

## Лабораторная работа №5

### Razor Pages. Передача файлов

#### 1. Цель работы.

Знакомство со сценариями построения веб-приложений, основанных на страницах. Знакомство с механизмом передачи файлов от клиента на сервер.

#### 2. Задача работы

Научиться создавать страницы Razor. Научиться передавать файлы на сервер. Научиться передавать файлы в REST – сервис.

**Время выполнения работы: 2 часа**

#### 3. Выполнение работы. Страницы администратора

##### 3.1. Постановка задачи

Требуется создать страницы администратора, позволяющие манипулировать данными в базе данных.

Страницы должны быть выполнены по сценарию Razor Pages.

**Страницы администратора должны располагаться в области (Areas) с названием Admin (см. п. 3.1 лабораторной работы №2).**

##### 3.2. Подготовка проекта.

Чтобы вручную не создавать страницы администратора, предлагается воспользоваться механизмом Scaffold.

Однако, данный механизм предполагает использование контекста базы данных. Для того, чтобы обойти данное ограничение предлагается в основном проекте **ВРЕМЕННО**

- **либо** описать контекст базы данных, аналогичный контексту проекта XXX.API

- **либо** сделать ссылку на проект XXX.API

После создания страниц администратора контекст БД, либо ссылку на проект XXX.API нужно будет удалить из основного проекта.

### 3.3. Создание страниц администратора

Сгенерируйте страницы (**Razor pages**) администратора для выполнения операций CRUD над объектами вашей предметной области согласно п.3.1.

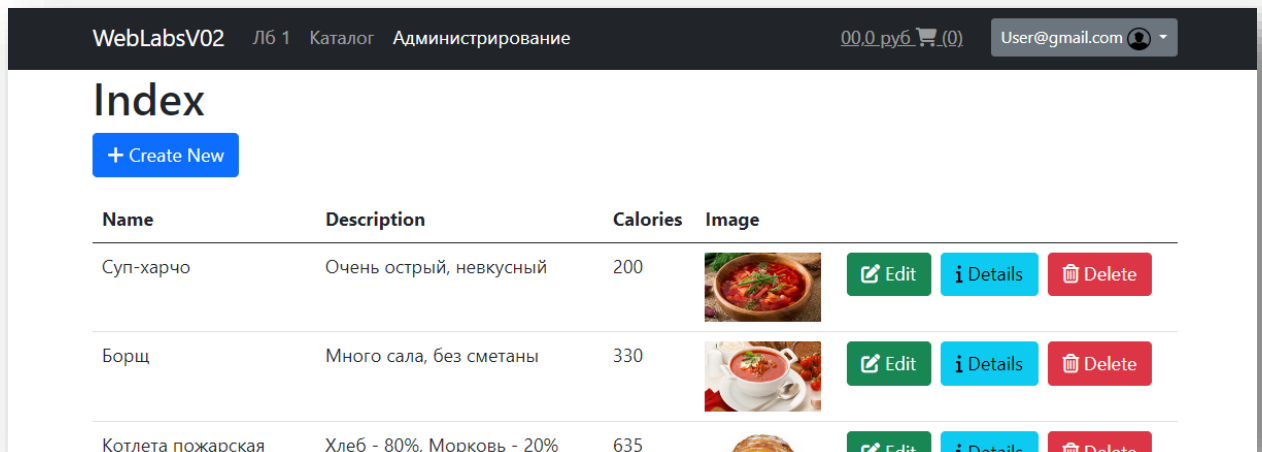
В коде моделей всех страниц замените использование контекста базы данных на использование IProductService

Для использования страницы макета и базовых пространств имен в папку, в которой находятся страницы администратора, скопируйте файлы \_ViewImports.cshtml и \_ViewStart.cshtml из папки Views

В разметке страницы Index вместо имени файла изображения укажите тэг для вывода изображения на страницу.

На странице Index оформите ссылки переключения между страницами с помощью иконок и стилей Bootstrap.

Запустите проект. Перейдите на страницу администрирования. Убедитесь, что страницы работают корректно.



## 4. Выполнение работы. Передача файлов

### 4.1. Постановка задачи

Страницы создания и редактирования объекта должны предоставлять возможность передачи изображения объекта.

