Dezvoltarea Aplicatiilor Web-Anul 3
Laborator 7

Exercitiul 1

Sa se adauge controller-ul ArticleController in care se vor implementa urmatoarele:

- Sa se adauge un model numit Article care sa contina Id, Title, Content si Date astfel: Models -> click dreapta -> Add -> Class -> Adaugam o clasa Article.cs

```
public class Article
{
        public int Id { get; set; }
        public string Title { get; set; }
        public string Content { get; set; }
        public DateTime Date { get; set; }
}
```

- Sa se adauge o metoda NonAction care va returna un array de obiecte de tip Article, array pe care o sa il folosim pentru afisare (procedam astfel deoarece in acest laborator nu o sa folosim baza de date, iar in acest mod ne cream articole pe care le putem prelucra)

```
public Article[] GetArticles()
{
        // Instantiem un array de articole
        Article[] articles = new Article[3];
        // Cream un articol
        for (int i = 0; i < 3; i++)
        {
            Article articol = new Article();
            articol.Id = i;
            articol.Title = "Articol " + (i + 1).ToString();
            articol.Content = "Continut articol " + (i + 1).ToString();
            articol.Date = DateTime.Now;
            // Il adaugam in array
            articles[i] = articol;
        }
        return articles;
}
```

- Sa se adauge toate metodele pentru operatiile de tip C.R.U.D. (Index – pentru listarea tuturor articolelor, Show – pentru vizualizarea unui articol, New – pentru crearea unui nou articol, Edit – pentru editarea unui articol, Delete – pentru stergerea unui articol)

- Pentru metoda Index o sa procedam astfel: folosim array-ul de articole creat, pe care o sa il afisam si in View – cream un view numit Index. Metoda Index sa fie redenumita in "listare" folosind selectori

  Metoda Index in Controller:

```
public ActionResult Index()
{
     Article[] articles = GetArticles();

     // Adaugam array-ul de articole in view
     ViewBag.Articles = articles;

     return View();
}
```

  Index.cshtml

```
@{
    ViewBag.Title = "Index";
}

<h2>@ViewBag.Title</h2>

@foreach (var article in ViewBag.Articles)
{
    <h1>@article.Title</h1>
    <p>@article.Content</p>
    <small>@article.Date.ToString()</small>
    <a href="/Article/Show/@article.Id">Afisare articol</a>
    <br />
    <hr />
    <br />
}
```

- Metoda Show va afisa detaliile articolului. In cazul in care articolul nu este gasit, sa afiseze un view de eroare cu explicatiile aferente (mesajul de eroare aruncat de exceptie si un mesaj – "Articolul cautat nu poate fi gasit"). View-ul pentru Show va contine si un link catre pagina de afisare a tuturor articolelor.

- Metoda Show sa aiba o varianta alternativa care va returna JSON. In cazul in care articolul nu poate fi gasit dupa ID sa se faca redirectionare catre Index.

```
// Se returneaza JSON cu articolul
return Json(articles[id], JsonRequestBehavior.AllowGet);
                // JSON nu permite returnarea prin metoda GET a
datelor, dar acest lucru
                // poate fi fortat prin setarea parametrului 2 cu
valoarea
                // JsonRequestBehavior.AllowGet
```

- Metoda New o sa aiba doua view-uri asociate: unul pentru afisarea formularului de creare a unui articol (aceasta metoda va avea implicit metoda GET) si un view (caruia ii dam un nume diferit – de exemplu: NewPostMethod) folosit in momentul in care datele pentru creare vor fi trimise catre server. Pentru trimiterea datelor se foloseste metoda POST astfel:

```
<form method="post" action="/Article/New">
      <button type="submit">Adauga articol</button>
</form>
```

- In View-ul pentru metoda New sa se adauge un formular care contine metoda POST, la fel ca exemplul anterior

Exemplu pentru metoda New si View-urile asociate:

```
// GET: afisam formularul de crearea a unui articol
  public ActionResult New()
  {
      return View();
  }

  // POST: trimitem datele articolului catre server pentru creare
  [HttpPost]
  public ActionResult New(Article article)
  {
      // ... cod creare articol ...
      return View("NewPostMethod");
  }
```

