

ПРАКТИЧНА РОБОТА №6

Розробка WEB додатків на базі ASP.NET MVC

*Фільтрація даних в ASP.NET MVC.**Набути практичних навиків організації можливості фільтрації даних на сторінці.*

ПОРЯДОК ВИКОНАННЯ ЗАВДАННЯ

В цій практичній роботі, розглянемо, як фільтрувати дані по визначеним критеріям в ASP.NET MVC. Реалізацію даного функціоналу виконаємо на прикладі з попередньої практичної роботи, де в нас були моделі футболістів та команд.

```
01 public class Player
02 {
03     public int Id { get; set; }
04     public string Name { get; set; }
05     public int Age { get; set; }
06     public string Position { get; set; }
07
08     public int? TeamId { get; set; }
09     public Team Team { get; set; }
10 }
11 public class Team
12 {
13     public int Id { get; set; }
14     public string Name { get; set; }
15     public string Coach { get; set; }
16
17     public ICollection<Player> Players { get; set; }
18     public Team()
19     {
20         Players = new List<Player>();
21     }
22 }
```

Допустимо, що при виводі списку футболістів, виникає необхідність для фільтрації даних за певними критеріями, наприклад за командою та позицією на полі. Так як ми будемо виводити в одному представленні і список всіх гравців, і критерії для вибору, які представляють списки команд і позиції, то додамо в проект ще одну модель під назвою **PlayersListViewModel**:

```
01 public class PlayersListViewModel
02 {
03     public IEnumerable<Player> Players { get; set; }
04     public SelectList Teams { get; set; }
05     public SelectList Positions { get; set; }
06 }
```

Модель допомагає реалізувати три властивості: для списку гравців, команд і позицій.

Щоб у вас запрацював метод **SelectList** не забудьте підключити наступний модуль:

```
01 using System.Web.Mvc;
```

Тепер на стороні контролера визначимо бізнес-логіку, яка буде виводити список гравців та виконувати фільтрацію. Простіше кажучи перепишемо метод **Index** наступним чином:

```
01 public ActionResult Index(int? team, string position)
02 {
03     IQueryable<Player> players = db.Players.Include(p => p.Team);
04     if (team != null && team != 0)
05     {
06         players = players.Where(p=>p.TeamId==team);
07     }
08     if (!String.IsNullOrEmpty(position) &&
09 !position.Equals("Bci"))
10     {
11         players = players.Where(p => p.Position == position);
12     }
```

```
13
14     List<Team> teams = db.Teams.ToList();
15     // встановлюємо початковий елемент, який дозволить вибрати
16     всі записи
17     teams.Insert(0, new Team { Name = "Bci", Id = 0 });
18
19     PlayersListViewModel plvm = new PlayersListViewModel
20     {
21         Players=players.ToList(),
22         Teams = new SelectList(teams, "Id", "Name"),
23         Positions = new SelectList(new List<string>()
24         {
25             "Bci",
26             "ФРВ",
27             "ЦАП",
28             "ЦЗ",
29             "ВРТ"
30         })
31     };
32     return View(plvm);
33 }
```

При зверненні до методу **Index**, незалежно від того чи були передані параметри **team** і **position**, все рівно в представленні буде виводитися список гравців. Якщо конкретний параметр був переданий, то до об'єкта **IQueryable**, що відповідає за вибірку даних, додається вираз:

```
01 | Where: players = players.Where(p=>p.TeamId==team);.
```

Для формування випадяючого списку команд в представленні, отримуємо його з БД:

```
01 | List<Team> teams = db.Teams.ToList();
```

Для можливості вибору гравців з будь-якої команди, додаємо в цей список новий пункт:

```
01 | teams.Insert(0, new Team { Name = "Bce", Id = 0 });
```

