGrid komponenta pokazala se kao kvalitetno rješenje uz pristupačan trošak.

Sales	Book		Price	Delivery terms		Bestse
				In Store	Shipping	
▼ -150	A Time to Kill	John Grisham	\$12.99	<input checked="" type="checkbox"/>	24 Hours	
▲ 1000	<u>Blood and Smoke</u>	Stephen King	\$0	<input checked="" type="checkbox"/>	24 Hours	
✗ -100	Boris Godunov	Alexandr Pushkin	\$7.15	<input checked="" type="checkbox"/>	1 Hour	
1000 ▼ 200	<u>The Rainmaker</u>	John Grisham	\$7.99	<input type="checkbox"/>	2 days	
▲ 350	The Green Mile	Stephen King	\$11.10	<input checked="" type="checkbox"/>	24 Hours	
▲ 700	Misery	Stephen King	\$7.70	<input type="checkbox"/>	na	
▼ -120	The Dark Half	Stephen King	\$0	<input type="checkbox"/>	2 days	
▲ 1500	The Partner	John Grisham	\$12.99	<input checked="" type="checkbox"/>	2 days	
▲ 500	It	Stephen King	\$9.70	<input type="checkbox"/>	na	
▲ 400	Cousin Bette	Honore de Balzac	\$0	<input checked="" type="checkbox"/>	1 Hour	

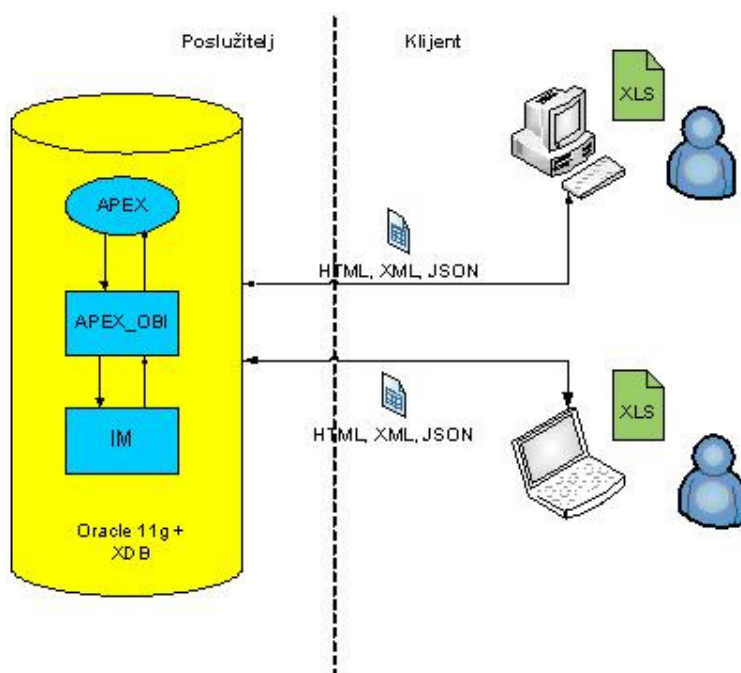
Slika 2 - Komponenta dhtmlxGrid

Uz APEX kao rješenje za probleme poslovne logike i platforme za web aplikaciju i ovakve obogaćene komponente moguće je uz malene troškove pokriti, u kratkom vremenu, sve zahtjeve opisane u predhodnom poglavlju.

3. ARHITEKTURA APLIKACIJE

Sustav Apex OBI je klijent-server web aplikacija razvijena korištenjem sustava APEX. Centralna komponenta sustava je Oracle 11g u sklopu koje je implementirana veza sa skladištem podataka i poslovna logika obrade Excel datoteka. Kao poslužitelj koristi se Oracle XML DB (XDB), ali se kao alternativa može koristiti i Apache Tomcat i mod_plsql.

