



“Kad je reč o razvoju dinamičkih Web aplikacija, PHP programeri su dosad imali osnovni editor izvornog koda i alate za debugovanje. Ono što zaista nedostaje u PHP razvoju su alati za brz vizualni razvoj i komponente i framework koji će da konkurišu alatima poput Visual Studio-a i Delphi-a i framework poput onog u ASP.NET-u”, rekao je Michael Swindell, CodeGear-ov potpredsednik proizvoda i strategije. ...a Delphi for PHP to upravo jeste.

Slika 1

Dakle, šta je Delphi for PHP?

Delphi for PHP je relativno novi product kompanije CodeGear koji služi za razvoj PHP aplikacija, php klasa i komponenata i sve to u Delphi stilu. Delphi for PHP je IDE (skraćeno od eng. integrated development environment – integrisano razvojno okruženje) RAD (skraćeno od engleskog Rapid Application Development – brz razvoj aplikacija) razvojno okruženje koje u sebi sadrži dizajnerski deo, editor koda, debugger, deployment wizard, komponente pod nazivom “VCL for PHP” koje sadrže veliki broj standardnih vizualnih komponenti, ali i koje su naravno prozirive, tj. možete i sami da kreirate svoje komponente i dodate ih ovom programerskom alatu, Delphi for PHP sadrži takođe i gotovo rešenje za brzo i lako povezivanje sa bazama podataka i integrisan php debugger.

VCL za PHP baziran je na najpopularnijim open source PHP skriptama i bibliotekama, to je open source biblioteka raspoloživa na SourceForge-u na adresi: <http://sourceforge.net/projects/vcl4php> Sastoji se iz vizualnih i nevizualnih komponenti. Vizualne se pojavljuju za vreme dizajniranja (programiranja) na formi ali su one vidljive i krajnjem korisniku za vreme izvršenja naše web aplikacije, u te komponente spadaju dugmadi, labele, listbox-ovi memo komponente i dr. dok nevizuelne koje dodajemo su one koje one su vidljive samo za vreme dizajniranja (programiranja) i one se ne vide za vreme izvršenje naše aplikacije, ali naravno one su bitan factor u programiranju.

Kada dodate neku od komponenta na formu ona je prikazana kao ikonica kojoj u ObjectInspector-u možemo da menjamo svojstva i prilagodjujemo date komponente svojim potrebama.

Probna verzija Delphi for PHP-a može se naći na lokaciji:

<http://cc.codegear.com/free/delphiphp>

Lakoću njegovog korišćenja pri izradi php aplikacija, uz kraća objašnjenja pojedinih delova ovog razvojnog okruženja pokušaću da vam približim kroz jedan mali primer u kome ću vam pokazati kako možete da kreirate svoj adresar na internetu.

Napomena, pre nego što predjemo na programiranje u D4PHP-u (Delphi for PHP) nešto malo i o onome što će nam trebati za rad u pozadini pošto ćemo da razvijamo ovaj primer na localhostu, a možemo da naučimo i nešto novo.

Postoje više alata koji će vam obezbediti podršku za php, i rad sa mysql bazom podataka na vašoj masini, ja sam se opredelio za WAMP server. (može se preuzeti sa adrese - <http://www.en.wampserver.com>)

Inače da bi mogli da testiramo na našem kompjuteru web aplikacije koje ćemo da pravimo u Delphi for PHP-u moramo da imamo podršku za PHP i MySQL (jer ćemo raditi i sa bazom podataka). WAMP će nam to omogućiti jer sadrži PHP, MySQL, Apache u sebi...

Da bi instalirali WAMP server samo pratite veoma jednostavnu instalaciju i uputstva koja vam daje installer. Kad server bude instaliran možete ga pronaći u sys tray-u (delu pored časovnika u donjem desnom uglu vašeg monitora). Pokrenućemo phpMyAdmin jednostavnim klikom na ikonicu servera koji smo instalirali i u padajućem meniju izabraćemo stavku phpMyAdmin i u browseru će se otvoriti novi prozor.

Možemo da koristimo phpMyAdmin da kreiramo baze podataka, tabele unutar baza i da dodajemo, menjamo, i brišemo sadržaj unutar baze. Jednostavno ćemo da ukucamo ime baze koju želimo da kreiramo u polje pod nazivom "Create new database" i kliknemo na dugme Create i automatski će biti kreirana baza pod imenom koje ste uneli.

Sada kada smo kreirali bazu podataka, veoma lako možemo da napravimo i tabelu u toj bazi u koju ćemo da smeštamo podatke koje unosimo. Jednostavno ćemo uneti ime tabele koju želimo da napravimo (u našem slučaju neka stoji "podaci" za ime tabele) i unecemo broj polja koja ćemo da koristimo (ja sam stavio za ovaj nas primer 5 – jedno za id, jedno za ime, jedno za prezime, jedno za adresu i jedno za broj telefona recimo).

