Введение в Razor Pages

ASP.NET Core

Сегодня

- Разберем что такое паттерн MVC и зачем он нужен
- Создадим полноценный веб сайт на ASP.NET Core Razor Pages

Razor Pages

- Модель программирования Razor Pages была представлена в ASP.NET Core 2.0 как способ создания многостраничных веб-сайтов с отрисовкой на стороне сервера
- Представляет собой шаблонный движок, который позволяет верстать страницы
- Сервер каждый раз рендерит HTML страницу и возвращает результат браузеру

Пример Razor Pages

Накладная (расшифровка) Сайт Интернет-магазин Оферта

Заказ 6445 ✔ оплачен

от 1 ноября 2021

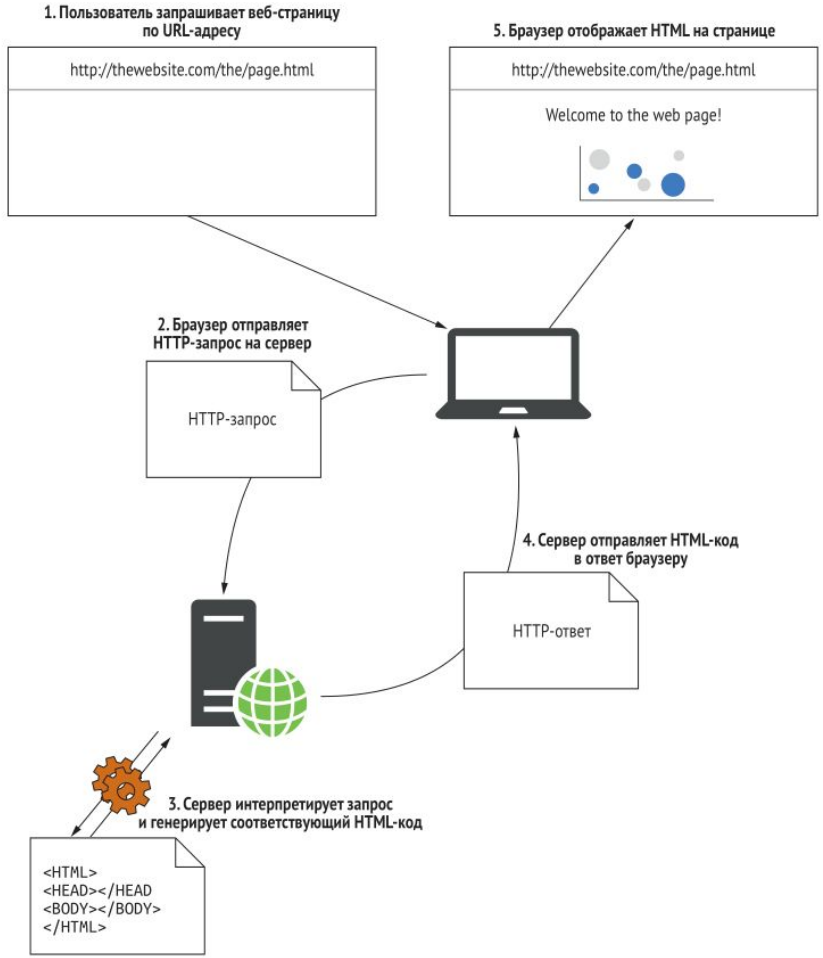
Накладная

№	Артикул	Название	Количество	Цена	Сумма
1	136313	С/порошок ARIEL M-Zim Color 5 д/цв. авт., 9кг	1,000	999,00 ₽	999,00 ₽
2	383293	Сыр ВЕЛИКОЛУКСКИЙ МОЛ КОМБИНАТ Маасдам 45% вес без змж	1,117	632,00 ₽	705,94 ₽
3	104908	Мороженое BASKIN ROBBINS Клубничное отличное без змж, 500мл	1,000	526,00 ₽	526,00 ₽
4	504841	Канзас стейк BLACK ANGUS из мрам. гов. WrC охл SKIN, 390г	1,000	498,00 ₽	498,00 ₽