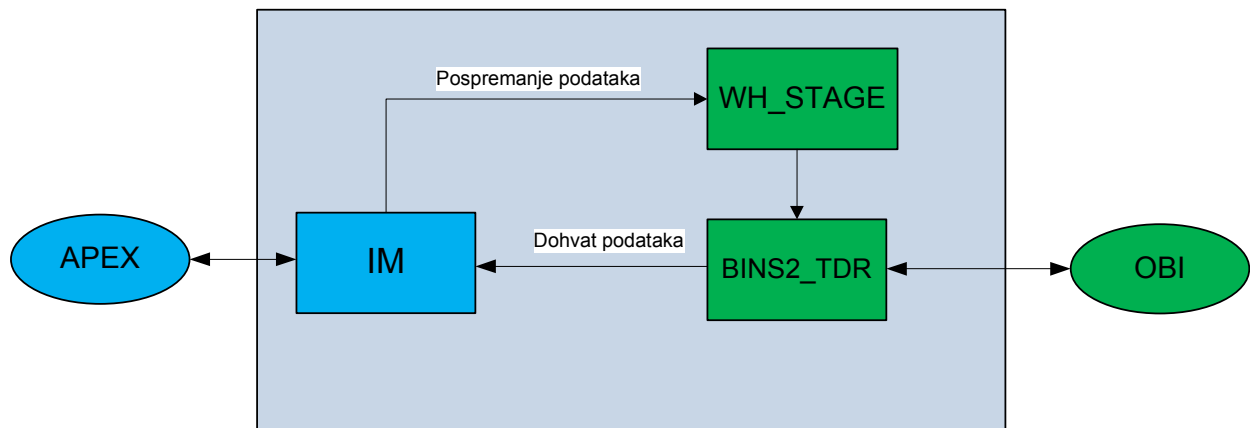
Serverski sloj implementiran je u Oracle Application Express (APEX) sustavu koji je dio Oracle 11g baze podataka, a logika aplikacije djelom je implementirana u sklopu baznog korisnika APEX_OBI. U sklopu baznog korisnika APEX_OBI implementirana je baza za pospremanje excel dokumenata, predložaka i primjena predložaka na dokumente, kao i procedure koje se brinu o stvaranju predložaka i upisivanju obrađenih podataka u skladište podataka. Jednostavni prikaz arhitekture aplikacije dan je Slika 3 - **Arhitektura aplikacije Apex OBI**.



Slika 3 - Arhitektura aplikacije Apex OBI

4. VEZA SA SUSTAVOM ORACLE BI

Bazni korisnik IM služi kao veza između baze predložaka i skladišta podataka u koju se unose podatci iz excel dokumenata. Ovaj korisnik služi isključivo kao sučelje te mora sadržavati poglede na sve relevantne tablice mjera i dimenzija u skladištu podataka te definirati "instead of" triggere za unos podataka u te tablice. Na ovaj način riješena je komunikacija između sustava Apex OBI i Oracle BI, ali je i ostavljeno prostora za moguću integraciju sa bilokojim drugim sustavom baziranim na skladištima podataka.



Slika 4 - Komunikacija sustava Oracle BI i Apex OBI

Opisano sučelje na jednostavan način rješava problem prijenosa podataka između dva sustava. Dodatno, potrebno je u sustavu OBI omogućiti korištenje aplikacije APEX obi uz iluziju da je riječ o djelu OBI-a. Kako bi na jednostavan način riješili ovaj problem, aplikaciju smo dodali u obi preko HTML elementa <iframe>. Jedini problem kod ovakvog pristupa je omogućiti *single sign-on* između dva sustava. Ova opcija je podržana pri korištenju LDAP-a te se stoga LDAP preporučuje kao izbor za autentifikacijsku i autorizacijsku shemu kod ovakvih sustava.

5. INTEGRACIJA DHTMLX KOMPONENTE U APEX

Apex se oslanja na PL/SQL za implementaciju poslovne logike na serverskoj strani te HTML i JavaScript za prezentaciju na klijentskoj strani. Zbog ovoga pristupa veoma je jednostavno oplemeniti prezentacijski sloj dodatnim komponentama poput dhtmlxGrida. Dovoljno je uključiti sve potrebne skripte u aplikaciju na jedan od nekoliko mogućih načina. Jedan od načina je da se skripte dodaju u zaglavlje stranice tako da se unesu u HTML Header atribut stranice. To je dobar pristup ako se koriste funkcije specifične za određenu stranicu ili ako se želi testirati neka funkcija prije nego se uključi u .js datoteku. Drugi način je kreirati praznu HTML regiju na stranici i u nju upisati skripte. Treći način je uključiti skripte unutar predloška stranice (page template) gdje se također mogu referencirati .js datoteke. Na ovaj način su datoteke dostupne cijeloj aplikaciji. Ovo je najefikasniji način zato što se .js datoteka učita prilikom prvog pregleda stranice u aplikaciji i pohrani u internetskom pregledniku. Skripte se u HTML-u dodaju jednostavnim konstruktom:

```
<script src="url"></script>
```