З реалізацією контролера закінчено, перейдемо до останнього кроку, це модифікація існуючого представлення **Index.cshtml**. Перевизначимо існуючий код представлення на наступний:

```
01 @model SoccerApp.Models.PlayersListViewModel
02 @{
03     ViewBag.Title = "Список гравців";
04     Layout = "~/Views/Shared/_Layout.cshtml";
05 }
06 <h2>Каталог гравців</h2>
07 <p>
08     @Html.ActionLink("Додати гравця", "Create")
09 </p>
10 <form method="get">
11     <div class="form-inline">
12         <label class="control-label">Команда: </label>
13         @Html.DropDownList("team", Model.Teams as SelectList,
14             htmlAttributes: new { @class = "form-control" })
15
16         <label class="control-label">Позиція гравця: </label>
17         @Html.DropDownList("position", Model.Positions as SelectList,
18             htmlAttributes: new { @class = "form-control" })
19
20         <input type="submit" value="Фільтр" class="btn btn-primary"
21     />
22     </div>
23 </form>
24 <br>
25 <table class="table">
26     <tr>
27         <th>Ім'я</th>
28         <th>Вік</th>
29         <th>Позиція</th>
30         <th>Команда</th>
31         <th></th>
32     </tr>
33
34     @foreach (var item in Model.Players)
35     {
36         <tr>
37             <td>
38                 @Html.DisplayFor(modelItem => item.Name)
39             </td>
40             <td>
41                 @Html.DisplayFor(modelItem => item.Age)
42             </td>
```

```
43         <td>
44             @Html.DisplayFor(modelItem => item.Position)
45         </td>
46         <td>
47             @Html.DisplayFor(modelItem => item.Team.Name)
48         </td>
49         <td>
50             @Html.ActionLink("Змінити дані", "Edit", new { @class
51 = "btn btn-warning", id = item.Id }) |
52             @Html.ActionLink("Переглянути дані", "TeamDetails",
53 new { @class = "btn btn-info", id = item.Id }) |
54             @Html.ActionLink("Видалити дані", "Delete", new {
55 @class = "btn btn-danger", id = item.Id })
56         </td>
57     </tr>
58 }
59 </table>
```

Перед таблицею з списком гравців додано форму, яка за допомогою запитів GET відправляє вибрані значення із випадających списків на сервер. Всі отримані і створені списки використовуються для формування об'єкта **PlayerListViewModel**, який передається в представлення.

My first ASP.NET app

Каталог гравців

[Додати гравця](#)

Команда:

Ім'я	Вік	Позиція	Команда	
Бруно Фернандес	26	ЦАП	Манчестер Юнайтед	Змінити дані Переглянути дані Видалити дані
Олександр Зінченко	22	ЛЗ	Манчестер Сіті	Змінити дані Переглянути дані Видалити дані
Мохамед Салах	28	ПФА	Ліверпуль	Змінити дані Переглянути дані Видалити дані
Нголо Канте	28	ЦОП	Челсі	Змінити дані Переглянути дані Видалити дані
Маркус Рашфорд	25	ФРВ	Манчестер Юнайтед	Змінити дані Переглянути дані Видалити дані

© 2021 - SoccerApp

My first ASP.NET app

Каталог гравців

[Додати гравця](#)Команда:

Ім'я	Вік	Позиція	Команда	
Бруно Фернандес	26	ЦАП	Манчестер Юнайтед	Змінити дані Переглянути дані Видалити дані
Маркус Рашфорд	25	ФРВ	Манчестер Юнайтед	Змінити дані Переглянути дані Видалити дані

© 2021 - SoccerApp

My first ASP.NET app

Каталог гравців

[Додати гравця](#)Команда:

Ім'я	Вік	Позиція	Команда	
Бруно Фернандес	26	ЦАП	Манчестер Юнайтед	Змінити дані Переглянути дані Видалити дані

© 2021 - SoccerApp

Індивідуальне завдання

Додати до свого варіанту завдань можливість використання фільтрації даних на сторінці.