Nakon što unesemo podatke za ime i broj polja tabele koju pravimo jednostavno ćemo da kliknemo na GO taster ba bi kreirali tabelu "podaci" u bazi podataka pod imenom "BazaAdresar".

Sada, pošto smo odredili broj polja koji ćemo da koristimo, da bi ih razlikovali moramo da unesemo i njihova imena, unecemo imena naših polja i tip za svako polje, ja ovde koristim u ovom primeru jedino za polje ID ono je celobrojni tip podataka (INT) jer ćemo da koristimo brojeve u tom polju, a za sva ostala polja ćemo da stavimo VARCHAR, jer ćemo da koristimo neke tekstualne podatke za vrednosti tih polja.

Takodje, vi možete da definišete i veličinu vrednosti koja će se unositi u datim poljima i još gomilu drugih podešavanja, ali posto je ovo osnovni tutorijal taj deo ćemo preskočiti, jer nam trenutno nije potreban.

Takodje možemo da definišemo i koje polje će nam biti polje primarnog ključa; u našem slučaju stavićemo da bude polje ID. Kada završite sa unosom imena polja i podešavanjem vezanim za tip polja i određivanje primarnog ključa jednostavno kliknite na "save" i biće kreirana i tabela unutar baze sa datim poljima, cime ste uspešno završili prvi deo ovog malog primera koji tek sledi.

Naravno, ne morate koristiti phpMyAdmin da bi kreirali bazu i tabelu unutar baze, to možete da uradite (isprogramirate) i u php-u sledecim kodom:

PHP kod za pravljenje baze:

```
<?php
```

```
/*
```

Umesto korisnickog imena i lozinke stavite vase podatke za povezivanje na vas server

```
Najcesce je mysql_connect("localhost","","");
```

```
Ili mysql_connect("localhost","root","");
```

Kada se radi o verziji wamp-a koji sam vam preporucio povezivanje na server se vrsi tako sto za korisnicko ime stavite root a za lozinku ne stavite nista, dakle mysql_connect("localhost","root","");

```
*/
```

```
$con = mysql_connect("localhost","root","");
```

```
mysql_query("CREATE DATABASE BazaAdresar") ;
```

```
echo "Napravljena je baza!";
```

```
?>
```

PHP Kod za pravljenje tabele:

```

-----
<?php
$con = mysql_connect("localhost","korisnickoime","lozinka");
    if (!$con)
        {
            die('Ne moze da se povezem na server! Greska: ' . mysql_error());
        }

    mysql_select_db("BazaAdresar", $con);

    mysql_query("CREATE TABLE
    podaci(
    id INT NOT NULL AUTO_INCREMENT
    ,PRIMARY KEY(id)
    ,ime VARCHAR(30)
    ,prezime VARCHAR(30)
    ,adresa VARCHAR(50)
    ,telefon VARCHAR(30))")
    or die(mysql_error());
    echo "Napravljena tabela!";
}
?>
-----

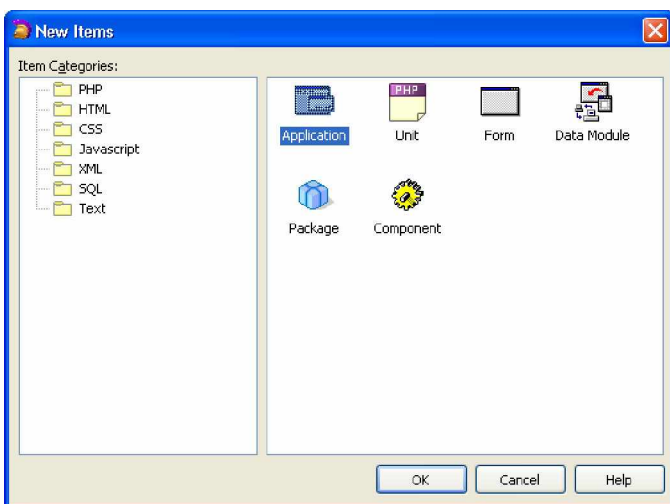
```

No, nećemo se previse zadržavati ovde, u daljem primeru ću vam pokazati kako mozete da se brzo i lako povežete na bazu podataka, kako da u nju unosite podatke, i kako da ih menjate, prikažete...

Prvi korak

Nakon sto ste downloadovali, i instalirali vas D4PHP i pokrenuli ga primetićete deo za rad sa vašim projektima, kreiranje novih, otvaranje postojećih projekata, fajlova, pomoc itd.

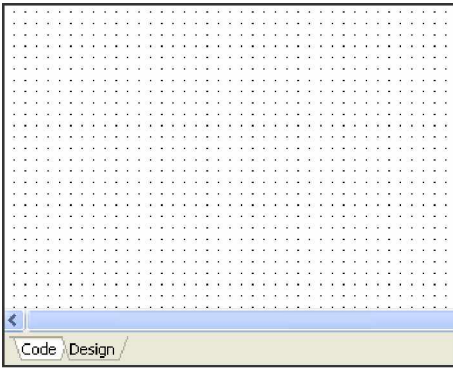
Pošto mi želimo da napravimo novi primer, novu aplikaciju, izabracemo opciju “new” i dočekace nas prozor kao na slici 2.