Za korištenje dhtmlxGrid komponente potrebno je na ovaj način uključiti sljedeće skripte:

```
<script src="/i/app1/codebase/dhtmlxcommon.js"></script>
<script src="/i/app1/codebase/dhtmlxgrid.js"></script>
<script src="/i/app1/codebase/dhtmlxgridcell.js"></script>
```

Ako se javi potreba za složenijom funkcionalnosti vezanom za vanjsku komponentu, poželjno je JavaScript kod odvojiti u zasebne datoteke. Takve datoteke treba pospremiti na način da im se može pristupiti preko URL-a te ih uključiti u HTML na isti način kao i skripte za komponentu, npr:

```
<script src="/i/appl/codebase/page1.js"></script>
```

Sve JavaScript datoteke referencirane u primjerima spremjene su u virtualni /i/ direktorij. Kako se spremaju datoteke u /i/ direktorij ovisi o HTTP Serveru kojeg koristi APEX. Ako se koristi Oracle HTTP Server i mod_plsql sadržaj /i/ direktorija je spremljen na datotečnom sustavu na lokaciji referenciranoj aliasom /i/. Lokacija se može saznati iz datoteke *dads.conf* (ako se koristi Oracle HTTP Server koji dolazi s bazom 11g) ili iz datoteke *marvel.conf* (ako se koristi Oracle Application Server 10g). Ako se koristi Embedded PL/SQL Gateway sadržaj direktorija /i/ je spremljen direktno u bazi unutar Oracle XML DB repozitorija. Direktoriju se može pristupiti WebDAV protokolom tako da se u Windowsima kreira WebFolder. Za lokaciju WebFoldera treba upisati:

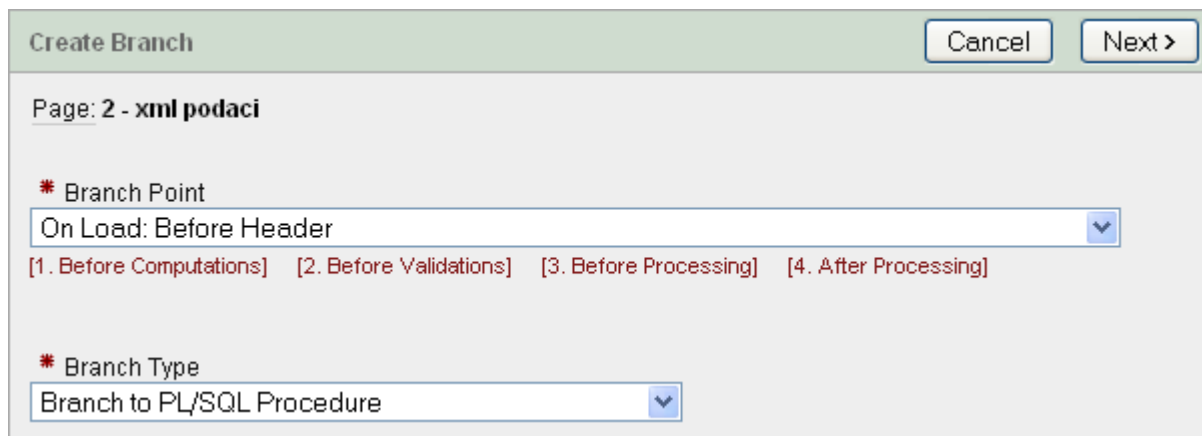
`http://Oracle_server_name:HTTP_port_number`

6. KOMUNIKACIJA APEX-a i DHTMLX KOMPONENTE

Da bi se u DhtmlxGrid komponenti mogli prikazati i obrađivati podaci, moraju se prvo dohvatiti iz baze. Komponenta može učitati podatke na dva načina: iz URL-a i iz stringa. Formati koje prepoznaje su XML, CSV, JSON i JS Array. Nakon manipulacije podacima komponenta mora poslati obavijest o novom stanju. Komunikacija se odvija preko APEX-a i to na dva načina: pozivom stranice na kojoj se nalazi XML i pozivom aplikacijskog procesa.

6.1. Poziv stranice na kojoj se nalazi XML

U aplikaciji Apex-OBI za dohvaćanje podataka u DhtmlxGrid komponentu koristi se poziv stranice na kojoj se nalazi XML. XML se generira u pl/sql proceduri koja se poziva prilikom učitavanja stranice. Za ostvarivanje komunikacije prvo je potrebno kreirati praznu stranicu i na njoj jedan *Branch*. Tip grananja je *Branch to PL/SQL Procedure*, a vrijeme grananja je *On Load: Before Header*.



