

Ajax zahtevi

Uvod u AJAX

- AJAX = Asynchronous JavaScript and XML
- AJAX omogućava parcijalno ažuriranje html strane razmenom malih količina podataka sa serverom
- Ažuriranje strane se vrši asinhrono
- Pri klasičnom zahtevu html strana mora da promeni kompletni sadržaj iako se mali deo strane menja
- AJAX koristi
 - **XMLHttpRequest** objekat za razmenu podataka sa serverom
 - JavaScript/DOM za prikaz podataka
- Svi moderni browser podržavaju XMLHttpRequest obejkat

Slanje zahteva web serveru

```
var zahtev = new XMLHttpRequest();
```

```
zahtev.open("GET", "/Home/Index", true);  
zahtev.send();
```

open(method, url, async)

- method- specificira se tip zahteva GET ili POST
- url – adresa na serveru
- async = true za asinhrono zahteve

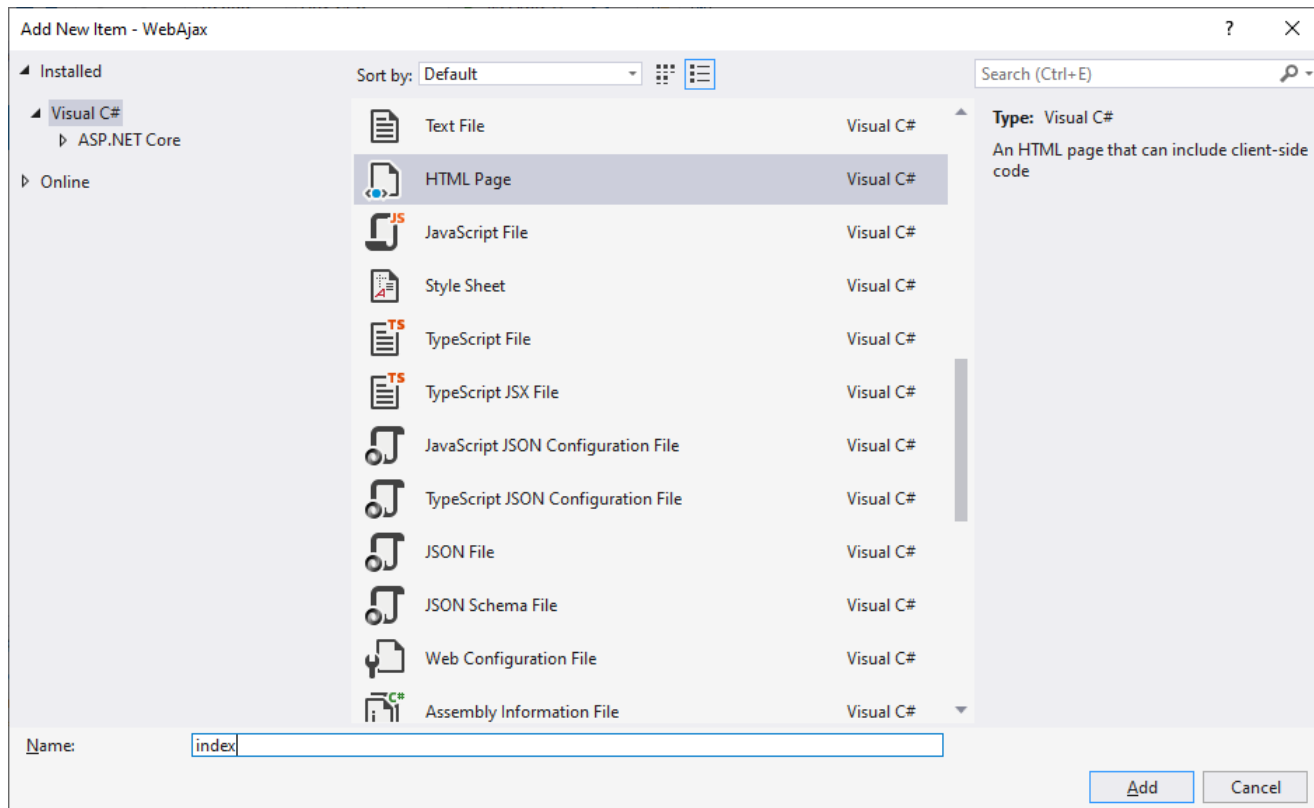
send() - GET

send(*string*) - POST

XMLHttpRequest objekat

- Svojstvo **responseText**, vraća tekst koji je odgovor servera najčešće u JSON formatu
- Svojstvo **responseXML**, odgovor servera se tretira kao XML
- Svojstvo **readystate** čuva stanje zahteva
 - 0-zahtev nije inicijalizovan
 - 1-uspostavljena konekcija sa serverom
 - 2-zahtev primljen
 - 3-obrada zahteva
 - 4-zahtev obrađen i odgovor spreman
- Svojstvo **onreadystatechange** definiše hendlersku funkciju koja se poziva kada se svojstvo **readystate** menja
- Svojstvo **status**
 - 200 – OK
 - 404- "Page not found"

Dodavanje strane index.html folder wwwroot



Strana index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Ajax zahtevi</title>
</head>
<body>
  <h1>Pocetna strana</h1>
</body>
</html>
```

Metoda Configure klase Startup

Postavlja se početna strana sajta index.html

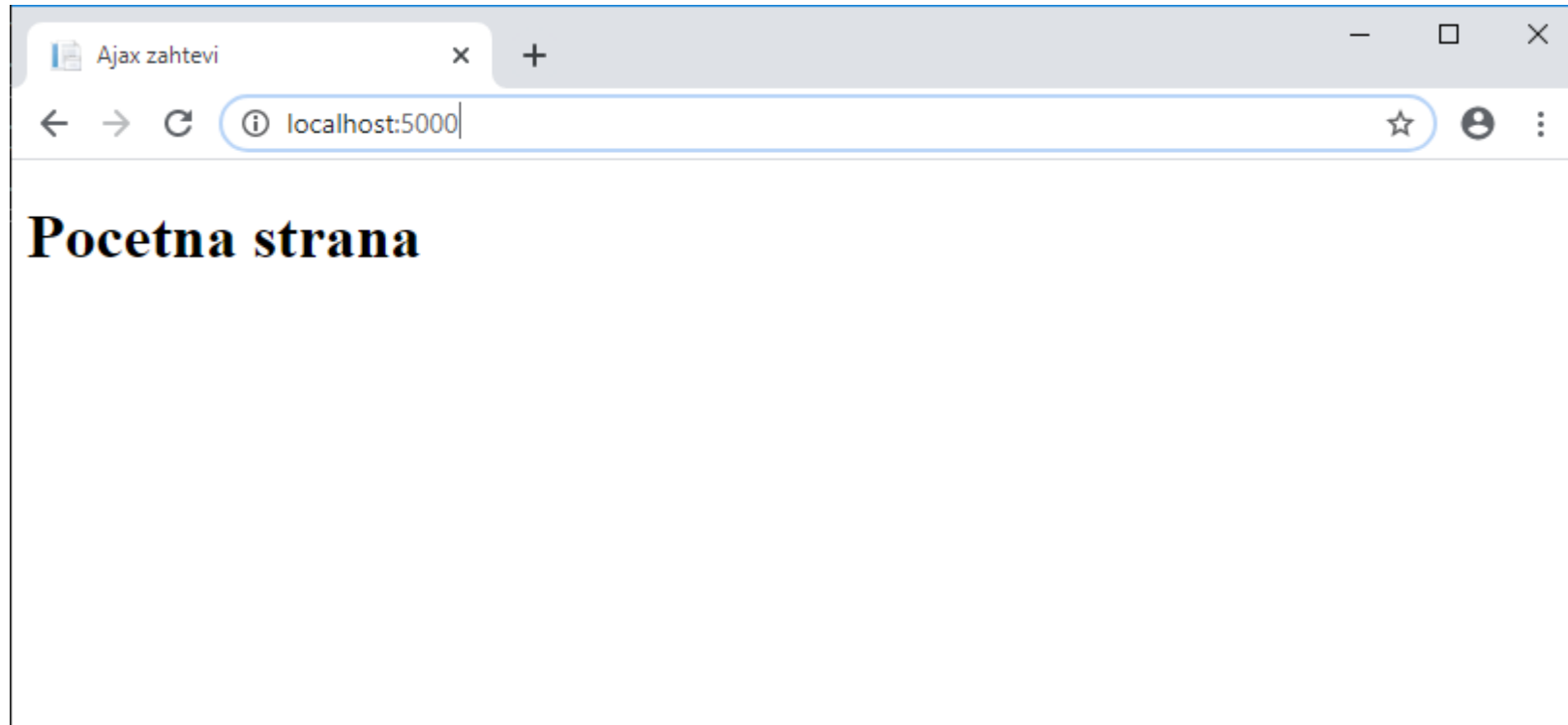
```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
    }
    app.UseDefaultFiles();
    app.UseStaticFiles();

    app.UseRouting();