Изображения должны сохраняться в проекте XXX.API, в папке wwwroot/Images.

Для исключения дублирования имен файлов назначать файлам изображения случайные имена.

При редактировании объекта, при замене изображения, предыдущий файл изображения должен удаляться.

*Соглашение об именовании:* при передаче файла через http(https) предлагается использовать имя **file**

## 4.2. Контроллер API для сохранения файлов

Для доступа клиента к файлам добавьте в классе program проекта XXX.API компонент Middleware для обработки запросов статических файлов.

Создайте API контроллер (FilesController) для сохранения и удаления файла в папке wwwroot/Images. При сохранении файла контроллер должен вернуть URL сохраненного файла.

Для работы с фалами в Api-контроллере понадобится путь к папке wwwroot. Для получения этого пути внедрите в конструктор контроллера объект IWebHostEnvironment.

Пример:

```
[Route("api/[controller]")]
[ApiController]
public class FilesController : ControllerBase
{
    // Путь к папке wwwroot/Images
    private readonly string _imagePath;

    public FilesController(IWebHostEnvironment webHost)
    {
        _imagePath = Path.Combine(webHost.WebRootPath, "Images");
    }
    [HttpPost]
    public async Task<IActionResult> SaveFile(IFormFile file)
    {
        if (file is null)
        {
            return BadRequest();
        }
        // путь к сохраняемому файлу
```

```

        var filePath = Path.Combine(_imagePath, file.FileName);
        var fileInfo = new FileInfo(filePath);
        // если такой файл существует, удалить его
        if (fileInfo.Exists)
        {
            fileInfo.Delete();
        }
        // скопировать файл в поток
        using var fileStream = fileInfo.Create();
        await file.CopyToAsync(fileStream);

        // получить Url файла
        var host = HttpContext.Request.Host;
        var fileUrl = $"https://{host}/Images/{file.FileName}";

        return Ok(fileUrl);
    }

    [HttpDelete]
    public IActionResult DeleteFile(string fileName)
    {
        . . .

        return Ok();
    }
}

```

#### 4.3. Сервис для работы с файлами (проект xxx.UI)

Опишите интерфейс `IFileService`, содержащий методы для сохранения и удаления файлов.

```

public interface IFileService
{
    /// <summary>
    /// Сохранить файл
    /// </summary>
    /// <param name="formFile">Файл, переданный формой</param>
    /// <returns>URL сохраненного файла</returns>
    Task<string> SaveFileAsync(IFormFile formFile);
    /// <summary>
    /// Удалить файл
    /// </summary>
    /// <param name="fileName">Имя файла</param>
    /// <returns></returns>
    Task DeleteFileAsync(string fileName);
}

```

Опишите сервис `ApiFileService`, который реализует интерфейс `IFileService` для взаимодействия с контроллером `FilesController` проекта `xxx.API`.

Пример:

```
public class ApiFileService : IFileService
{
    private readonly HttpClient _httpClient;

    public ApiFileService(HttpClient httpClient)
    {
        _httpClient = httpClient;
        _httpContext = HttpContextAccessor.HttpContext;
    }
    public async Task DeleteFileAsync(string fileUri)
    {
        . . .
    }

    public async Task<string> SaveFileAsync(IFormFile formFile)
    {
        // Создать объект запроса
        var request = new HttpRequestMessage
        {
            Method = HttpMethod.Post
        };
        // Сформировать случайное имя файла, сохранив расширение
        var extension = Path.GetExtension(formFile.FileName);
        var newName = Path.ChangeExtension(Path.GetRandomFileName(), extension);
        // Создать контент, содержащий поток загруженного файла
        var content = new MultipartFormDataContent();
        var streamContent = new StreamContent(formFile.OpenReadStream());
        content.Add(streamContent, "file", newName);
        // Поместить контент в запрос
        request.Content = content;
        // Отправить запрос к API
        var response = await _httpClient.SendAsync(request);
        if (response.IsSuccessStatusCode)
        {
            // Вернуть полученный Url сохраненного файла
            return await response.Content.ReadAsStringAsync();
        }
        return String.Empty;
    }
}
```

**Примечание:** обратите внимание, что имя контента «file» должно совпадать с именем параметра «IFormFile file», передаваемого в метод SaveFile контроллера в проекте XXX.API

Зарегистрируйте сервис HttpClient для созданного файлового сервиса в классе HostingExtensions:

```
builder.Services.AddHttpClient<IFileService, ApiFileService>(opt =>
    opt.BaseAddress = new Uri($"{UriData.ApiUri}Files"));
```

#### 4.4. Доработка сервиса ApiProductService (проект xxx.UI)

Добавьте в методы создания и редактирования объектов возможность сохранения файла изображения с помощью сервиса IFileService.