Slika 2.

U novootvorenom prozoru biramo da li zelimo da pravimo D4PHP aplikaciju, php aplikaciju, formu, html stranu, css fajl ili nesto drugo. Posto smo odlucili da napravimo adresar korisceni d4php i njegove vcl komponente, njegove RAD i IDE čari, izabraćemo u okviru new php items-a “Application” opciju i dočekace nas razvojno okruženje u kome možete da primetite nekoliko važnijih celina, to su dizajnerski deo razvojnog

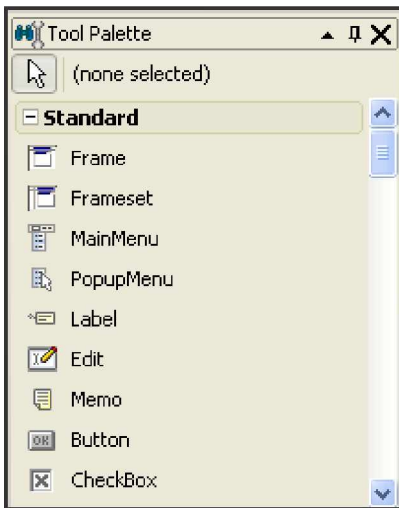
okruženja (slika 3), code view - deo u kome unosimo kod nase aplikacije, deo sa VCL komponentama (slika 4) i object inspector (slika 5) u kome možete da definišete svojstva nekih od objekata aplikacije.



Dizajnerski deo

Dizajnerski deo služi za kao što sam naziv kaže za dizajniranje vaše web aplikacije. Na ovaj deo postavljate sve vaše komponente i prilagođavate izgled vaše aplikacije.

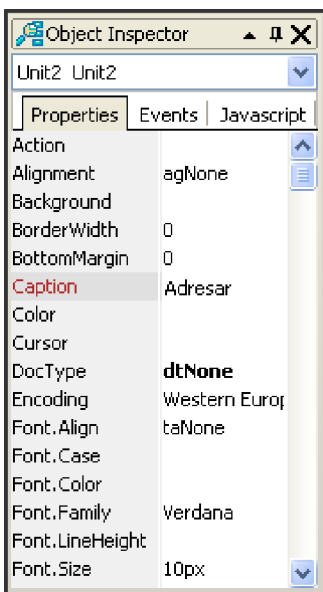
Slika 3



Paleta sa alatima

Paleta sa alatima sadrži vizuelne i nevizuelne komponente koje možete koristiti dok razvijate vašu aplikaciju. Komponentu možete dvostrukim klikom dodati direktno na formu. Ukoliko posmatrate kod vaše aplikacije preko Kod Editora, paleta sa alatima prikazuje segmente koje možete dodati vašoj aplikaciji.

Slika 4



Inspektor Objekata

Inspektor objekata vam dozvoljava da podešavate svojstva za vreme programiranja (dizajniranja, projektovanja) i da napravite handlerne događaja za neku komponentu. Ovo omogućava vezu između izgleda vaše aplikacije i koda koji je pokreće. Inspektor objekata sadrži tri kartice:

Svojstva, Događaji i JavaScript događaji.

Karticu sa svojstvima možete da koristite da promenite attribute komponenti koju koristite, dodate joj neku početnu vrednost, promenite izgled, ako se radi o logičkoj komponenti dodelite true ili false logičko stanje i sl.

Karticu sa događajima možete da iskoristite da definišete neku akciju koja će se izvršiti na određen događaj vezan za datu komponentu, npr. Neki događaj na pokret miša, na aktivaciju vašeg programa i sl.

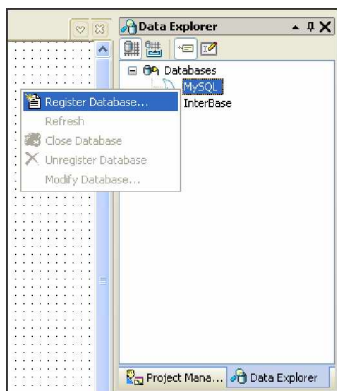
Karticu sa JavaScript događajima možete koristiti da bi odredili JavaScript događaj izabrane komponente.

Slika 5

Prvi korak će nam biti povezivanje baze koje smo napravili i tabele unutar te baze sa našom web aplikacijom. U gornjem desnom delu D4PHP-a možete primetiti “Data Explorer” (slika 6) deo, on nam omogućuje da povežemo baze, i da pretražujemo strukturu tih baza, tabele i polja unutar njih, a isto tako veoma popularan metod drag and drop (prevuci i pusti) i da veoma brzo napravimo aplikaciju koja koristi bazu koju smo prevukli na dizajnerski deo, tj. na glavnu formu našeg programa.