    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller=Home}/{action=Index}/{id?}");
    });
}
```

Početna strana aplikacije



Model folder klasa Osoba

```
namespace WebAjax.Models
{
    public class Osoba
    {
        public int OsobaId { get; set; }
        public string Ime { get; set; }
        public string Prezime { get; set; }
        public string Telefon { get; set; }
    }
}
```

Klasa HomeController

```
public class HomeController : Controller
{
    public List<Osoba> listaOsoba = new List<Osoba>();

    public HomeController()
    {
        Osoba os1 = new Osoba { OsobaId = 1, Ime = "Janko", Prezime = "Petrovic", Telefon = "123-456" };
        Osoba os2 = new Osoba { OsobaId = 2, Ime = "Mika", Prezime = "Mikic", Telefon = "345-458" };
        Osoba os3 = new Osoba { OsobaId = 3, Ime = "Ivana", Prezime = "Ivanovic", Telefon = "567-785" };
        Osoba os4 = new Osoba { OsobaId = 4, Ime = "Jovana", Prezime = "Jovanovic", Telefon = "789-6756" };

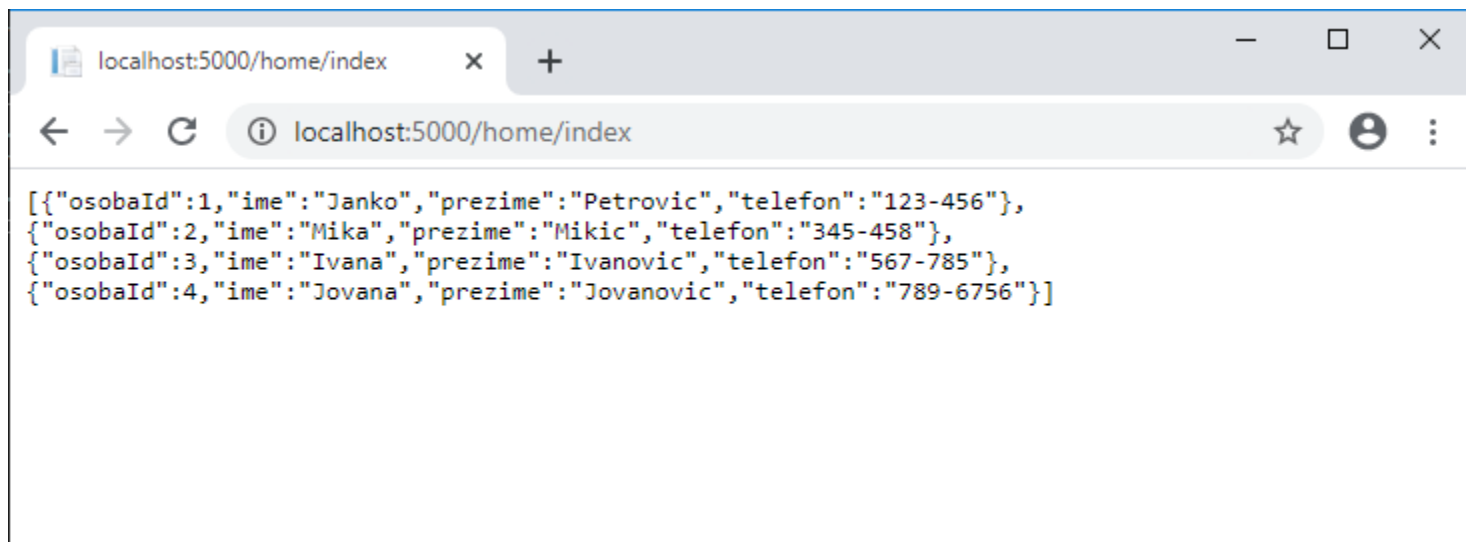
        listaOsoba.Add(os1);
        listaOsoba.Add(os2);
        listaOsoba.Add(os3);
        listaOsoba.Add(os4);
    }
}
```

Metoda Index klase HomeController

```
public JsonResult Index(int id = 0)
{
    if (id == 0)
    {
        return Json(listaOsoba);
    }
    else
    {
        Osoba os = listaOsoba.SingleOrDefault(o => o.OsobaId == id);

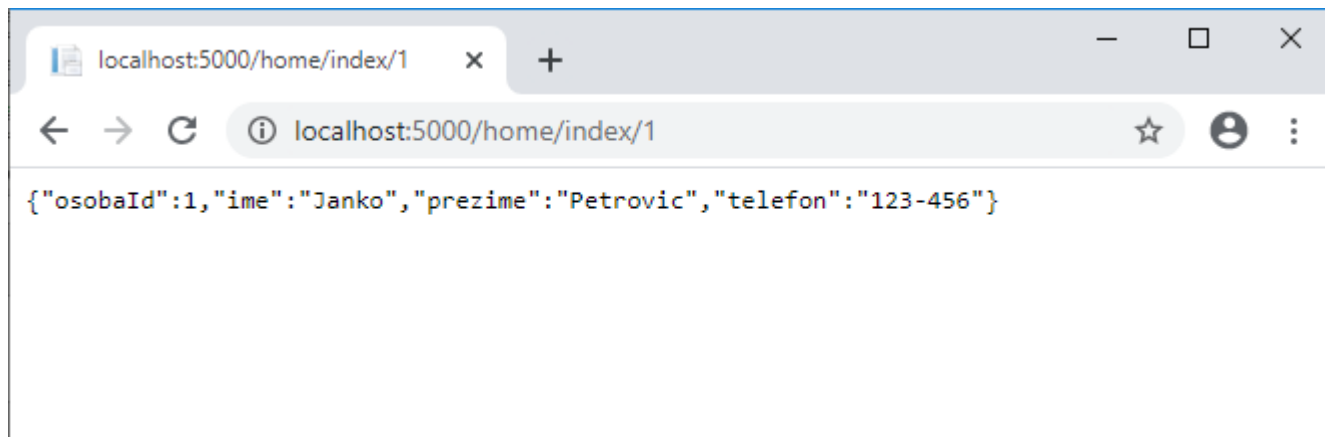
        if (os != null)
        {
            return Json(os);
        }
        else
        {
            return Json(new Osoba());
        }
    }
}
```

Podaci koje generiše server



A screenshot of a web browser window. The address bar shows 'localhost:5000/home/index'. The main content area displays a JSON array of four objects, each representing a person with attributes: osobaId, ime, prezime, and telefon.

```
[{"osobaId":1,"ime":"Janko","prezime":"Petrovic","telefon":"123-456"},  
{ "osobaId":2,"ime":"Mika","prezime":"Mikic","telefon":"345-458"},  
{ "osobaId":3,"ime":"Ivana","prezime":"Ivanovic","telefon":"567-785"},  
{ "osobaId":4,"ime":"Jovana","prezime":"Jovanovic","telefon":"789-6756"}]
```



A screenshot of a web browser window. The address bar shows 'localhost:5000/home/index/1'. The main content area displays a single JSON object representing the first person from the array above.