Пример:

```
public async Task<ResponseData<Dish>> CreateProductAsync(
    Dish product, IFormFile? formFile)
{
    // Первоначально использовать картинку по умолчанию
    product.Image = "Images/noimage.jpg";

    // Сохранить файл изображения
    if (formFile != null)
    {
        var imageUrl = await _fileService.SaveFileAsync(formFile);
        // Добавить в объект Url изображения
        if (!string.IsNullOrEmpty(imageUrl))
            product.Image = imageUrl;
    }

    var uri = new Uri(_httpClient.BaseAddress.AbsoluteUri + "Dishes");
    var response = await _httpClient.PostAsJsonAsync(uri, product,
        _serializerOptions);
    if (response.IsSuccessStatusCode)
    {
        var data = await response
            .Content
            .ReadFromJsonAsync<ResponseData<Dish>> (_serializerOptions);
        return data;
    }
    _logger.LogError($"-----> object not created. Error:
    {response.StatusCode.ToString()}");
    return ResponseData<Dish>
```

```

        .Error($"Объект не добавлен. Error:
    {response.StatusCode.ToString()}");
    }

```

#### 4.5. Доработка страниц Create и Update

*В коде* модели страниц Create и Update добавьте свойство

```

[BindProperty]
public IFormFile? Image { get; set; }

```

Передайте объект Image в сервис IProductService при создании/изменении объекта.

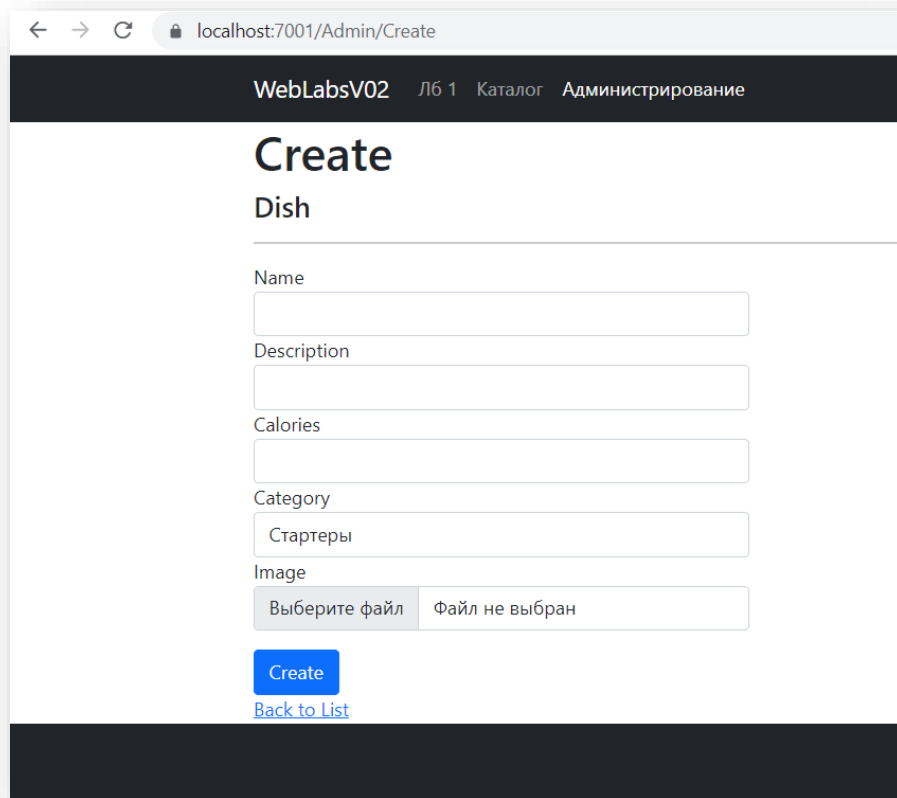
Пример:

```
await _service.CreateProduct(Dish, Image);
```