U “Databases” delu unutar Data Explorer-a primetićete dva poddela, MySQL i InterBase, oni predstavljaju tip baze koje možemo da koristimo. U našem primeru koristićemo MySQL bazu podataka jer ogroman broj servera na internetu podržava ovu bazu podataka a i naš WAMP server takođe.

Ono što je potrebno to je da registrujemo bazu koju ćemo da koristimo, to radimo tako što otvorimo pomocni PopUp (iskačuci) meni desnim klikom na item MySQL i izaberemo opciju “Register Database...” nakon čega ćete dobiti prozor kao na slici 7. U ovom delu birate tip baze podataka na koju se registrujemo, u našem slučaju znači MySQL, ime za konekciju koje će vam pomoći da se izborite sa većim brojem baza koje koristite da bi mogli da ih lakše raspoznate i sl. Zatim biramo tip hosta na koji se povezujemo, tu treba da unesemo “localhost”, a D4PHP sam odredi port na kome ce da server obradjuje našu web aplikaciju

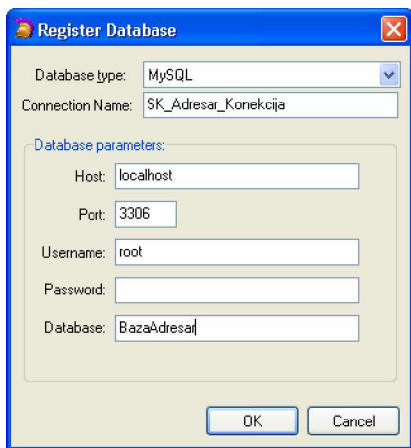


Data Explorer

Data Explorer vam omogućava da pregledate tabele i polja baze podataka. Korišćenjem kontekstnog menija možete praviti i upravljati vezama baza podataka.

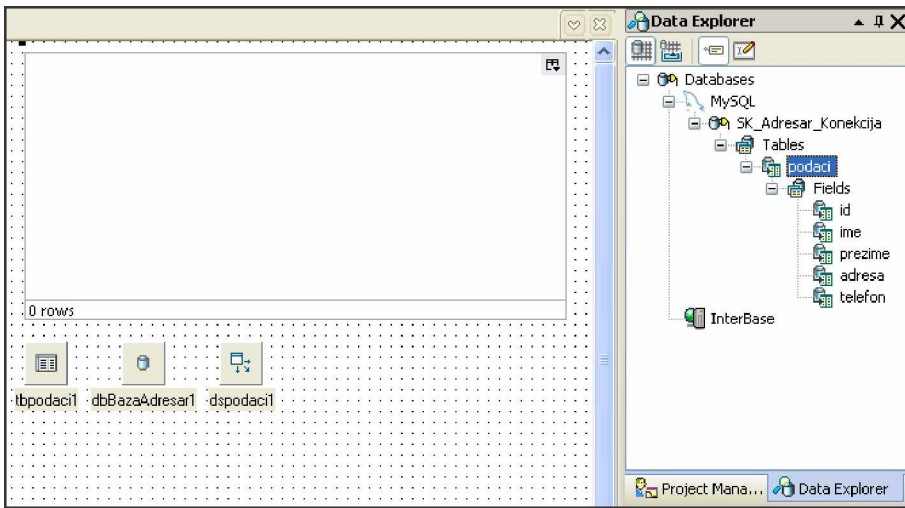
Takođe možete da prevlačite (kopirate, drag&drop) informacije u većinu formi da bi projektovali vaše aplikacije sa bazama podataka brzo.

Slika 6



Slika 7

Nakon što smo uneli sve tražene podatke, klikom na OK dugme u datom prozoru izabrana baza će biti registrovana za korišćenje. Iako je baza registrovana za korišćenje ona nije još uvek uključena u našu aplikaciju, možemo najbrže i najlakše da je uključimo tako što ćemo da prevučemo tabelu iz registrovane baze na formu a D4PHP će automatski da kreira na formi komponente za povezivanje sa bazom, i automatski ih sam poveže (Slika 8).



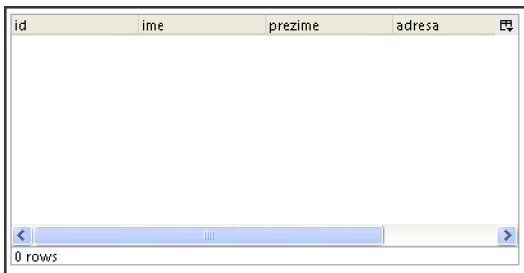
Slika 8

Na formi ćemo imati sada četiri komponente od kojih je jedna vizuelna (DBGrid) i tri koje nisu vizuelne (Table, Database i DataSource) komponente. DBGrid komponenta nam služi za prikaz podataka iz baze, a ove ostale za međusobno povezivanje ovih komponenti među sobom i povezivanje na bazu koju smo registrovali.