```
{ "osobaId":1,"ime":"Janko","prezime":"Petrovic","telefon":"123-456" }
```

Strana index.html

```
<body>
  <h1>Pocetna strana</h1>
  <span id="span1"></span>
  <hr>
  <form>
    Unesite id osobe:<input id="Text1" type="text"><br>
    <input id="Button1" type="button" value="Pronadji" onclick="PronadjiOsobu()">
  </form>
  <br />
  <span id="span2"></span>

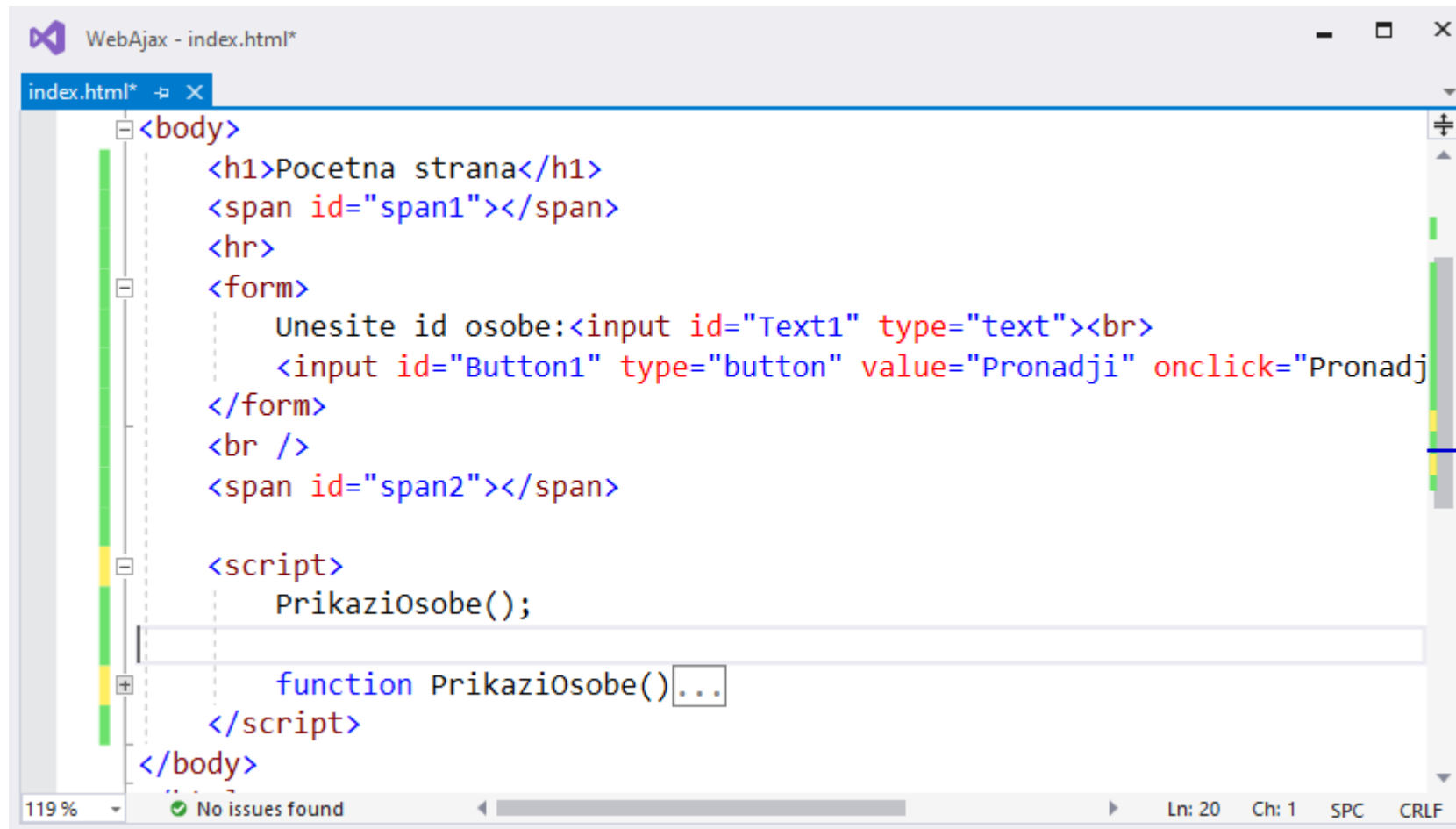
  <script>

  </script>
</body>
```

Javascript funkcija koja generiše ajax zahtev -1

```
function PrikaziOsobe() {  
    // pristup objektu modelu html elementa  
    var span1 = document.getElementById("span1");  
    var zahtev = new XMLHttpRequest();  
  
    zahtev.open("GET", "/Home/Index", true);  
  
    zahtev.onreadystatechange = function () {  
        if (this.readyState == 4 && this.status == 200) {  
            var osobe = JSON.parse(this.responseText);  
  
            var s = "<ul>";  
  
            for (var i = 0; i < osobe.length; i++) {  
                s += `<li>  
                    ${osobe[i].ime}  
                    ${osobe[i].prezime}  
                </li>`;  
            }  
            s += "</ul>";  
  
            span1.innerHTML = s;  
        }  
    };  
    zahtev.send();  
}
```

Generisanje AJAX zahteva pri učitavanju strane



The screenshot shows a code editor window titled "WebAjax - index.html*". The code is as follows:

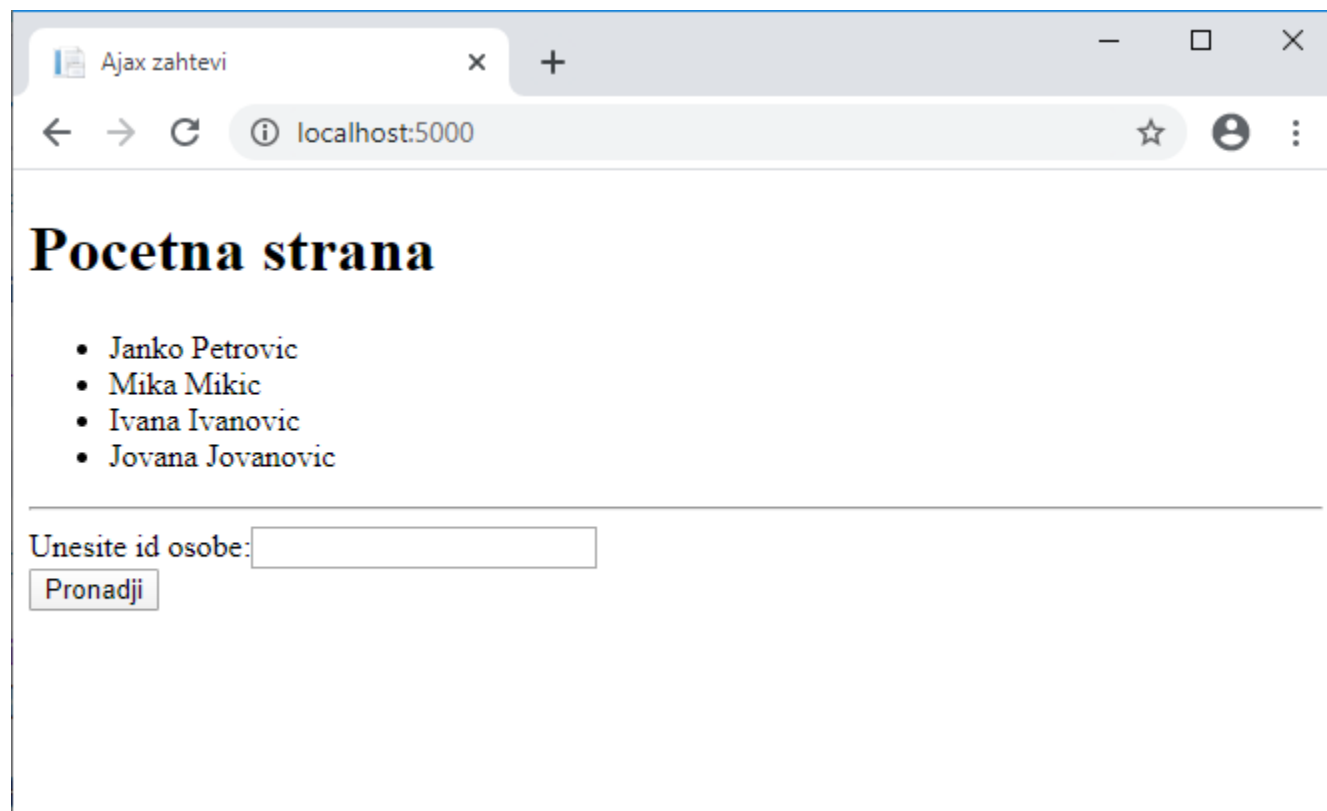
```
<body>
  <h1>Pocetna strana</h1>
  <span id="span1"></span>
  <hr>
  <form>
    Unesite id osobe:<input id="Text1" type="text"><br>
    <input id="Button1" type="button" value="Pronadji" onclick="Pronadji"
  </form>
  <br />
  <span id="span2"></span>

