

**Сучасні інтернет технології. Лабораторне заняття №5**  
**ГЛОБАЛІЗАЦІЯ ТА ЛОКАЛІЗАЦІЯ ЗАСТОСУНКУ ASP.NET CORE**

Виконала: студентка МІТ-41, Панченко Владислава

Завдання для виконання (max - 100 балів)

1. Ознайомитися з теоретичними основами глобалізації та локалізації:  
поняття Culture, UICulture, RFC 4646, ISO 639, ISO 3166 (сайт Microsoft).  
(max - 10 балів)

### 1. Глобалізація (Globalization, G11n)

Глобалізація — це процес проєктування та розробки програми таким чином, щоб її можна було легко адаптувати до різних культурних та регіональних вимог.

Глобалізація охоплює інтернаціоналізацію (Internationalization, I18n), яка фокусується на виділенні всіх ресурсів (рядків, повідомень, графіки), що залежать від мови та культури, та створення коду, незалежного від цих даних.

### 2. Локалізація (Localization, L10n)

Локалізація — це процес адаптації інтернаціоналізованої програми до конкретного регіонального ринку або мови.

Локалізація включає переклад інтерфейсу користувача та вмісту для конкретної мови (наприклад, української), а також забезпечення правильного форматування для певної культури.

#### 1. Culture (Культура)

Поняття Culture в .NET і загалом у програмуванні визначає формат відображення даних, які залежать від регіону:

Формати дати та часу: день/місяць/рік (Україна) проти місяць/день/рік (США).

Формати чисел: Роздільник цілої та дробової частини (кома vs. крапка).

Формати валют: Символ валюти (₴, \$, €) та його розташування.

Сортування рядків.

У .NET це поняття представлено класом System.Globalization.CultureInfo.

## 2. UICulture (Культура інтерфейсу користувача)

UICulture (User Interface Culture) визначає, якою мовою відображаються текстові ресурси (рядки, меню, повідомлення, ключі з файлів .resx). Наприклад, якщо Culture встановлено на en-US (для форматування в доларах США), але UICulture встановлено на uk-UA, користувач побачить інтерфейс українською мовою, але дані (дата, час, валюта) будуть відформатовані відповідно до американських стандартів.

UICulture використовується для вибору правильного файлу ресурсів (наприклад, Resource.uk-UA.resx).

## 3. Стандарти кодування культур

Культури кодуються за міжнародними стандартами. В .NET та веброзробці використовується формат, визначений RFC 4646 (раніше RFC 3066).

RFC 4646 (та 3066): Визначає синтаксис для тегів ідентифікації мови (Language Tags), які поєднують мову та регіон.

Формат: <мова>-<РЕГІОН>.

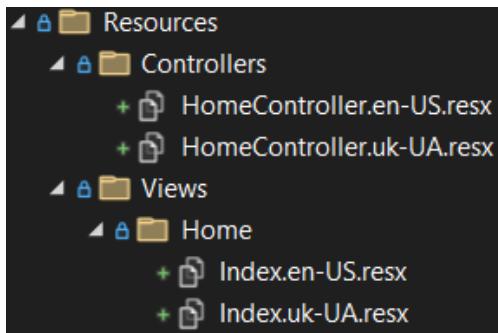
Приклади: uk-UA (Українська, Україна), en-US (Англійська, США), es-ES (Іспанська, Іспанія).

Стандарт	Призначення
<b>ISO 639-1 / ISO 639-2</b>	Дво- або трилітерний код для <b>мови</b> (наприклад, uk для української, en для англійської).
<b>ISO 3166-1</b>	Дволітерний код для <b>країни/регіону</b> (наприклад, UA для України, US для США).

## 2. Додати до ASP.NET Core застосунку підтримку локалізації через AddLocalization() з вказанням папки ресурсів. (max - 5 балів)

```
builder.Services.AddLocalization(options => options.ResourcesPath = "Resources");
```

3. Створити ресурсні файли .resx для базової культури та щонайменше двох додаткових. (max - 10 балів)



4. Налаштювати RequestLocalizationOptions з підтримкою щонайменше трьох культур. Встановити культуру за замовчуванням. (max - 10 балів)

```
builder.Services.Configure<RequestLocalizationOptions>(options =>
{
    var supportedCultures = new[]
    {
        new CultureInfo("en-US"), // Англійська (США)
        new CultureInfo("uk-UA"), // Українська
        new CultureInfo("es")     // Іспанська
    };

    options.DefaultRequestCulture = new RequestCulture("en-US");
    options.SupportedCultures = supportedCultures;
    options.SupportedUITcultures = supportedCultures;
});
```

5. Додати middleware UseRequestLocalization() у правильному місці конвеєра запиту. Пояснити, чому порядок має значення. (max - 5 балів)

```
app.UseRequestLocalization();

if (app.Environment.IsDevelopment())
{
    app.UseMigrationsEndPoint();
}
else
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}
```

Порядок проміжного ПЗ у конвеєрі ASP.NET Core має значення, тому що запити проходять через цей конвеєр послідовно. Кожне проміжне ПЗ виконує свою роботу, а потім передає запит наступному.