Ukoliko bi sada pokrenuli našu aplikaciju (slika 9), u web browseru bi se pokazala samo DBGrid komponenta (slika 10)



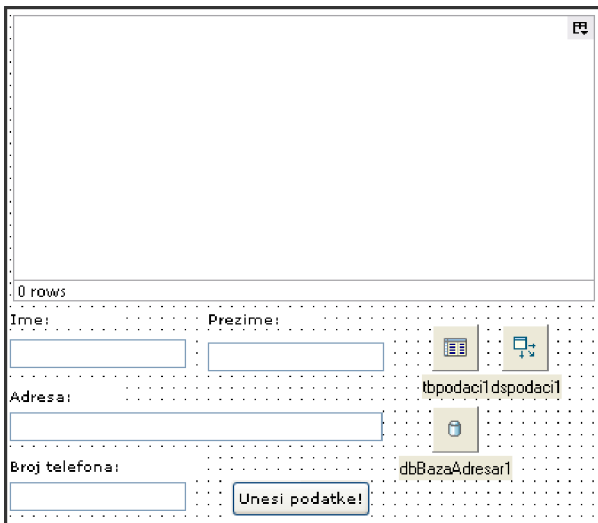
Slika 9



Slika 10

Pošto je naša baza prazna trenutno moramo da napravimo deo za unos podataka u nju da bi naš adresar imao smisla. Posto ćemo da unosimo informacije o imenu, prezimenu, adresi, broju telefona neke osobe u nas adresar dodacemo polja za unos tih informacija. U Tool Palette delu pronadjite “Edit” i label komponente i postavite ih na odredjenim delovima forme. Nama je potrebno cetiri Edit komponenti a label komponente cemo da postavimo da nam daju usmerenje koja Edit komponenta cemu služi i biće nam potrebna jedna Button komponenta koja će da aktivira kod za unos datih podataka. Kad prevučete na formu date komponente i rasporedite ih po vašoj želji u ObjectInspectoru u Properties delu prvo ćemo da promenimo tekst Label komponenata radi lakšeg kasnijeg snalaženja. Tekst Label komponenata promenite tako što u ObjectInspector Properties delu pronadete stavku Caption i dodelite joj vrednost onoga što želite da piše u datoj Label komponenti, ja ću da stavim kratko i jasno Label komponentama vrednost Ime, Prezime, Adresa, i Broj telefona. Button komponenta primetićete da ima isto ovo svojstvo sto znaci da i njoj mozemo da promenimo tekst u nama odgovarajuci, za nas primer ja sam stavio “Unesi podatke!”.

Takodje, svaka od komponenata ima i svoje “Name” svojstvo tj. ime i preko tog svojstva joj se pristupa u kodu. Da bi se lakše snašli mi ćemo da Edit komponentama promenimo imena iz Edit1, Edit2... u EditIme, EditPrezime, EditAdresa, EditTelefon a Button komponenti cemo dodeliti ime “ButtonUnesiPodatke”.



Slika 11

Kada ste završili dizajniranje počecemo sa programiranjem Button komponente, tj. dodacemo mu kod kojim ce da unese podatke koje stavimo u data polja i smesti ih u bazu a onda da se i ti podaci prikažu u DBGrid komponentu.

Da bi isprogramirali dato dugme napravićemo dvoklik na Button komponentu i D4PHP ce vas prebaciti u Code View i tu unesemo kod za unos podataka u bazu! (slika 12)

```

27      function ButtonUnesiPodatkeClick($sender, $params)
28      {
29          $this->tbpodaci1->open();
30          $this->tbpodaci1->append();
31          $this->tbpodaci1->ime = $this->EditIme->Text;
32          $this->tbpodaci1->prezime = $this->EditPrezime->Text;
33          $this->tbpodaci1->adresa = $this->EditAdresa->Text;
34          $this->tbpodaci1->telefon = $this->EditTelefon->Text;
35          $this->tbpodaci1->post();
36          $this->tbpodaci1->close();
37          $this->tbpodaci1->Refresh();
38      }
39
40

```

Slika 12

D4PHP kod za unos podataka u bazu:

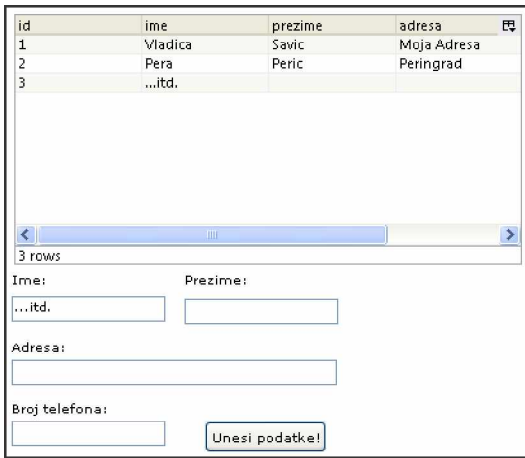
```

-----
function ButtonUnesiPodatkeClick($sender, $params)
{
    $this->tbpodaci1->open();
    $this->tbpodaci1->append();
    $this->tbpodaci1->ime = $this->EditIme->Text;
    $this->tbpodaci1->prezime = $this->EditPrezime->Text;
    $this->tbpodaci1->adresa = $this->EditAdresa->Text;
    $this->tbpodaci1->telefon = $this->EditTelefon->Text;
    $this->tbpodaci1->post();
    $this->tbpodaci1->close();
    $this->tbpodaci1->Refresh();
}
-----