  <script>
    PrikaziOsobe();

    function PrikaziOsobe()...
  </script>
</body>
```

The status bar at the bottom indicates "119%", "No issues found", and "Ln: 20 Ch: 1 SPC CRLF".

Podaci na strani index.html



Prosleđivanje podataka ajax zahteva callback funkciji

- Podatke dobijene ajax zahtevom potrebno je proslediti nekoj funkciji
- Callback funkcija je funkcija koja se prosleđuje kao parametar drugoj funkciji

```
function CitajPodatke(CallBackFunkcija) {  
    var span1 = document.getElementById("span1");  
    var zahtev = new XMLHttpRequest();  
  
    zahtev.open("GET", "/Home/Index", true);  
  
    zahtev.onreadystatechange = function () {  
  
        if (this.readyState == 4 && this.status == 200) {  
  
            var osobe = JSON.parse(zahtev.responseText);  
            CallBackFunkcija(osobe);  
        }  
    };  
  
    zahtev.send();  
}
```

Primer callback funkcije

```
function PrikaziOsobe1(osobe) {  
    var s = "<ul>";  
  
    for (var i = 0; i < osobe.length; i++) {  
        s += `<li>  
        ${osobe[i].ime}  
        ${osobe[i].prezime}  
        </li>`;  
    }  
    s += "</ul>";  
  
    span1.innerHTML = s;  
}
```

```
<script>  
...  
CitajPodatke(PrikaziOsobe1);  
</script>
```

Javascript funkcija PronadjiOsobu

```
function PronadjiOsobu() {  
  
    var span2 = document.getElementById("span2");  
    var Text1 = document.getElementById("Text1");  
  
    var id = Text1.value;  
    if (isNaN(id)) {  
        alert("Unesite broj");  
        span2.innerHTML = "";  
        Text1.value = "";  
        return;  
    }  
  
    var zahtev = new XMLHttpRequest();  
    zahtev.open("GET", `/Home/Index/${id}`, true);  
    zahtev.send();  
  
    ....  
}
```

```
zahtev.onreadystatechange = function () {  
  
    if (this.readyState == 4 && this.status == 200)  
    {  
  
        var osoba = JSON.parse(this.responseText);  
  
        if (osoba == null) {  
            span2.innerHTML = "Nije pronadjena osoba";  
        }  
        else {  
            span2.innerHTML = `  
                Id: ${osoba.osobaId}<br>  
                Ime: ${osoba.ime}<br>  
                Prezime: ${osoba.prezime}<br>  
                Telefon: ${osoba.telefon}  
            `;  
        }  
    }  
};
```

Rezultat pretrage



Ajax zahtevi x +

localhost:5000

Pocetna strana

- Janko Petrovic
- Mika Mikic
- Ivana Ivanovic
- Jovana Jovanovic

Unesite id osobe:

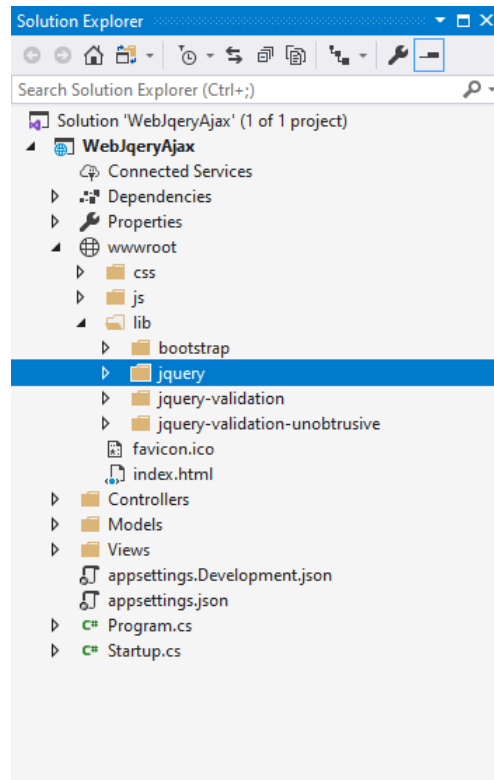
Id: 1
Ime: Janko
Prezime: Petrovic
Telefon: 123-456

jQuery poddržka za ajax

\$.get() metoda

- \$.get() metoda se koristi za asinhrono GET zahteve ka serveru
- Get zahtevi se keširaju i podaci uzeti GET zahtevom se ne menjaju
- Vraća jqXHR objekat koji nadograđuje funkcionalnost XMLHttpRequest objekta

Instalirane biblioteke pri kreiranju ASP.NET Core MVC web aplikacije



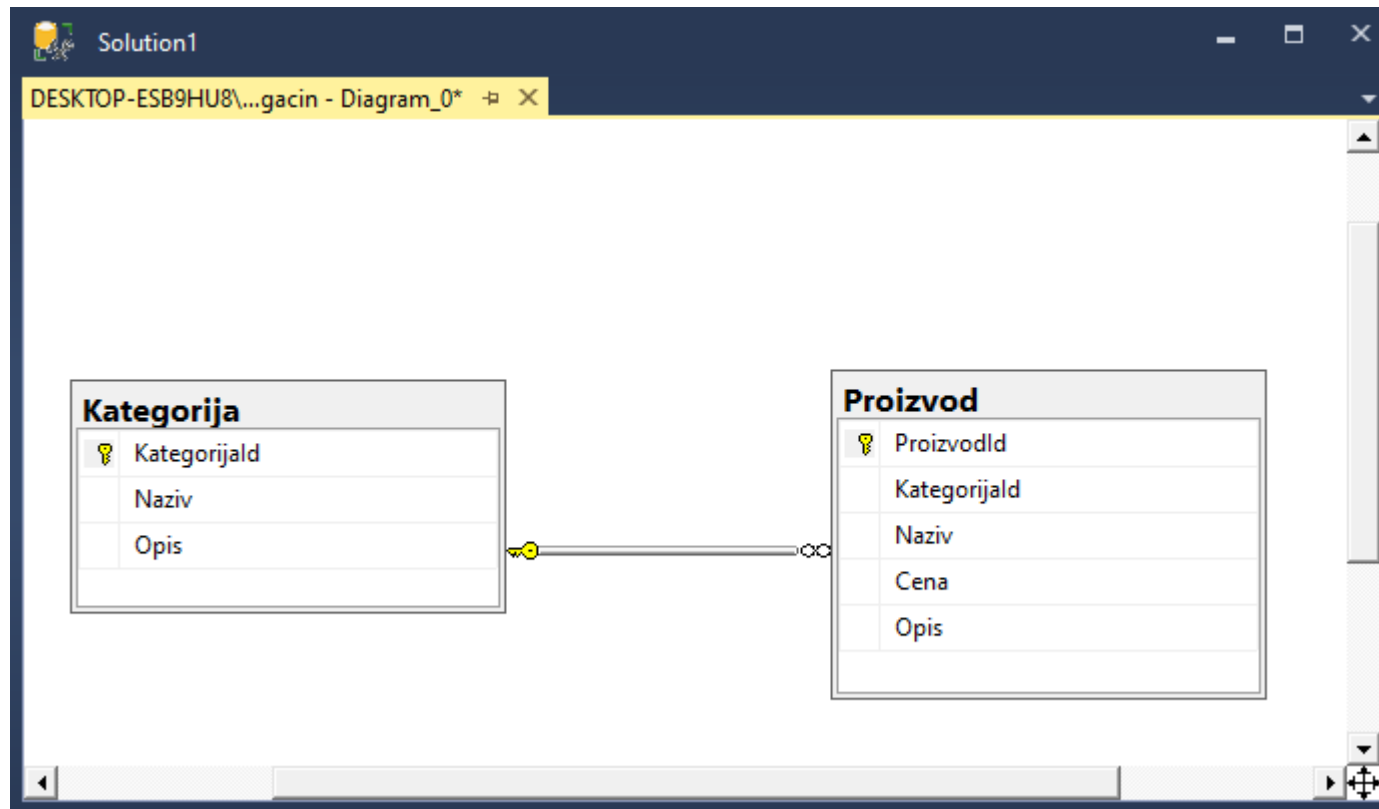
Metoda \$.getJSON(uri)

Učitavanje JSON podataka sa servera korišćenjem AJAX GET zahteva