Slika 5 - Kreiranje grananja

Na sljedećem ekranu je potrebno upisati ime procedure u bazi koja će generirati XML kojim će se puniti komponenta. U nastavku je dan primjer jedne takve procedure.

```

begin
  owa_util.mime_header('application/xml', false);
  owa_util.http_header_close;
  htp.p('<?xml version="1.0" encoding="UTF-8"?>');
  htp.p('<rows>');
  htp.p('  <head>');
  htp.p('    <column type="ed">Ime</column>');
  htp.p('    <column type="ed">Prezime</column>');
  htp.p('  </head>');
  htp.p('  <row id="1">');
  htp.p('    <cell>Hrvoje</cell>');
  htp.p('    <cell>Torbašinović</cell>');
  htp.p('  </row>');
  htp.p('  <row id="2">');
  htp.p('    <cell>Davor</cell>');
  htp.p('    <cell>Delač</cell>');
  htp.p('  </row>');
  htp.p('</rows>');
  apex_application.g_unrecoverable_error := true;
end;

```

Posljednja naredba (`apex_application.g_unrecoverable_error := true;`) govori APEX-u da stane s procesiranjem i prikazivanjem stranice. Bez nje bi APEX u stranicu dodao još dio koda koji dodaje na svaku stranicu i generirani XML više ne bi bio valjan.

Nakon izrade stranice na kojoj se nalazi XML, potrebno ju je pozvati iz `DhtmlXGrid` komponente i tako inicijalizirati samu tablicu. Za početak, potrebno je definirati `<div>` element u HTML-u unutar kojega ćemo smjestiti tablicu:

```

<div id="gridbox" style="width:500px; height:470px;
background-color:white;overflow:hidden"></div>

```

Sama inicijalizacija tablice vrši se u sklopu JavaScripta. Najjednostavnije je definirati HTML element `<script>` koji sadrži sljedeći odsječak koda:

```

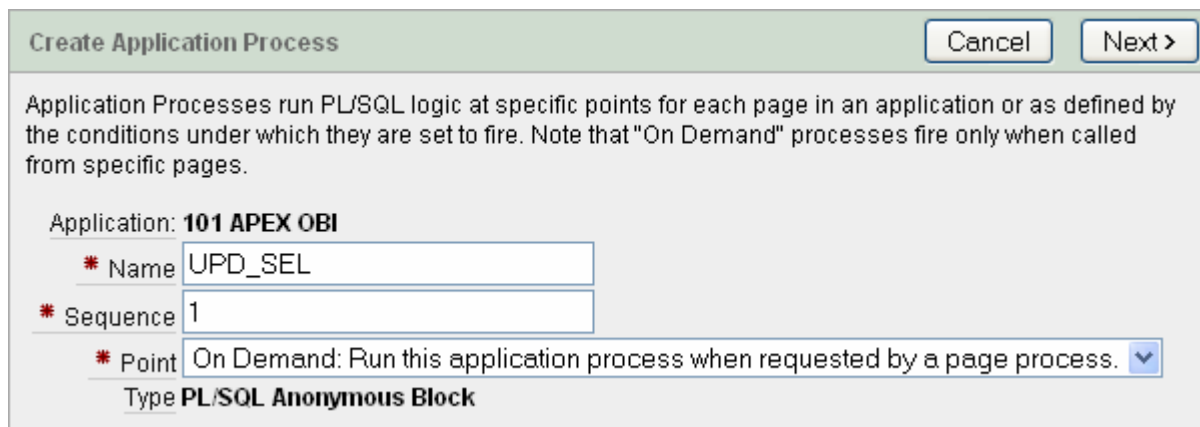
var mygrid = new dhtmlXGridObject('gridbox');
mygrid.setImagePath("i/appl/slike");
mygrid.init();
mygrid.attachEvent("onXLE", function()
{
  doAfterLoad();
});
mygrid.loadXML("/apex/f?p=&APP_ID.:2:&SESSION.:::::" );
mygrid.setSkin("dhx_skyblue");