```

U ovom kodu dakle otvaramo tabelu podaci koju smo registrovali i pripremamo je za unos podataka a onda svakom od polja tabele dodeljujemo određenu vrednost, odnosno tekst koji unosimo u Edit komponente, a nakon toga postujemo tekst u bazu, zatvaramo je i osvežimo da bi videli u DBGridu novounete podatke.

Nakon što pokrenemo našu aplikaciju opet i popunimo formular sa informacijama koje smo postavili za unos i kliknemo na dugme “Unesi podatke!” u vašem browseru vas adresar će biti biti “nahrانjen” novim informacijama. (Slika 13)

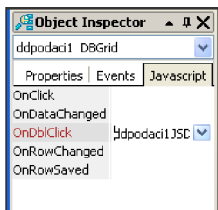


Slika 13

I ovim smo završili prvi deo ovog malog upoznavanja sa ovim razvojnim okruženjem kroz naš primer sa adresarom.

Naravno, ne unose se uvek tačne informacije, često može doći i do grešaka prilikom unosa, i zato je potrebno da isprogramiramo deo koji će da služi iz izmenu unetih podataka.

Za izmenu podataka u našoj bazi, isprogramiraćemo JavaScript na događaj dvoklika na DBGrid komponentu. Kada obeležite DBGrid komponentu, u Object Inspector JavaScript kartici videćete na koje sve događaje JavaScript neki može biti pokrenut, pošto mi želimo da se naš aktivira na dupli klik unutar DBGrida, izabraćemo događaj “OnDbClick” (Slika 14).



Da bi isprogramirali neki događaj, tj. počeli da programirate neki događaj u desnom delu od naziva koji predstavlja na koji događaj će nešto biti aktivirano u prazan prostor desno od tog naziva u istom redu potrebno je da napravite dvoklik i D4PHP će vas prebaciti u editor koda unutar funkcije koja se automatski kreira za dati objekat gde možete da dodajete svoj kod.

Slika 14.

JavaScript kod za omogućavanje izmene podataka u bazi:

```

-----
function ddpodaci1JSDbClick($sender, $params)
{
  ?>
  //Add your javascript code here
  var temp = DBGrid1.getTableModel().getValue(0, DBGrid1.getFocusedRow());
  location = "unit2.php?id=" + temp;
  return(false);
  <?php
-----

```

Ovaj deo koda nam omogućava da kad pokrenemo naš program i kad uradimo dvoklik unutar nekog polja gde se prikazuju informacije koje smo uneli ranije, da možemo da ih izmenimo tako što ukucamo tekst koji želimo da se prikaze u datom polju i potom samo kliknemo na neko drugo polje ili pritisnemo Enter na tastaturi i novouneti tekst automatski biva zapamćen u našu bazu.

Naravno, dodavanje i izmena unetih podataka često nije dovoljno i javi se i potreba da se neki podaci izbrišu iz baze.

Za to možemo da dodamo jos jednu Edit i Button komponentu na formu i daćemo joj ime EditID jer ćemo da brišemo nekog iz naše liste kontakata tako što upišemo njegov ID broj unutar našeg EditID polja i kliknemo na dugme za brisanje, a dato dugme isprogramiram na sledeci nacin.

Kod za dugme za brisanje podataka iz baze

```
-----  
function ButtonIzbrisiClick($sender, $params)  
{  
    $this->dbBazaAdresar1->execute("DELETE FROM podaci WHERE id='".$this->EditID->Text.'");  
    $this->tbpodaci1->Refresh();  
}
```

Ovaj kod kao što možete da vidite izvršava SQL upit koji nam kaže da izbrišemo sve podatke u redu gde je polje id jednako vrednosti koju unesemo u EditID polje na našoj formi, i da se tabela podaci osveži kako bi videli izvršenu akciju.

I čestitam, ako ste preživeli sve ovo dovde, vaša web aplikacija adresar je napravljena.

Sve što vam sad preostaje je da je pokrenete opet i da se igrate malo njome, slobodno eksperimentišite malo, dodajte neke nove opcije i sl.

| id | ime | prezime | adresa |
|----|---------|---------|-------------|
| 1 | Vladica | Savic | Moja Adresa |
| 2 | Pera | Peric | Peringrad |
| 3 | ...itd. | | |

3 rows

Ime: Prezime:

Adresa:

Broj telefona:

Gotova web aplikacija! (Slika 15)

Slika 15

Nadam se da sam vam zagolicao malo želju da se upoznate sa ovim programom.