Хоча UseRequestLocalization() може працювати й до UseRouting(), часто провайдери культури (Culture Providers), які визначають мову користувача, можуть залежати від інформації про маршрут.

Наприклад, провайдер RouteDataRequestCultureProvider шукає культуру у параметрах URL (наприклад, /uk-UA/Home/Index). Якщо UseRouting() не виконано, параметри маршруту ще не визначені, і провайдер не зможе витягти код культури.

6. Реалізувати типізовану локалізацію у контролері через IStringLocalizer та у Razor View через IViewLocalizer. (max - 15 балів)

```
using Microsoft.AspNetCore.Mvc.Localization
[Inject] IViewLocalizer Localizer

{@
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <!-- Текст із файлу ресурсів (Index.*.resx) -->
    <h1 class="display-4">@Localizer["HeaderTitle"]</h1>

    <!-- Текст із контролера -->
    <div class="alert alert-info">
        @ ViewData["MessageFromController"]
    </div>

    <div class="card mt-4">
        <div class="card-body">
            <h5>Тест форматів:</h5>
            <p>Поточна дата: <strong>@DateTime.Now.ToString("yyyy-MM-dd")</strong></p>
            <p>Валюта (1000): <strong>@(1000.ToString("C"))</strong></p>
        </div>
    </div>
</div>
```

```

0 references
public HomeController	ILogger<HomeController> logger, IWebRepository repository, MyConfiguration config, IStringLocalizer<HomeController> localizer)
{
    _logger = logger;
    _repository = repository;
    _config = config;
    _localizer = localizer;
}

[AllowAnonymous]
0 references
public async Task<IActionResult> IndexAsync()
{
    ViewData["MessageFromController"] = _localizer["WelcomeMessage"];
    var users = await _repository.ReadAll<WebApplicationUser>().ToListAsync();
    return View(users);
}

```

7. Реалізувати перемикання мов, що встановлює культуру через cookies. Перевірити збереження обраної культури між запитами та сесіями. (max - 15 балів)

```

using Microsoft.AspNetCore.Localization;
using Microsoft.AspNetCore.Mvc;

namespace WebApplication1.Controllers
{

    public class SelectLanguageController : Controller
    {
        [HttpPost]

        public IActionResult SetLanguage(string culture, string returnUrl)
        {
            Response.Cookies.Append(
                CookieRequestCultureProvider.DefaultCookieName,
                CookieRequestCultureProvider.MakeCookieValue(new RequestCulture(culture)),
                new CookieOptions { Expires = DateTimeOffset.UtcNow.AddYears(1) }
            );

            return LocalRedirect(returnUrl);
        }
    }
}

```

```

@using Microsoft.AspNetCore.Builder
@using Microsoft.AspNetCore.Localization
@using Microsoft.AspNetCore.Mvc.Localization
@using Microsoft.Extensions.Options

@inject IViewLocalizer Localizer
@inject IOptions<RequestLocalizationOptions> LocOptions

 @{
    var requestCulture = Context.Features.Get<IRemoteFeature>();
    var cultureItems = LocOptions.Value.SupportedUICultures
        .Select(c => new SelectListItem { Value = c.Name, Text = c.NativeName })
        .ToList();
    var returnUrl = string.IsNullOrEmpty(Context.Request.Path) ? "/" : $"~{Context.Request.Path.Value}";
}

<div class="mx-3">
    <form id="selectLanguage" asp-controller="SelectLanguage" asp-action="SetLanguage"
        asp-route-returnUrl="@returnUrl" method="post">
        <select name="culture">
            onchange="this.form.submit()"
            asp-for="@requestCulture.RequestCulture.UICulture.Name"
            asp-items="cultureItems"
            class="form-select form-select-sm">
        </select>
    </form>
</div>

```

8. Реалізувати підтримку параметрів URL (?culture=en-US) для вибору культури. Перевірити зміну мови. (max - 5 балів)

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

9. Реалізувати локалізацію форматів дат, чисел і валют у залежності від обраної культури. (max - 10 балів)

```
@using Microsoft.AspNetCore.Mvc.Localization
@inject IViewLocalizer Localizer

{@
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <!-- Текст із файлу ресурсів (Index.*.resx) -->
    <h1 class="display-4">@Localizer["HeaderTitle"]</h1>

    <!-- Текст із контролера -->
    <div class="alert alert-info">
        @ViewData["MessageFromController"]
    </div>

    <div class="card mt-4">
        <div class="card-body">
            <h5>Тест форматів:</h5>
            <p>Поточна дата: <strong>@DateTime.Now.ToString("yyyy-MM-dd")</strong></p>
            <p>Валюта (1000): <strong>@(1000.ToString("C"))</strong></p>
        </div>
    </div>
</div>
```

10. Зафіксувати зміни у проєкті на GitHub. (max - 10 балів)

11. Оформити звіт. (max - 5 балів)