```
var zahtev = $.getJSON("/Home/VratiProizvode");
```

```
var zahtev = $.getJSON("/Home/VratiProizvode/" + id);
```


Baza podataka



Primenom alata EF Core Power Tools kreirati model podataka na osnovu baze Magacin

Models folder klasa Proizvod

```
[Table("Proizvod")]
public partial class Proizvod
{
    [Key]
    public int ProizvodId { get; set; }
    public int KategorijaId { get; set; }
    [Required]
    [StringLength(120)]
    public string Naziv { get; set; }
    [Column(TypeName = "decimal(10, 2)")]
    public decimal Cena { get; set; }
    [StringLength(120)]
    public string Opis { get; set; }

    [JsonIgnore]
    [ForeignKey(nameof(KategorijaId))]
    [InverseProperty("Proizvodi")]
    public virtual Kategorija Kategorija { get; set; }
}
```

Model klasa Kategorija

```
[Table("Kategorija")]
public partial class Kategorija
{
    public Kategorija()
    {
        Proizvodi = new HashSet<Proizvod>();
    }

    [Key]
    public int KategorijaId { get; set; }
    [Required]
    [StringLength(70)]
    public string Naziv { get; set; }
    [StringLength(120)]
    public string Opis { get; set; }

    [InverseProperty(nameof(Proizvod.Kategorija))]
    public virtual ICollection<Proizvod> Proizvodi { get; set; }
}
```

DbContext klasa

```
public partial class MagacinContext : DbContext
{

    public MagacinContext(DbContextOptions<MagacinContext> options)
        : base(options)
    {
    }

    public virtual DbSet<Kategorija> Kategorije { get; set; }
    public virtual DbSet<Proizvod> Proizvodi { get; set; }

}
```

appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=.\SqlExpress;Initial Catalog=Magacin;Integrated Security=True"
  }
}
```

Registracija DbContext klase kao servisa

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<MagacinContext>(opcije =>
        opcije.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));
    services.AddControllersWithViews();
}
```

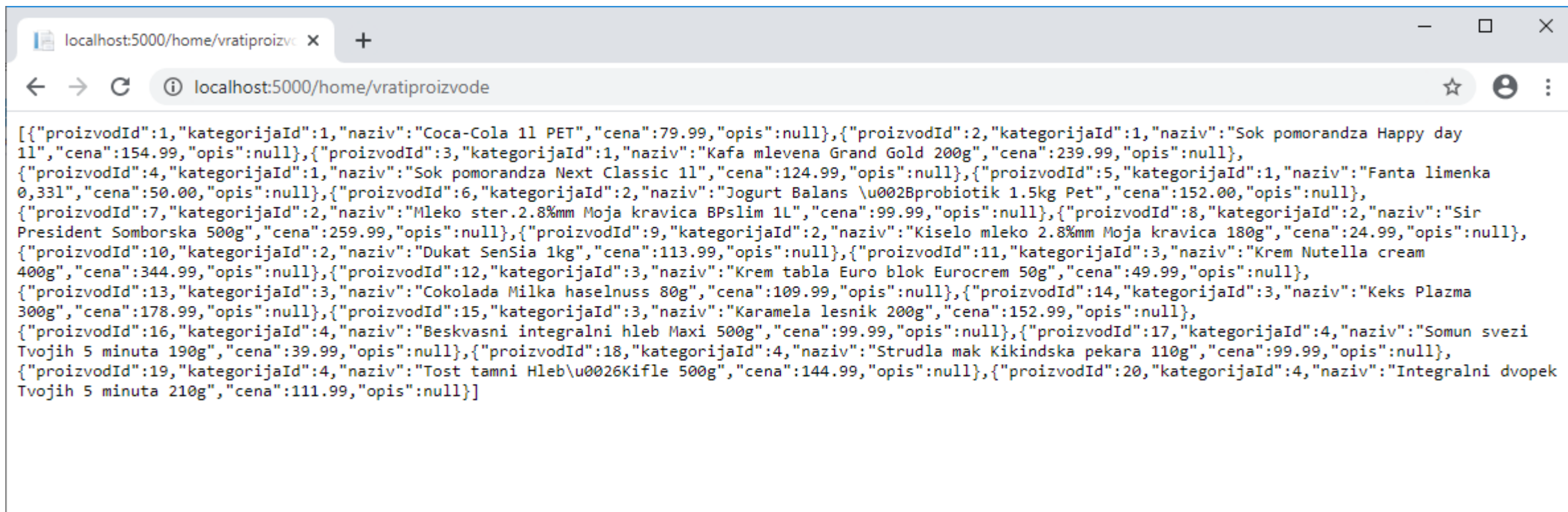
Metoda VratiProizvode() klasa klasa HomeController

```
public class HomeController : Controller
{
    private readonly MagacinContext db;

    public HomeController(MagacinContext _db)
    {
        db = _db;
    }
    public JsonResult VratiProizvode(int id =0)
    {
        if (id != 0)
        {
            Proizvod p1 = db.Proizvodi.Find(id);