New.cshtml

```
@{
    ViewBag.Title = "New";
}

<h2>Creare articol</h2>
<p>Afisare formular de adaugare student</p>
<form method="post" action="/Article/New">
    <button type="submit">Adauga articol</button>
</form>
```

NewPostMethod.cshtml

```
@{
    ViewBag.Title = "NewPostMethod";
}

<h2>NewPostMethod</h2
```

- Se procedeaza in acelasi mod si cu metoda Edit

- In pagina Show putem insera si formularul corespunzator stergerii intrarii prin metoda DELETE, folosind si un buton la fel ca in exemplele anterioare

Exercitiul 2:

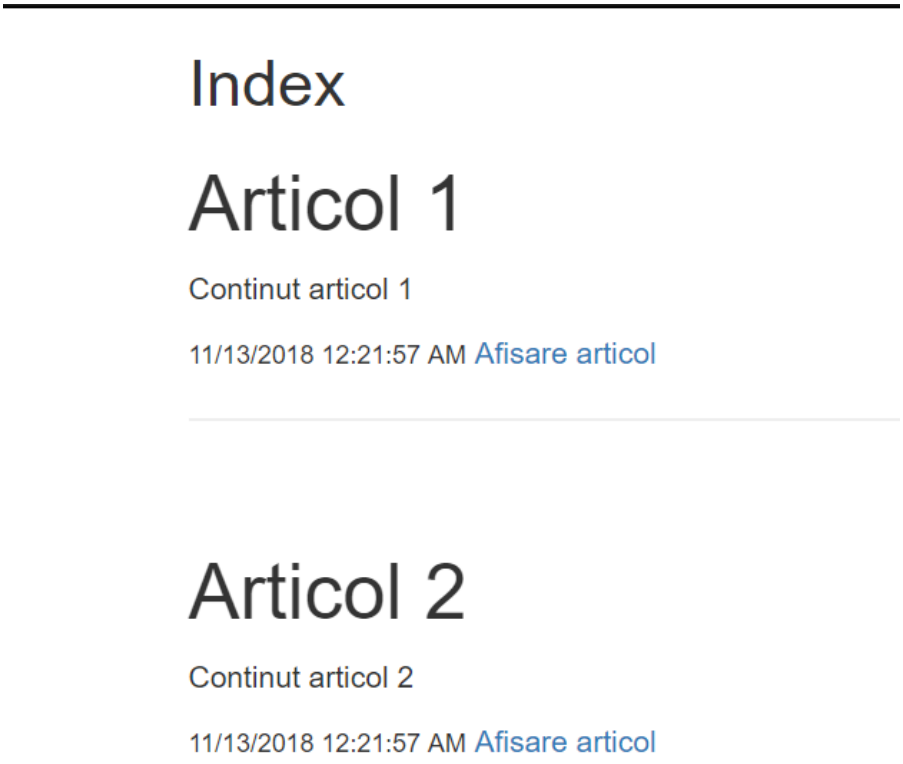
Sa se creeze un folder numit "ActionFilters". Sa se implementeze o clasa numita LogFilter.cs care contine un filtru **OnActionExecuting**. Acest filtru va afisa un mesaj in consola IDE-ului care va contine urmatoarele:

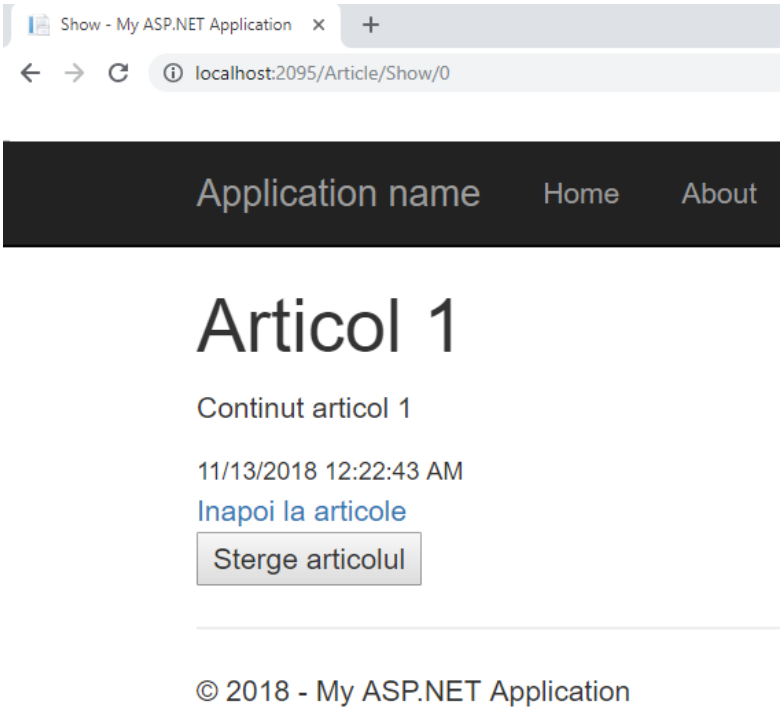
- URL-ul accesat - **`filterContext.HttpContext.Request.Url`**
- Data la care a fost accesat URL-ul
- Adresa IP a utilizatorului - **`UserHostAddress`**
- Detalii despre browser-ul utilizatorului - **`UserAgent`**

Implementati clasa si metodele sale, iar apoi folositi acest filtru custom la nivelul actiunilor Index si Show in controller-ul ArticleController.

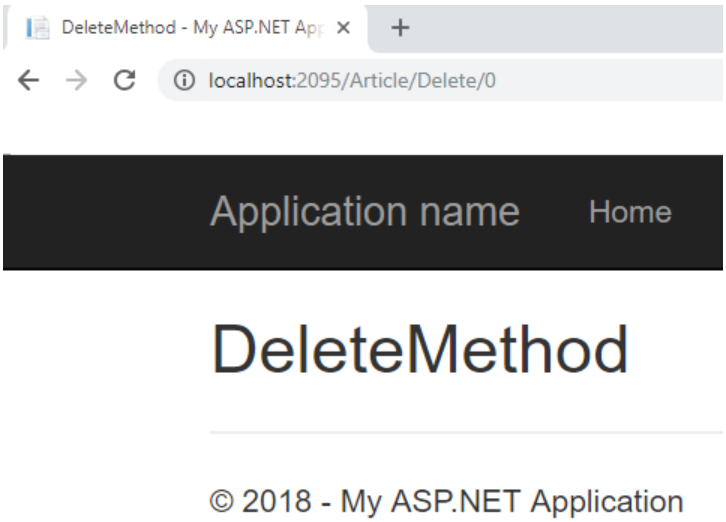
Exercitiul 3:

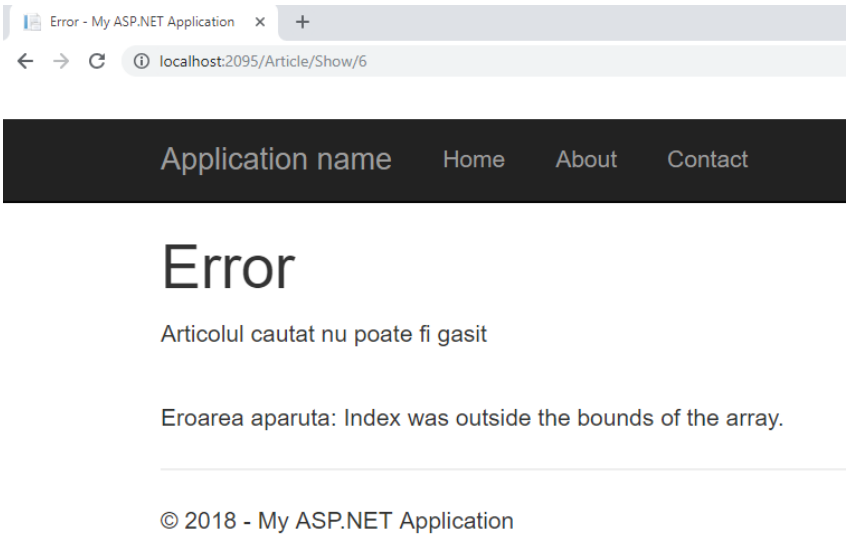
Sa se adauge pe actiunea Index din ArticleController un cache care dureaza 120 de secunde. Porniti aplicatia si incarcati acea pagina. Dupa incarcarea si vizualizarea paginii, modificati View-ul asociat acesteia, adaugand un mesaj. Observati ce se intampla in timpul celor doua minute si dupa scurgerea timpului.