Na velikom broju programerskih foruma može se приметiti i to da je veliki broj Delphi programera odusevljen ovim alatom, dok se naravno može naći i onaj deo PHP programera koji nisu baš impresionirani ovakvim načinom programiranja, ali u svakom slučaju moja je preporuka da probate ovaj alat i da sami odlučite da li vam odgovara.

Par godina sam radio u Delphi-u i tu pravio desktop aplikacije, a od skora sam počeo sa razvojem web aplikacija i mislim da sam našao najbolji alat za to. Pošto mi je Delphi bio prirastao za srce, iskreno nisam mogao da se odvojim od IDE-a (Integrated development environment-a) i da se prepustim nekom od tekst editora, odlucio sam se za Delphi for PHP iz razloga što mislim da je to jedan od alata koji će vrlo brzo da zaživi medju razvojnim alatima te vrste i “preraste ih” u svakom pogledu.

Naravno, neće se svi složiti samnom u vezi toga - to je moje mišljenje, a svako ima pravo na drugačije zar ne ?

Svako dobro svim čitaocima ovog tutorijala. Vladica Savić - [mail: vladicasavic@yahoo.com]

D4PHP – Tutorijal - korak po korak u praksi – Drugi deo – Vladica Savić

Kao što sam vam obećao evo i nastavka za započeti tutorijal o D4PHP-u.

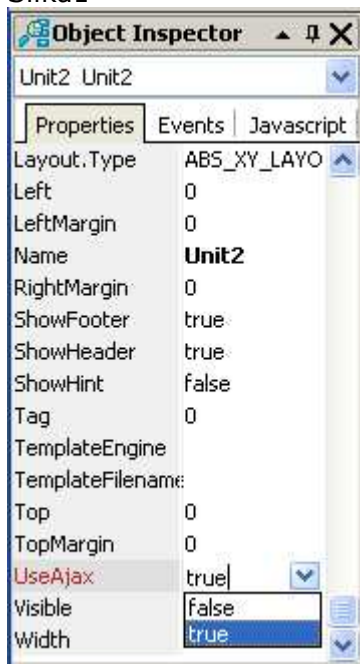
U ovom primeru ćemo se pozabaviti nešto malo Ajax tehnologijom (Ajax je akronim za Asynchronous JavaScript And XML), on inače nije novi programski jezik, već relativno nova tehnika za kreiranje boljih, brzih i interaktivnijih web aplikacija koja se sve češće koristi, a koju ćemo da iskoristimo za unos podataka u bazu. Ajax se izvršava na klijentskoj strani (browseru) i koristi asinhroni prenos podataka između web browsera i web servera, što znači da će naša web aplikacija, koristeći Ajax, moći da prima i da šalje podatke bez stalnog učitavanja cele web strane, što naravno ubrzava rad.

U ovom primeru ćemo koristiti isti način da napravimo novu bazu koristeći phpMyAdmin. Dakle kreiramo bazu ajaxd4php (tako sam ja dao ime svojoj bazi) i unutar nje tabelu tabelaaajax sa poljima id koje je tipa INT koje će imati svojstvo auto_increment i koje će nam ujedno biti i polje primarnog ključa, zatim polje Ime i polje Prezime.

Kada završimo sa tim, na isti način kao i u prethodnom primeru pokrenemo Delphi for PHP i u njemu novu aplikaciju.

Postoće naša aplikacija koristiti Ajax to trebamo i da joj saopštimo. U ObjectInspectoru pronaći ćemo svojstvo UseAjax i postavimo ga na True (Slika 1).

Slika 1



Nakon što to uradimo postavimo na formu dve Label komponente, dve Edit komponente i jedan Button (Slika 2).

Slika 2

Ime:

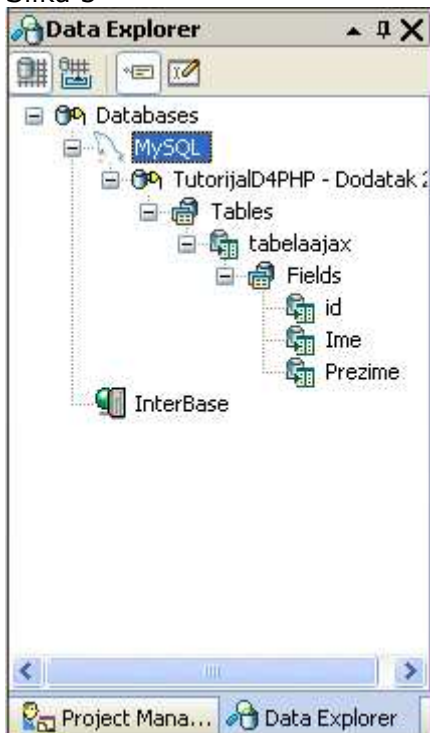
Prezime:

Unesi u bazu!