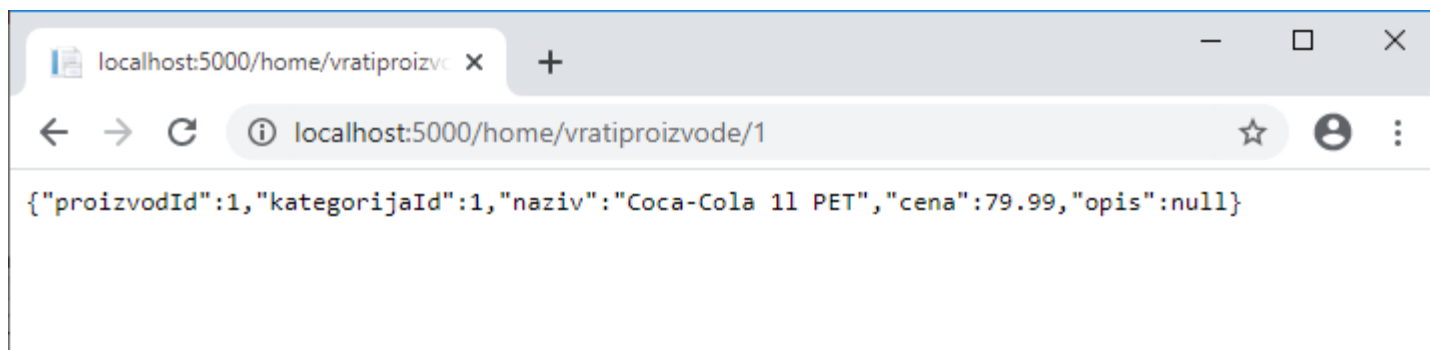
            if (p1 != null)
            {
                return Json(p1);
            }
            else
            {
                return Json(new Proizvod());
            }
        }
        else
        {
            return Json(db.Proizvodi.ToList());
        }
    }
}
```

Poziv serverske metode VratiProizvode



A screenshot of a web browser window. The address bar shows the URL `localhost:5000/home/vratiproizvode`. The main content area displays a large JSON array of product objects. Each object contains fields for `proizvodId`, `kategorijaId`, `naziv`, `cena`, and `opis`. The products listed include Coca-Cola 1l PET, Sok pomorandza Happy day 1l, Kafa mlevena Grand Gold 200g, Sok pomorandza Next Classic 1l, Fanta limenka 0,33l, Jogurt Balans \u002Bprobiotik 1.5kg Pet, Mleko ster.2.8%mm Moja kravica BPslim 1l, Sir President Somborska 500g, Kiselo mleko 2.8%mm Moja kravica 180g, Dukat SenSia 1kg, Krem Nutella cream 400g, Krem tabla Euro blok Eurocrem 50g, Cokolada Milka haselnuss 80g, Keks Plazma 300g, Karamela lesnik 200g, Beskvasni integralni hleb Maxi 500g, Somun svezi Tvojih 5 minuta 190g, Strudla mak Kikindska pekara 110g, Tost tamni Hleb\u0026Kifle 500g, and Integralni dvopek Tvojih 5 minuta 210g.

```
[{"proizvodId":1,"kategorijaId":1,"naziv":"Coca-Cola 1l PET","cena":79.99,"opis":null}, {"proizvodId":2,"kategorijaId":1,"naziv":"Sok pomorandza Happy day 1l","cena":154.99,"opis":null}, {"proizvodId":3,"kategorijaId":1,"naziv":"Kafa mlevena Grand Gold 200g","cena":239.99,"opis":null}, {"proizvodId":4,"kategorijaId":1,"naziv":"Sok pomorandza Next Classic 1l","cena":124.99,"opis":null}, {"proizvodId":5,"kategorijaId":1,"naziv":"Fanta limenka 0,33l","cena":50.00,"opis":null}, {"proizvodId":6,"kategorijaId":2,"naziv":"Jogurt Balans \u002Bprobiotik 1.5kg Pet","cena":152.00,"opis":null}, {"proizvodId":7,"kategorijaId":2,"naziv":"Mleko ster.2.8%mm Moja kravica BPslim 1l","cena":99.99,"opis":null}, {"proizvodId":8,"kategorijaId":2,"naziv":"Sir President Somborska 500g","cena":259.99,"opis":null}, {"proizvodId":9,"kategorijaId":2,"naziv":"Kiselo mleko 2.8%mm Moja kravica 180g","cena":24.99,"opis":null}, {"proizvodId":10,"kategorijaId":2,"naziv":"Dukat SenSia 1kg","cena":113.99,"opis":null}, {"proizvodId":11,"kategorijaId":3,"naziv":"Krem Nutella cream 400g","cena":344.99,"opis":null}, {"proizvodId":12,"kategorijaId":3,"naziv":"Krem tabla Euro blok Eurocrem 50g","cena":49.99,"opis":null}, {"proizvodId":13,"kategorijaId":3,"naziv":"Cokolada Milka haselnuss 80g","cena":109.99,"opis":null}, {"proizvodId":14,"kategorijaId":3,"naziv":"Keks Plazma 300g","cena":178.99,"opis":null}, {"proizvodId":15,"kategorijaId":3,"naziv":"Karamela lesnik 200g","cena":152.99,"opis":null}, {"proizvodId":16,"kategorijaId":4,"naziv":"Beskvasni integralni hleb Maxi 500g","cena":99.99,"opis":null}, {"proizvodId":17,"kategorijaId":4,"naziv":"Somun svezi Tvojih 5 minuta 190g","cena":39.99,"opis":null}, {"proizvodId":18,"kategorijaId":4,"naziv":"Strudla mak Kikindska pekara 110g","cena":99.99,"opis":null}, {"proizvodId":19,"kategorijaId":4,"naziv":"Tost tamni Hleb\u0026Kifle 500g","cena":144.99,"opis":null}, {"proizvodId":20,"kategorijaId":4,"naziv":"Integralni dvopek Tvojih 5 minuta 210g","cena":111.99,"opis":null}]
```



A screenshot of a web browser window. The address bar shows the URL `localhost:5000/home/vratiproizvode/1`. The main content area displays a single JSON object representing the first product from the array above.

```
{"proizvodId":1,"kategorijaId":1,"naziv":"Coca-Cola 1l PET","cena":79.99,"opis":null}
```


Korisnički interfejs strane

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>jQuery Ajax</title>
</head>
<body>
  <h1>Svi proizvodi</h1>
  <ul id="ul1">
  </ul>
  <hr />
  <form>
    <label for="Number1">Unesite Id:</label>
    <input type="number" id="Number1" name="Number1"> <br><br>

    <input id="Button1" type="button" value="Pronadji" onclick="Pronadji()" />
  </form>

  <p id="p1">

  </p>

  <script src="lib/jquery/dist/jquery.min.js"></script>

</body>
</html>
```

Generisanje asinhronog get zahteva

```
<script>
  function PrikaziProizvode() {

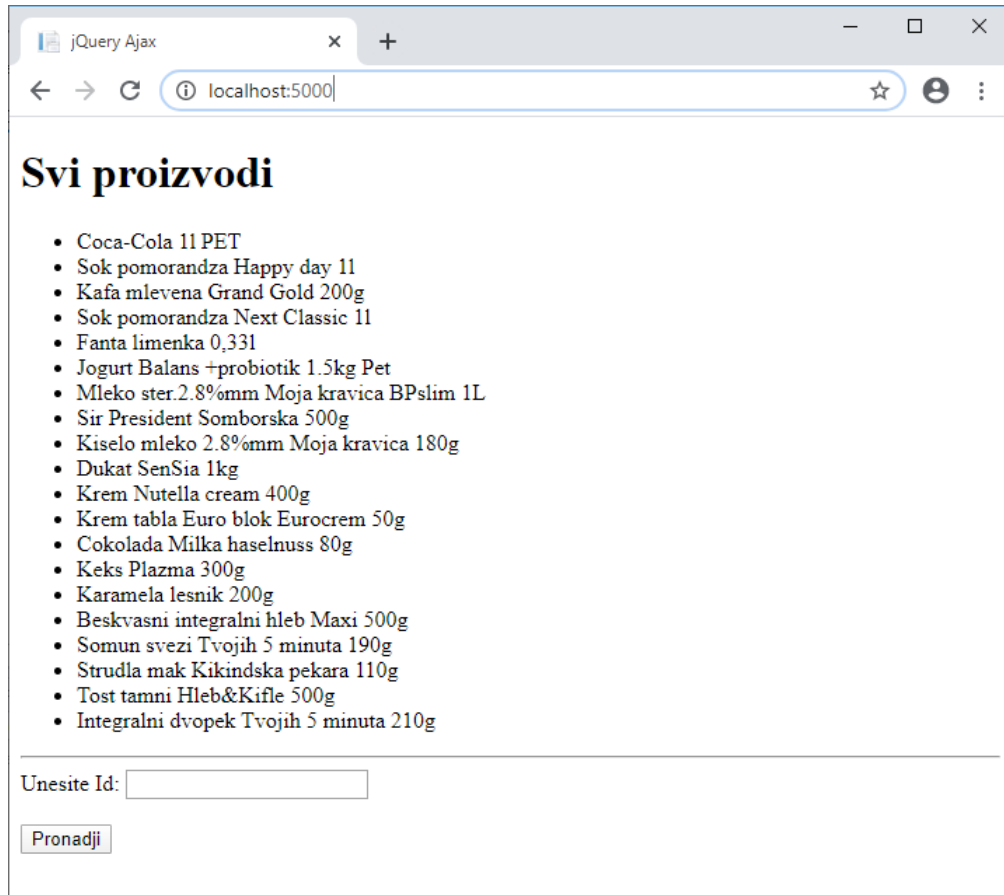
    var ul1 = $("#ul1");
    var zahtev = $.getJSON("/Home/VratiProizvode");

    zahtev.done(function (proizvodi) {
      for (const i in proizvodi) {
        ul1.append(`<li>${proizvodi[i].naziv}</li>`);
      }
    });

    zahtev.fail(function (gr) {
      $("#p1").text(gr.statusText);
    });
  }

  $(function () {
    PrikaziProizvode();
  });
</script>
```

Rezultat zahteva



The screenshot shows a web browser window with the title 'jQuery Ajax' and the address bar containing 'localhost:5000'. The main content area displays the heading 'Svi proizvodi' followed by a list of 20 items. At the bottom, there is a search input field labeled 'Unesite Id:' and a 'Pronadji' button.