*В разметке* страниц Create и Update добавьте тэг для выбора и передачи файла. «Привяжите» его к свойству модели Image с помощью тэг-хелпера `asp-for`

**Примечание:** для передачи файлов в тэге form необходимо установить атрибут `enctype="multipart/form-data"`

Оформите тэг с помощью классов Bootstrap (см. <https://getbootstrap.com/docs/5.1/forms/input-group/#custom-file-input> )



← → ↻ localhost:7001/Admin/Create

WebLabsV02 ЛБ 1 Каталог Администрирование

## Create

### Dish

Name

Description

Calories

Category

Image

Выберите файл Файл не выбран

Create

[Back to List](#)

Запустите проект. Убедитесь, что изображение сохраняется на сервере.

#### 4.5.1. Оформление страницы редактирования (*необязательно*)

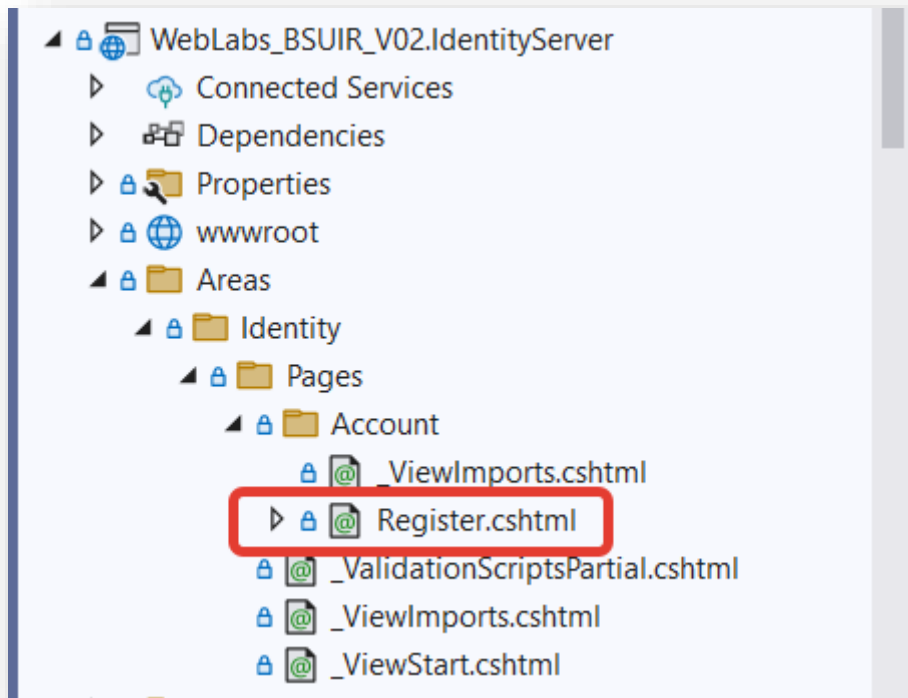
При редактировании объекта необходимо выводить текущее изображение объекта

### 5. Контрольные вопросы

- 1) Чем сценарий Razor Pages отличается от MVC?
- 2) Что такое модель страницы Razor?
- 3) Как обрабатываются запросы к страницам Razor?
- 4) Как в разметке страницы Razor получить доступ к свойствам модели страницы?
- 5) Как привязать данные формы к модели страницы?
- 6) Для чего используется директива @page ?



7) Какой Url будет у страницы Register на рисунке?



8) Чем отличается адрес страницы «/index» от «./index»?

9) Как указать дополнительный сегмент маршрута к странице (например, id)?

10) Какой Middleware используется для передачи статического контента клиенту?

11) Какой интерфейс реализует объект, полученный сервером, при передаче файла от клиента на сервер?

12) Что такое поставщики файлов (FileProvider)?

13) Как получить путь к папке «wwwroot»?

14) Форма передает файл. В каком виде этот файл будет представлен на сервере?

15) Какой атрибут нужно установить в тэге <form>, чтобы можно было передавать файлы?