Dva pristupa ADO.NET podacima

SqlDataReader i SqlDataAdapter na istom primeru

Knjige

povezani pristup

diskonektovani pristup

Isti podaci, dva načina čitanja

Tabela Knjige

Id Naziv Godina Stanje

1 SqlDataReader

Konekcija je otvorena dok čitač prolazi kroz redove.

2 SqlDataAdapter

Adapter preuzima podatke u DataSet, a zatim zatvara konekciju.

Ishodi učenja

- objasniti uloge ADO.NET komponenti
- razlikovati povezani i diskonektovani pristup
- čitati zapise pomoću SqlDataReader-a
- popuniti DataSet pomoću SqlDataAdapter-a
- prikazati oba rezultata u istoj GridView kontroli

ZAJEDNIČKI PRIMER

Organizacija projekta

Datoteka	Odgovornost
Knjiga.cs	Model jednog zapisa iz tabele
Funkcionalnost.cs	Konekcija, upiti i ADO.NET objekti
Default.aspx	Dugmad i GridView kontrola
Default.aspx.cs	Dodela DataSource i poziv DataBind()

GridView ne zna kako su podaci preuzeti. Dobija samo gotov izvor podataka.

Model jednog zapisa

```
public class Knjiga
{
    public int Id { get; set; }
    public string Naziv { get; set; }
    public int Godina { get; set; }
    public bool Stanje { get; set; }
}
```

Jedan objekat

predstavlja jedan red
iz SQL tabele Knjige

List<Knjiga>

predstavlja više redova

Data Provider komponente

Komponenta	Uloga
SqlConnection	Uspostavlja vezu sa SQL Serverom
SqlCommand	Izvršava SQL naredbu
SqlDataReader	Čita rezultat red po red
SqlDataAdapter	Prenosi rezultat u DataSet ili DataTable

SqlDataAdapter pripada Data Provider-u. Poređenje se pravi između pristupa, ne između DataAdapter-a i Data Provider-a.

Povezani pristup

SQL baza — **SqlConnection** — **SqlDataReader** — **List<Knjiga>**

Konekcija mora biti otvorena tokom while (citac.Read()) petlje.

Posle kopiranja zapisa u listu čitač i konekcija mogu da se zatvore.

DataReader metoda: priprema

```
public List<Knjiga> PrikaziReader()
{
    List<Knjiga> knjige = new List<Knjiga>();

    string upit =
        "SELECT Id, Naziv, Godina, Stanje FROM Knjige";

    using (SqlConnection konekcija =
        new SqlConnection(connectionString))
    {
        SqlCommand komanda =
            new SqlCommand(upit, konekcija);

        konekcija.Open();
```

DataReader metoda: čitanje

```
        using (SqlDataReader citac =
            komanda.ExecuteReader())
        {
            while (citac.Read())
            {
                Knjiga knjiga = new Knjiga();

                knjiga.Id = int.Parse(citac["Id"].ToString());
                knjiga.Naziv = citac["Naziv"].ToString();
                knjiga.Godina = int.Parse(citac["Godina"].ToString());
                knjiga.Stanje = bool.Parse(citac["Stanje"].ToString());

                knjige.Add(knjiga);
            }
        }
    }
    return knjige;
}
```

Code-behind ostaje kratak

```
protected void btnReader_Click(object sender, EventArgs e)
{
    Funkcionalnost funkcionalnost =
        new Funkcionalnost(connectionString);

    GridView2.DataSource =
        funkcionalnost.PrikaziReader();

    GridView2.DataBind();
}
```

GridView dobija List<Knjiga>. Čitač i konekcija već su zatvoreni.

Diskonektovani pristup

SQL baza — **SqlDataAdapter** — **DataSet** — **DataTable**

Adapter dobija upit i connection string. Fill() otvara konekciju, preuzima podatke i zatvara je.

DataSet može da sadrži više DataTable tabela i njihove DataRelation veze.

DataAdapter puni DataSet

```
public DataTable PrikaziDataAdapter()
{
    string upit =
        "SELECT Id, Naziv, Godina, Stanje FROM Knjige";

    SqlDataAdapter adapter =
        new SqlDataAdapter(upit, connectionString);

    DataSet podaci = new DataSet();

    adapter.Fill(podaci, "Knjige");

    return podaci.Tables["Knjige"];
}
```

Šta radi Fill metoda?

```
adapter.Fill(podaci, "Knjige");
```

- adapter izvršava SELECT upit
- podaci je DataSet koji se popunjava
- "Knjige" je naziv DataTable objekta u DataSet-u
- Fill otvara i zatvara konekciju kada je ona prethodno zatvorena

Ne puni se adapter. Adapter prenosi podatke u DataSet.

Struktura popunjenog DataSet-a

DataSet podaci

DataTable "Knjige"

DataColumn	Id, Naziv, Godina, Stanje
DataRow	svaka knjiga predstavlja jedan red
DataRelation	veza između dve DataTable tabele

Code-behind ima isti obrazac

```
protected void btnDataAdapter_Click(object sender, EventArgs e)
{
    Funkcionalnost funkcionalnost =
        new Funkcionalnost(connectionString);

    GridView2.DataSource =
        funkcionalnost.PrikaziDataAdapter();

    GridView2.DataBind();
}
```

GridView dobija DataTable izdvojenu iz DataSet-a.

Isti GridView, različit izvor

DataReader

```
GridView2.DataSource =
    funkcionalnost.PrikaziReader();

GridView2.DataBind();
```

List<Knjiga>

DataAdapter

```
GridView2.DataSource =
    funkcionalnost.PrikaziDataAdapter();

GridView2.DataBind();
```

DataTable iz DataSet-a

Poređenje pristupa

Osobina	DataReader	DataAdapter + DataSet
Način rada	povezani	diskonektovani
Konekcija	otvorena tokom čitanja	otvorena samo tokom Fill()
Rezultat u primeru	List<Knjiga>	DataTable
Čitanje	red po red	svi podaci u memoriji
Najbolja primena	jednostavan prikaz	više tabela i relacije
Memorija	manja potrošnja	veća potrošnja

Izbor zavisi od zadatka

Za jednu tabelu i jednostavno čitanje

SqlDataReader + List<Knjiga>

Za više tabela, relacije i obradu u memoriji

SqlDataAdapter + DataSet

Za ovu nastavnu jedinicu prikazujemo oba pristupa jer svaki objašnjava drugu grupu ADO.NET komponenti.

PROVERA RAZUMEVANJA

Samostalni rad

- napraviti tabelu Filmovi: Id, Naziv, Godina i Dostupan
- napisati metodu PrikaziReader() koja vraća List<Film>
- napisati metodu PrikaziDataAdapter() koja vraća DataTable
- prikazati obe metode u istoj GridView kontroli
- objasniti kada je konekcija otvorena u oba primera