Svi proizvodi

- Coca-Cola 1l PET
- Sok pomorandza Happy day 1l
- Kafa mlevena Grand Gold 200g
- Sok pomorandza Next Classic 1l
- Fanta limenka 0,33l
- Jogurt Balans +probiotik 1.5kg Pet
- Mleko ster.2.8%mm Moja kravica BPslim 1L
- Sir President Somborska 500g
- Kiselo mleko 2.8%mm Moja kravica 180g
- Dukat SenSia 1kg
- Krem Nutella cream 400g
- Krem tabla Euro blok Eurocrem 50g
- Cokolada Milka haselnuss 80g
- Keks Plazma 300g
- Karamela lesnik 200g
- Beskvasni integralni hleb Maxi 500g
- Somun svezi Tvojih 5 minuta 190g
- Strudla mak Kikindska pekara 110g
- Tost tamni Hleb&Kifle 500g
- Integralni dvopek Tvojih 5 minuta 210g

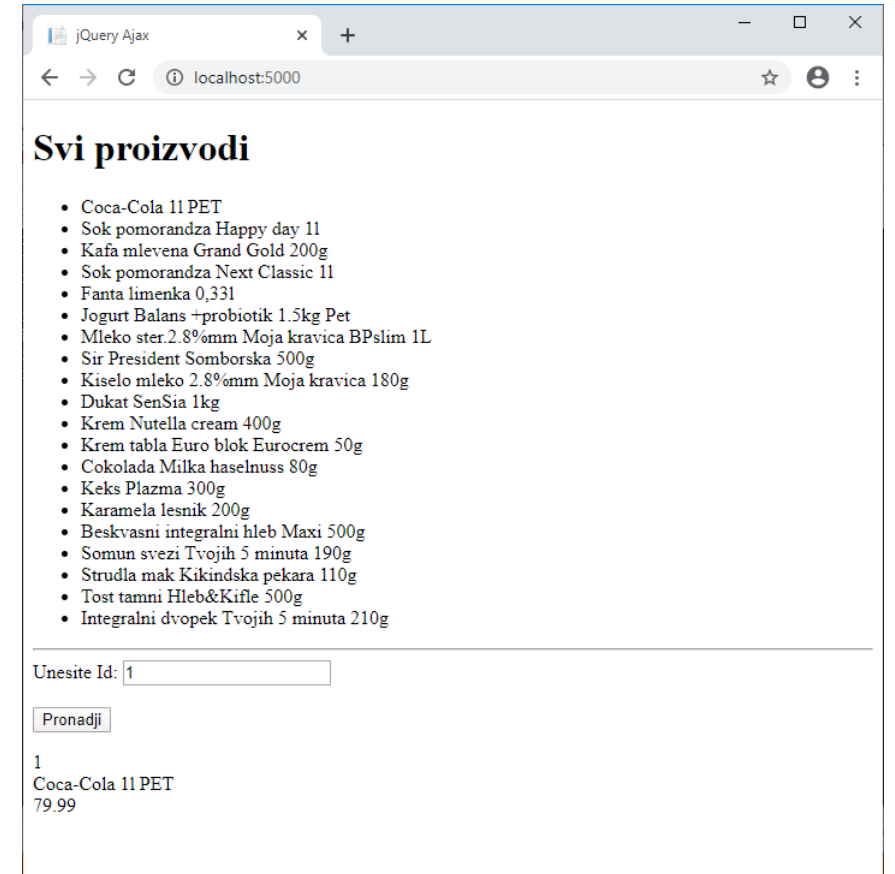
Unesite Id:

Pronadji

Prosleđivanje parametara \$.get() metodi

```
function Pronadji() {
    var id = $("#Number1").val();
    if (isNaN(id)) {
        alert("Unesite broj");
        return;
    }
    var zahtev = $.getJSON("/Home/VratiProizvode/" + id);
    zahtev.done(function (proizvod) {

        if (proizvod.proizvodId != 0) {
            $("#p1").html(
                `${proizvod.proizvodId} <br>
                ${proizvod.naziv}<br>
                ${proizvod.cena}`
            );
        }
        else {
            $("#p1").html("Ne postoji proizvod za dati id");
        }
    });
    zahtev.fail(function (gr) {
        $("#p1").html(gr.statusText);
    });
}
```



The screenshot shows a web browser window with the title "jQuery Ajax" and the address bar "localhost:5000". The page content is as follows:

Svi proizvodi

- Coca-Cola 1l PET
- Sok pomorandza Happy day 1l
- Kafa mlevena Grand Gold 200g
- Sok pomorandza Next Classic 1l
- Fanta limenka 0,33l
- Jogurt Balans +probiotik 1.5kg Pet
- Mleko ster.2.8%mm Moja kravica BPslim 1L
- Sir President Somborska 500g
- Kiselo mleko 2.8%mm Moja kravica 180g
- Dukat SenSia 1kg
- Krem Nutella cream 400g
- Krem tabla Euro blok Eurocrem 50g
- Cokolada Milka haselnuss 80g
- Keks Plazma 300g
- Karamela lesnik 200g
- Beskvasni integralni hleb Maxi 500g
- Somun svezi Tvojih 5 minuta 190g
- Strudla mak Kikindska pekara 110g
- Tost tamni Hleb&Kifle 500g
- Integralni dvopek Tvojih 5 minuta 210g

Unesite Id:

1
Coca-Cola 1l PET
79.99

Metoda \$.ajax()

- Izvršava asinhroni http AJAX zahtev
- Metodi \$.ajax() prosleđuje se java script objekat koji sadrži name, value parove
- ključ **type** specificira tip zahteva (podrazumevano GET)
- ključ **url** specificira url adresu zahteva
- ključ **data** specificira podatke koji se prosleđuju do servera
- ključ **dataType** specificira tip podataka koje očekujemo od servera
- ključ **contentType** specificira tip podataka koje šaljemo ka serveru

Metoda \$.ajax()-2

- Metoda \$.ajax() vraća **jqXHR** objekat koji u sebi sadrži ugrađeni **XMLHttpRequest** objekat:

```
var zahtev = $.ajax({  
    ....  
});
```

- **zahtev.done**(function(podaci){...}) –testiranje uspešnost ajax zahteva kada se podaci prosleđuju definisanoj callback anonimnoj funkciji
- **zahtev.fail**(function (xhr, status, error) {...}) – anonimna funkcija koja se izvršava u slučaju greške pri izvršavanju ajax zahteva

Metoda Vratikategorije() HomeController

```
public JsonResult Vratikategorije()
{
    return Json(db.Kategorije.ToList());
}
```