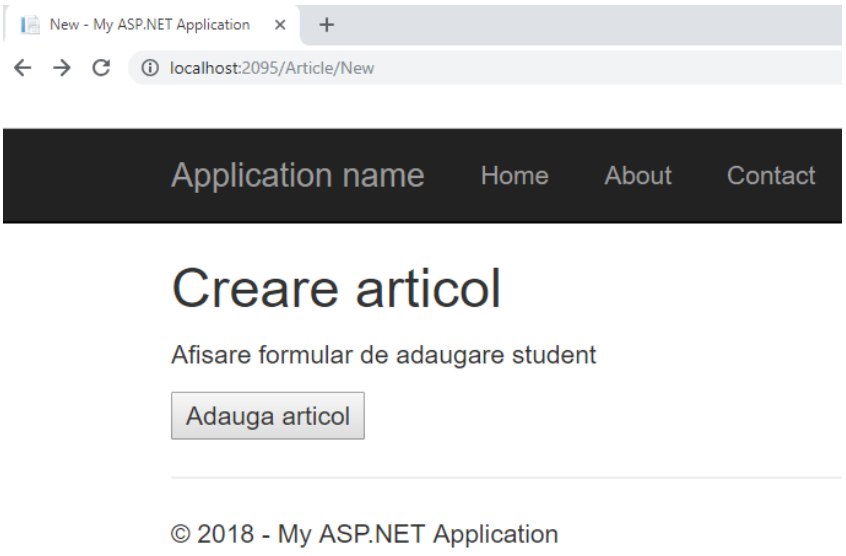




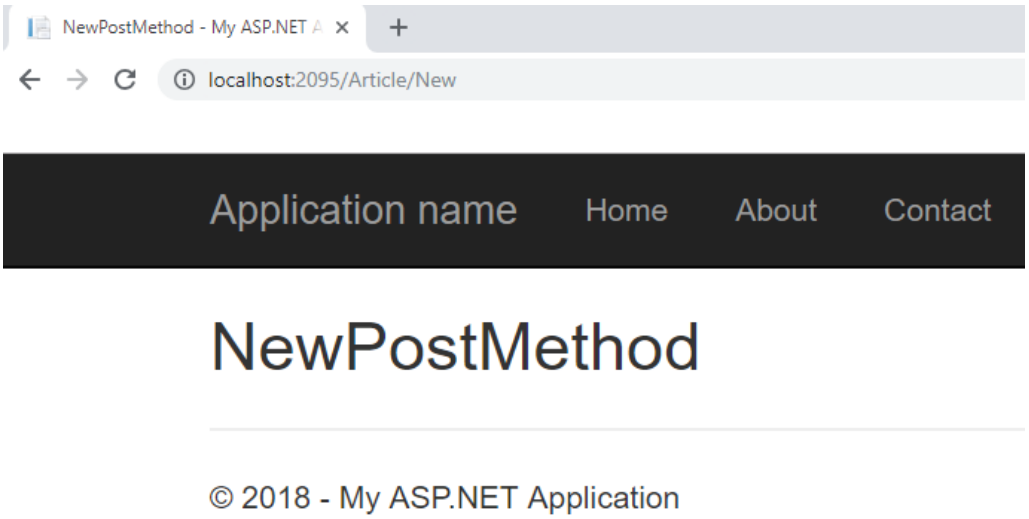
Dupa apasarea butonului din pagina Show.cshtml se va afisa pagina urmatoare:

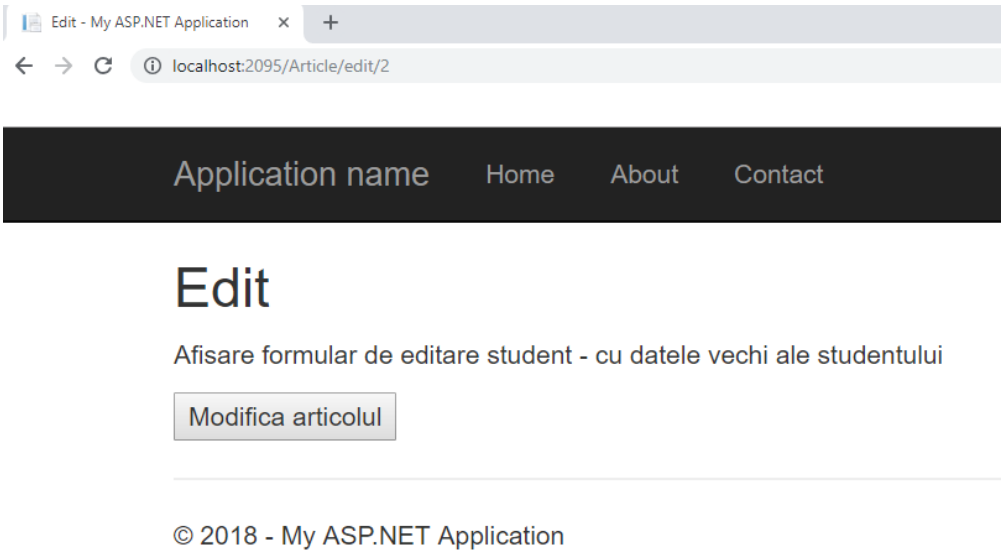



Error - My ASP.NET Application
localhost:2095/Article/Show/6
Application name
Home
About
Contact
Error
Articolul cautat nu poate fi gasit
Eroarea aparuta: Index was outside the bounds of the array.
© 2018 - My ASP.NET Application

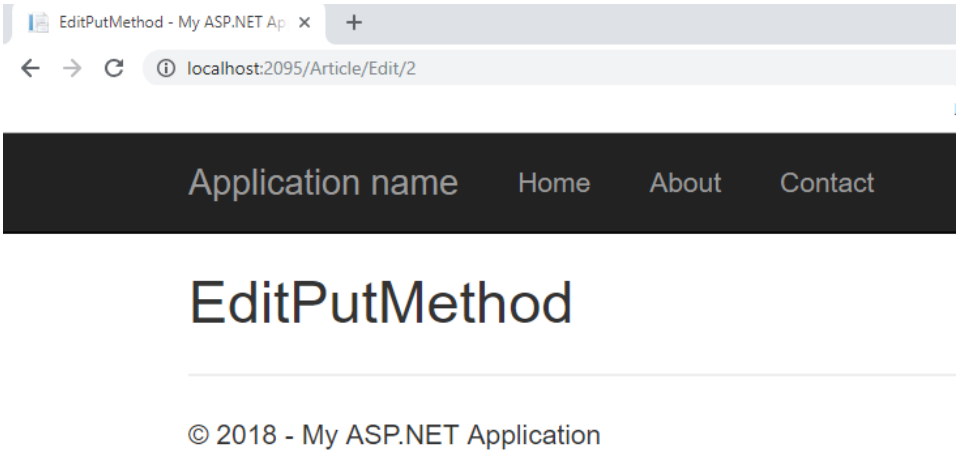

New - My ASP.NET Application
localhost:2095/Article/New
Application name
Home
About
Contact
Creare articol
Afisare formular de adaugare student
Adauga articol
© 2018 - My ASP.NET Application

Dupa apasarea butonului din imaginea anterioara (Adauga Articol) se va afisa pagina urmatoare:





Dupa apasarea butonului Modifica articol se va afisa pagina:



Afisarea Json-ului:


```
{"Id":0,"Title":"Articol 1","Content":"Continut articol 1","Date":"\/Date(1542061710168)\/"}
```