Edit1 komponenti u koju cemo da unosimo imena promenice ime u EditIME, Edit2 komponenti promenice ime u EditPREZIME, a label komponentama postavice Caption svojstvo u 'Ime:' i u 'Prezime:' a dugmetu 'Unesi u bazu!' kako bi se korisnik lakse orijentisao koje edit polje je za sta sluzi, i dugme za sta sluzi.

Sledeci korak nam je registrovanje baze koju cemo da koristimo (ovaj korak je takodje objasnjen u prethodnom delu). Nakon sto registrujemo bazu, prevucicemo iz DataExplorer dela 'tabelaaajax' tabelu (Slika 3), i u dizajnerskom delu ce se kreirati DBGrid, Table, Database i Datasource komponente koje nam omogucavaju rad sa bazom (Slika 4).

Slika 3



Slika 4

Ime:

Prezime:

0 rows

tbtabelaajax1 dbajaxd4php1 dstabelaajax1

Posto smo završili i ovaj deo preći ćemo na kodiranje. Selektovaćemo dugme za unos podataka u bazu koje smo ranije dodali na formu, i u ObjectInspectoru u delu JavaScript na OnClick događaj dodaćemo sledeći kod:

Kod za dugme za unos podataka na JavaScript OnClick događaj:
Code:

```
function Button1JSClick($sender, $params)
{
    ?>
    //Add your javascript code here

    var Ime = document.getElementById('EditIME');
    var Prezime = document.getElementById('EditPREZIME');

    if (Ime.Value == "" || Prezime == "")
    {
        //Ukoliko neki podatak nije unet necemo zvati ajax
        return(false);
    } else
    {
        var params = [];
        params.push(Ime.value);
        params.push(Prezime.value);
    }
}
```

```

<?php
echo $this->Button1->ajaxCall("UnesiPodatke");
?>
//Vracamo false, forma nece biti submitovana
return(false);
<?php
}

```

Posto ovaj deo koda uzima vrednost iz EditIME i EditPREZIME komponente, proverava da li su uneti podaci i ukoliko jesu prosledjuje vrednosti u params[0] (za ime) i params[1] (za prezime) i poziva funkciju 'UnesiPodatke' preko ajax poziva.

Kod za funkciju 'UnesiPodatke':

Code:

```

/* Funkcija kojom ce se izvršiti umetanje podataka u bazu
ukoliko su Edit komponente popunjene nekim tekstom */
function UnesiPodatke($sender, $params)
{
$this->tbtabelaajax1->Open();
$this->tbtabelaajax1->Append();
$this->tbtabelaajax1->Ime = $params[0];
$this->tbtabelaajax1->Prezime = $params[1];
$this->tbtabelaajax1->Post();
$this->tbtabelaajax1->Close();
$this->tbtabelaajax1->setActive(false);
$this->tbtabelaajax1->setActive(true);
}

```

U ovom delu koda definisemo sta zelimo da se izvrši nad datom bazom. Kao sto vidite ovaj deo koda služi za unos podataka u bazu i u polje Ime se unosi vrednost koju nosi \$params[0] koja predstavlja podatak za ime koje se unosi u bazu, a u polje Prezime se unosi vrednost koju nosi \$params[1] koja predstavlja podatak za prezime koje treba da unesemo, podaci se unesu, baza se zatvori i poslednje dve linije koda prakticno služe za osvezavanje.

Sada sledi samo deo da pokrenemo nasu web aplikaciju i da vidimo da li radi ono sto je ocekivano, tj. da vrsi unos podataka u bazu, ali bez osvezavanja web strane svaki put kad unesemo novi podatak.

Pokrenucemo nasu web aplikaciju i u browseru cemo popuniti formu i klikom na dugme 'Unesi u bazu!' izvrsice se kod koji smo uneli i podaci ce biti uneti u bazu bez ponovnog ucitavanja cele strane (Slika 5).

Slika 5

Ime:







Prezime:

| id | Ime | Prezime |
|----|---------|-----------------|
| 1 | Vladica | Savic |
| 2 | Pera | Peric |
| 3 | itd. | Blablahkovic... |

3 rows

Naravno, provericemo ipak i u bazi da vidimo da li su podaci zaista uneti, otvoricemo phpMyAdmin i proveriti dali u tabeli tabelaaajax u datoj bazi ajaxd4php zaista postoje podaci koje smo uneli. I evo kao sto se na slici ispod vidi (Slika 6) podaci su zaista tu.

Slika 6

| | id | Ime | Prezime |
|--|----|---------|-----------------|
| <input type="checkbox"/>   | 1 | Vladica | Savic |
| <input type="checkbox"/>   | 2 | Pera | Peric |
| <input type="checkbox"/>   | 3 | itd. | Blablahkovic... |

Na isti nacin kako smo unosili podatke u bazu, analogno sa time mogu se vrsiti i druge operacije nad podacima u bazi, editovanje, brisanje i dr.

To bi continued... i svako dobro svima koji su dogurali do kraja i ovog dela.

Vladica Savic

Mail: vladicasavic@yahoo.com

Sajt: <http://vladicasavic.iz.rs>