Post metoda HomeController

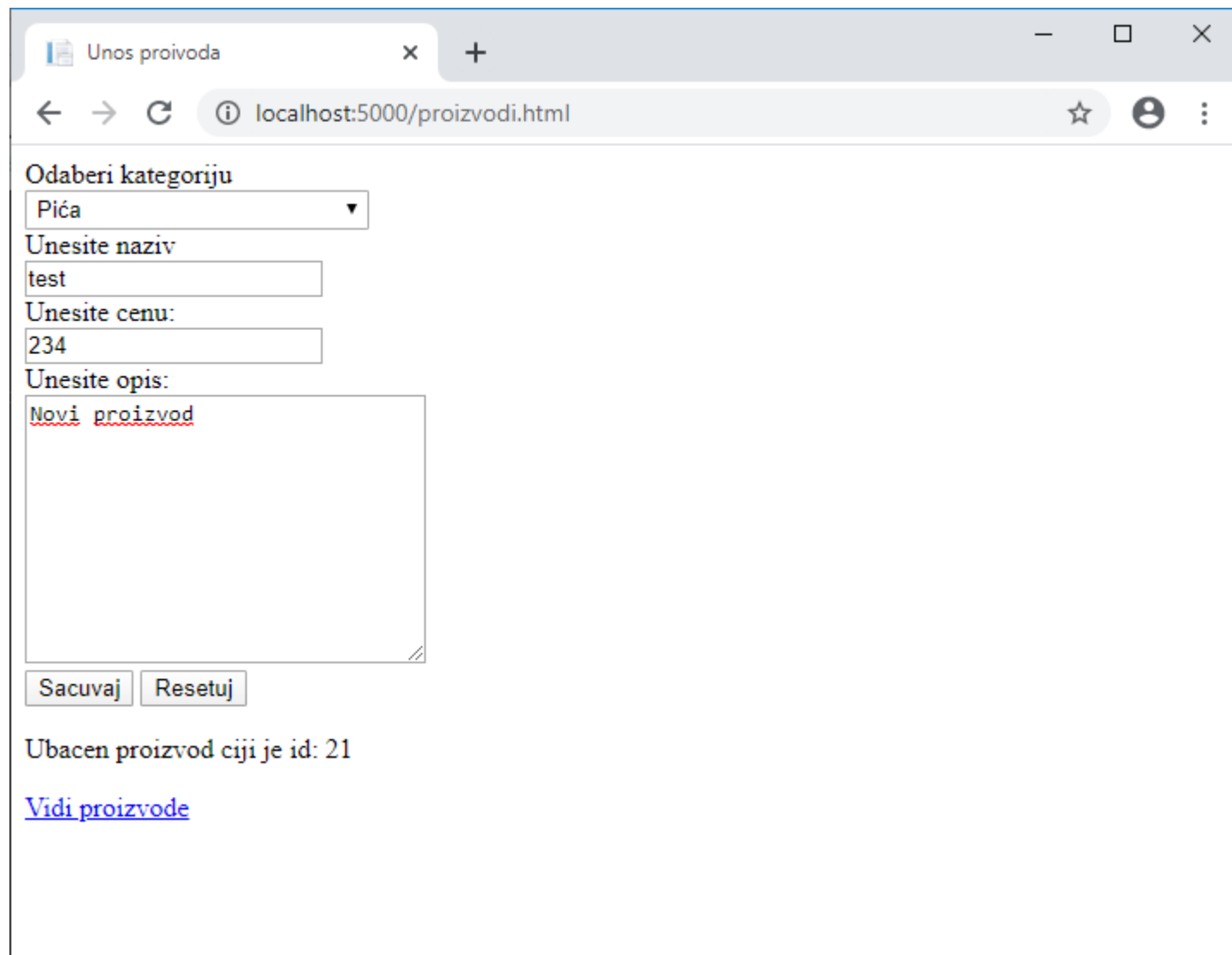
```
[HttpPost]
public int UbaciProizvod(Proizvod p)
{
    try
    {
        db.Add(p);
        db.SaveChanges();
        return p.ProizvodId;
    }
    catch (Exception)
    {
        return -1;
    }
}
```


Prosleđivanje name-value parova

```
var podaci = $("#form1").serialize();  
  
var zahtev = $.ajax({  
    type: "POST",  
    url: "/Home/UbaciProizvod",  
    data: podaci,  
    dataType: "text"  
});
```

```
contentType: "application/x-www-form-urlencoded; charset=UTF-8"
```

Korisnički interfejs



Unos proizvoda

localhost:5000/proizvodi.html

Odaberi kategoriju

Pića

Unesite naziv

test

Unesite cenu:

234

Unesite opis:

Novi proizvod

Sacuvaj Resetuj

Ubacen proizvod ciji je id: 21

[Vidi proizvode](#)

Html forma

```
<form id="form1" action="" method="get">
  <label for="Select1">Odaberi kategoriju</label> <br>
  <select name="KategorijaId" id="Select1">

  </select> <br>

  <label for="TextNaziv">Unesite naziv</label> <br>
  <input type="text" name="Naziv" id="TextNaziv" required> <br>

  <label for="TextCena">Unesite cenu:</label> <br>
  <input type="text" name="Cena" id="TextCena" required> <br>

  <label for="TextOpis">Unesite opis:</label> <br>
  <textarea name="Opis" id="TextOpis" cols="30" rows="10" required></textarea>
  <br>

  <input type="button" value="Sacuvaj" onclick="Sacuvaj()">
  <input type="reset" value="Resetuj">
</form>
<p id="p1">

</p>
<a href="/index.html">Vidi proizvode</a>

<script src="/lib/jquery/dist/jquery.min.js"></script>
<script src="/lib/jquery-validation/dist/jquery.validate.min.js"></script>
```

jQuery biblioteka za validaciju

```
<script src="/lib/jquery/dist/jquery.min.js"></script>  
<script src="/lib/jquery-validation/dist/jquery.validate.min.js"></script>
```

Prikaz kategorija u select elementu

```
function PrikaziKategorije() {  
  
    var zahtev = $.getJSON("/Home/VratiKategorije");  
  
    zahtev.done(function (kategorije) {  
  
        for (const i in kategorije) {  
            var opt = $("<option></option>");  
            opt.text(kategorije[i].naziv);  
            opt.val(kategorije[i].kategorijaId);  
  
            $("#Select1").append(opt);  
        }  
    });  
  
    zahtev.fail(function (grr) {  
        $("#p1").html(grr.statusText);  
    });  
  
}
```

```
$(function () {  
    PrikaziKategorije();  
});
```

Prosleđivanje name-value parova-1

```
function Sacuvaj() {  
    $("#form1").validate();  
  
    if ( $("#form1").valid() == false) {  
        return;  
    }  
  
    var podaci = $("#form1").serialize();  
  
    var zahtev = $.ajax({  
        type:"POST",  
        url:"/Home/UbaciProizvod",  
        data: podaci,  
        dataType: "text"  
    });  
    ...  
}
```

Prosleđivanje name-value parova-2

```
zahtev.done(function (id) {  
  
    if (id > 0) {  
        $("#p1").html("Ubacen proizvod ciji je id: " + id);  
    }  
    else{  
        $("#p1").html("Greska pri unosu proizvoda");  
    }  
});  
  
zahtev.fail(function (grr) {  
    $("#p1").html(grr.statusText);  
});
```

Pitanje 1

Za generisanje AJAX zahteva ka web serveru koristi se sledeći javascript objekat:

- a. XMLHttpRequest
- b. AjaxRequest
- c. Ajax

Odgovor: a

Pitanje 2

Za generisanje GET zahteva ka web serveru kojim se od servera dobijaju podaci u JSON formatu koristi se jquery funkcija

- a. \$.getJSONText()
- b. \$.Json()
- c. \$.getJSON()

Odgovor: c

Pitanje 3

Kako se kod `$.ajax()` jquery metode specificira tip podataka koji se prosleđuju web serveru

- a. pomoću ključa `contentType` objekta koji se prosleđuje metodi
- b. pomoću ključa `dataType` objekta koji se prosleđuje metodi
- c. pomoću ključa `type` objekta koji se prosleđuje metodi

Odgovor: a