```

Prva linija deklarira varijablu `mygrid` te stvara `dhtmlXGrid` objekt čija će referenca biti pospremljena u varijabli `mygrid`. Druga i treća linija postavljaju putanju na slike i inicijaliziraju tablicu. Na tablicu je moguće postaviti rutine za obradu razni događaja. U primjeru se na događaj "onXLE" postavlja funkcija za obradu "doAfterLoad". Događaj "onXLE" se okida nakon što je završeno punjenje tablice podacima pozivom funkcije "loadXML". U primjeru se tablica puni podacima koji su dobiveni iz XML-a generiranog na APEX stranici 2. To se ostvarije jednostavnim pozivom funkcije "loadXML" kojoj se kao parametar predaje APEX URL. Ovdje posebnu pozornost treba uputiti na *cross-site scripting*. Pošto je `dhtmlXGrid` implementiran u JavaScriptu, funkciji `loadXML` nije moguće predati URL koji pokazuje na neku drugu domenu. Svi moderni preglednici će blokirati `XMLHttpRequest` iz JavaScripta prema stranicama izvan domene aplikacije.

6.2. Poziv aplikacijskog procesa

Kod razvoja složenih aplikacija javlja se potreba za pozivom procedura na serverskoj strani te slanjem podataka iz vanjske komponente na server. APEX ima ugrađeno rješenje za komunikaciju klijenta i servera u JavaScriptu kompatibilnu sa svim većim Internet preglednicima.



The screenshot shows a dialog box titled "Create Application Process". It has "Cancel" and "Next >" buttons. The text inside reads: "Application Processes run PL/SQL logic at specific points for each page in an application or as defined by the conditions under which they are set to fire. Note that 'On Demand' processes fire only when called from specific pages." Below this, the application is identified as "101 APEX OBI". The fields are: Name: "UPD_SEL", Sequence: "1", Point: "On Demand: Run this application process when requested by a page process." (selected from a dropdown), and Type: "PL/SQL Anonymous Block".

Slika 6 - Kreiranje aplikacijskog procesa

Za početak potrebno je kreirati "On Demand" aplikacijski proces. Tip procesa je pl/sql anonimni blok i u njemu se pišu naredbe koje se trebaju izvršavati u bazi. Dobra praksa je sav kod koji sadrži logiku aplikacije spremirati u procedure i funkcije u bazi, a iz APEX-a ih samo pozivati s određenim parametrima. Tada je za tekst procesa primjerice potrebno samo upisati sljedeće:

```
:PLK_ID := pck_ao_http.f_create_template(:P5_NAZIV, :P5_JSON)
```

Pozivanje kreiranog aplikacijskog procesa iz JavaScripta demonstrirano je sljedećim isječkom koda:

```
var get = new htmldb_Get(null, $('#pFlowId').value,
'APPLICATION_PROCESS=UPD_SEL', 0);
get.add("P5_JSON", JSON.stringify(dataContainer));
var gReturn = get.get();
get = null;
```

Objekt `htmldb_Get` koristi se za RPC (*engl. Remote Procedure Call*) i to na način da se direktno poziva proces definiran u APEX-u. Parametri se mogu prenositi preko *itema* na stranici kao što je to u primjeru pokazano postavljanjem vrijednosti parametra procesa "UPD_SEL" u item "P5_JSON". Za prijenos veće količine podataka ili strukturiranih podataka preporuča se korištenje formata *JSON* (JavaScript Object Notation) pošto je podržan za većinu programskih jezika, pa tako i PL/SQL.

ZAKLJUČAK

Opisanim pristupom uspješno smo razvili web aplikaciju za unos podataka iz Excel datoteka u skladište podataka. Korištenje APEX-a i vanjskih dhtml komponenata ubrzalo je razvoj aplikacije te smanjilo troškove razvoja u usporedbi sa razmatranim tehnologijama. Aplikaciju smo radili sa manje od pola godine radnog iskustva u razvoju poslovnih aplikacija, a od zamisli do realizacije utrošeno je nešto manje od četiri čovjek/mjeseca.

Integracijom APEX-a sa vanjskim dhtml komponentama dobivamo platformu za brzi razvoj web aplikacija sa iznimno složenim sučeljima. Ovim pristupom moguće je napraviti kvalitetne sustave u kratkom roku uz veoma maleni trošak u usporedbi sa ostalim opcijama. Mane pristupa su potreba za naprednim poznavanjem APEX-a i JavaScripta te nekompatibilnost jezičnih procesora za JavaScript u Internet preglednicima. Unatoč svemu, ovaj pristup svakako treba uzeti na razmatranje kod planiranja razvoja web aplikacija.