

Dezvoltarea Aplicatiilor Web-Anul 2

Laborator 4

Exercitiul 1

Se considera entitatile **Article** si **Category** cu urmatoarele proprietati:

Article

- Id (int – primary key)
- Title (string – titlul este obligatoriu)
- Content (string – continutul este obligatoriu)
- Date (DateTime)
- CategoryId (int)

Category:

- CategoryId (int primary key)
- CategoryName (string – numele este obligatoriu)

Pentru adaugarea cheii primare si a constrangerilor pentru campurile obligatorii (**Vezi** Curs 4, adaugarea constrangerilor si a cheii primare).

Sa se creeze modelele corespunzatoare, contextul pentru baza de date si relatia dintre aceste modele folosind **Entity Framework** (**Vezi** Curs 4, sectiunea Entity Framework).

Pentru relatia dintre cele doua modele, Article si Category, exista urmatoarea precizare: un articol apartine unei categorii, iar o categorie are mai multe articole (**Vezi** Curs 4, sectiunea relatiile dintre modele (Join)).

Sa se implementeze C.R.U.D. pentru modelul de Articole impreuna cu Categoria acestuia. Sa se implementeze componentele astfel:

- Sa se adauge Controller-ul necesar pentru procesarea categoriilor (**CategoriesController**) – (Vezi Curs 4, sectiunea Folosirea Entity Framework C.R.U.D):

- Metoda **Index** afiseaza toate categoriile, ordonate alfabetic dupa numele categoriei
- Metoda **Show** afiseaza o singura categorie
- Metoda **New** creeaza o noua categorie
- Metoda **Edit** modifica numele unei categorii
- Metoda **Delete** sterge o categorie din baza de date

- Sa se adauge Controller-ul necesar pentru procesarea articolelor (**ArticlesController**):

- Metoda **Index** afiseaza toate articolele din baza de date impreuna cu denumirea categoriei din care face parte articolul

Exemplu:

```
public ActionResult Index()
{
    //var articles = from article in db.Articles
    //                select article;
    // Solutia: Eager loading

    var articles = db.Articles.Include("Category");
    ViewBag.Articles = articles;

    return View();
}

@foreach (var article in ViewBag.Articles)
{
    <h2>@article.Title</h2>
    <p>@article.Content</p>
    <p>@article.Date</p>
    <p>@article.Category.CategoryName</p>
    <a href="/Articles/Show/@article.Id">Afisare articol</a>
    <br />
    <hr />
    <br />
}
```

```

}
<a href="/Categories/Index/">Afisare lista categorii</a>
<br />
<a href="/Articles/New">Adauga articol</a>
<br />

```

- Metoda **Show** afiseaza datele unui articol si categoria acestuia

```

public ActionResult Show(int id)
{
    Article article = db.Articles.Find(id);
    ViewBag.Article = article;
    ViewBag.Category = article.Category;

    return View();
}

```

```

<h1>Afisare articol</h1>
<h2>@ViewBag.Article.Title</h2>
<p>@ViewBag.Article.Content</p>
<p>@ViewBag.Article.Date</p>

<p>Categorie: @ViewBag.Category.CategoryName</p>

<a href="/Articles/Edit/@ViewBag.Article.Id">Modifica articol</a>

<form method="post" action="/Articles/Delete/@ViewBag.Article.Id">
    @Html.HttpMethodOverride(HttpVerbs.Delete)
    <button type="submit">Sterge articol</button>
</form>

<hr />
<a href="/Articles/Index">Inapoi la articole</a>
<br />
<a href="/Articles/New">Adauga articol</a>
<br />

```

- In **adaugarea** de articol sa se selecteze dintr-un dropdown (element html select) o categorie existenta in baza de date; data articolului sa fie data si ora din momentul adaugarii

```
public ActionResult New()
{
    var categories = from cat in db.Categories select cat;
    ViewBag.Categories = categories;
    return View();
}

[HttpPost]
public ActionResult New(Article article)
{
    //TODO
}

<form method="post" action="/Articles/New">
    <label>Titlu articol</label>
    <input type="text" name="Title" />
    <br />
    <label>Continut articol</label>
    <textarea name="Content"></textarea>
    <br />
    <input type="hidden" name="Date" value="@DateTime.Now" />
    <br />

    <label>Selectati categoria</label>

    <select name="CategoryId">
        @foreach (var category in ViewBag.Categories)
        {
            <option
value="@category.CategoryId">@category.CategoryName</option>
        }
    </select>
    <br />
    <button type="submit">Adauga articol</button>
</form>
```

- In **modificarea** articolului sa se afiseze datele articolului si categoria anterioara a acestuia (deja selectata); la modificarea articolului sa se modifice si data de creare a articolului

```
public ActionResult Edit(int id)
{
    Article article = db.Articles.Find(id);
    ViewBag.Article = article;
    ViewBag.Category = article.Category;
    var categories = from cat in db.Categories select cat;
    ViewBag.Categories = categories;
    return View();
}

[HttpPut]
public ActionResult Edit(int id, Article requestArticle)
{
    try
    {
        Article article = db.Articles.Find(id);
        if (TryUpdateModel(article))
        {
            article.Title = requestArticle.Title;
            article.Content = requestArticle.Content;
            article.Date = requestArticle.Date;
            article.CategoryId = requestArticle.CategoryId;
            db.SaveChanges();
        }
        return RedirectToAction("Index");
    }
    catch (Exception e)
    {
        return View();
    }
}
```

```
<form method="post" action="/Articles/Edit/@ViewBag.Article.Id">
    @Html.HttpMethodOverride(HttpVerbs.Put)
    <label>Titlu articol</label>
    <br />
    <input type="text" name="Title" value="@ViewBag.Article.Title" />
    <br /><br />
    <label>Continut articol</label>
    <br />
    <input type="text" name="Content" value="@ViewBag.Article.Content" />
```

```

<br /><br />
<label>Data</label>
<br />
<input type="text" name="Date" value="@ViewBag.Article.Date" />
<br />
<select name="CategoryId">
    @foreach (var category in ViewBag.Categories)
    {
        <option
            @if (category.CategoryId ==
ViewBag.Category.CategoryId) {
                @:selected="selected"
            }
        >
value="@category.CategoryId">@category.CategoryName</option>
    }
</select>
<br />
<button type="submit">Modifica articol</button>
</form>

```

- Posibilitatea **stergerii** unui articol

```

[Http...]
public ActionResult Delete(int id)
{
    //TODO
}

```