Синтаксис Razor

https://docs.microsoft.com/en-us/aspnet/core/mvc/views/razor?view=aspnetcore-6.0

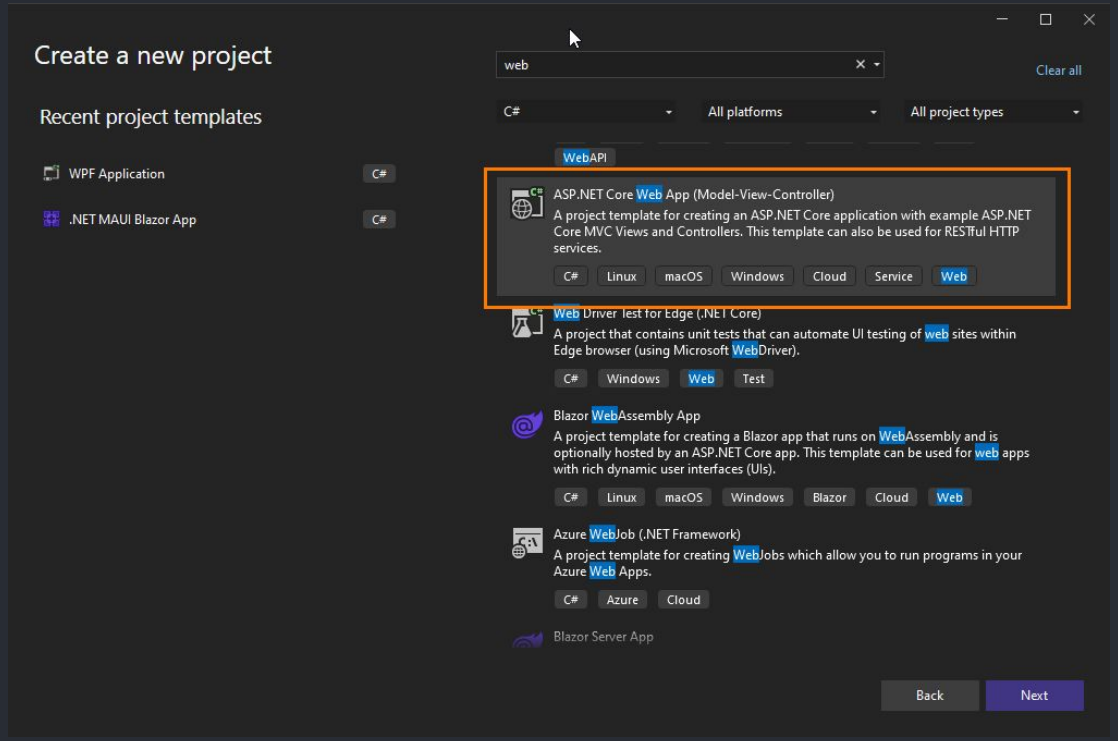
Запрос веб страницы



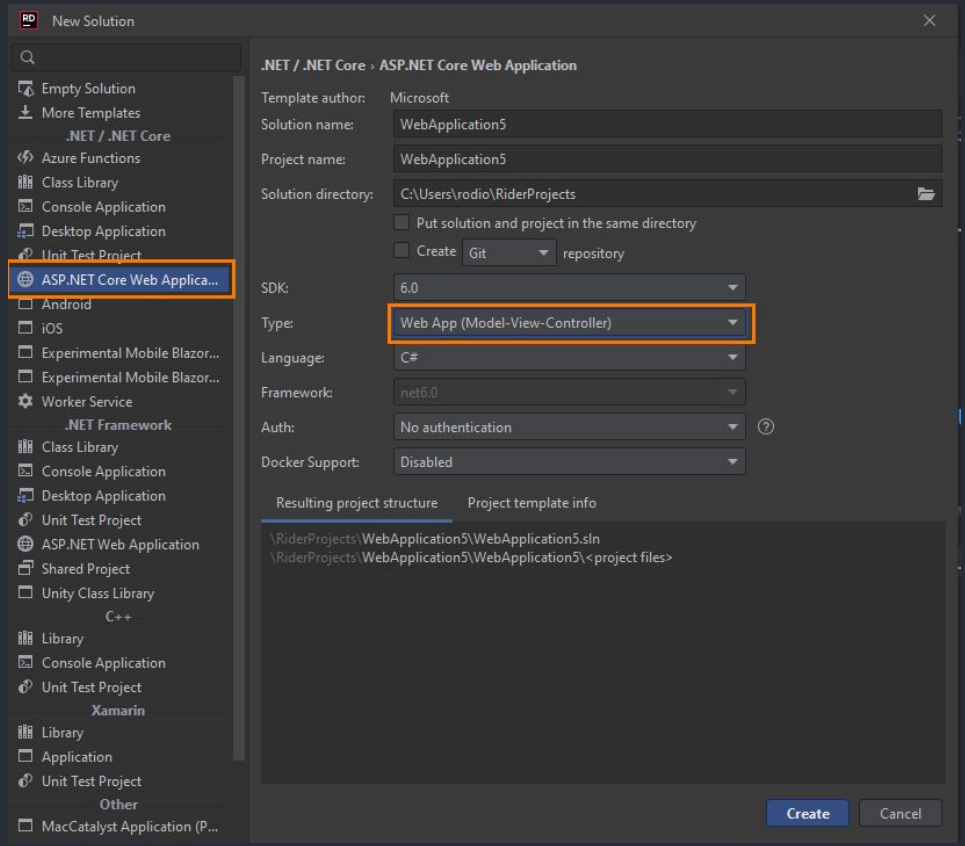
Razor Pages

DEMO

Как создать проект Razor Pages в VS

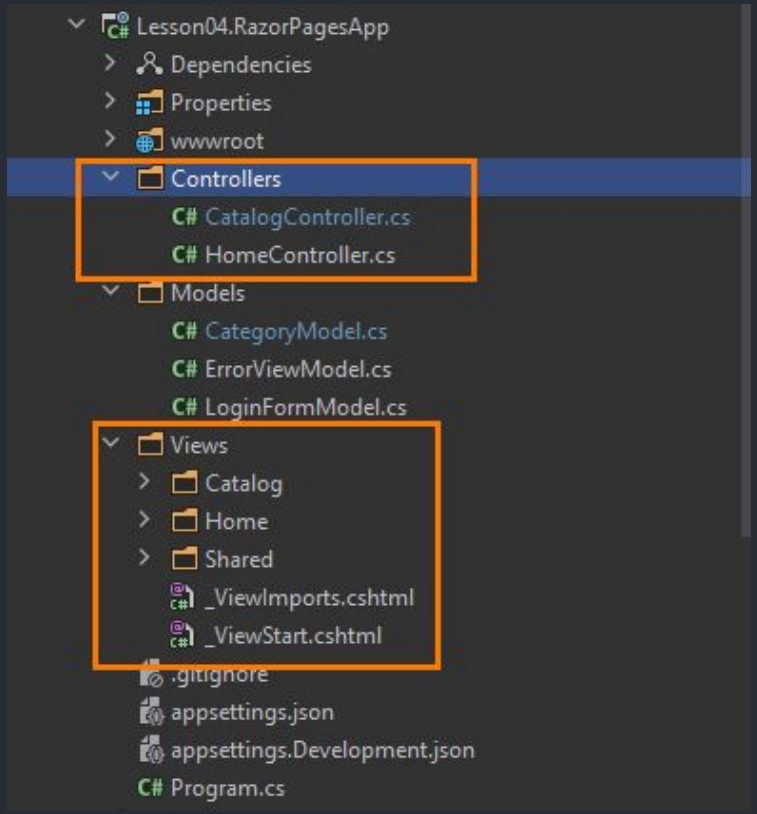


Как создать проект Razor Pages в Rider



MVC

Вспомним структуру проекта в Razor Pages



Паттерн MVC (Model-View-Controller)

- Архитектурный паттерн
- Появился в 1987 в ЯП Smalltalk
- Разделяет приложение на части, еще их называют слои
- Суть паттерна в отделении логики приложения (Model) от вида (View)
- В настоящее время распространен в веб приложениях

https://www.guru99.com/mvc-tutorial.html

Паттерн MVC (Model-View-Controller)

- View – “представление” или “отображение”. Визуальное представление данных.
  - HTML-страница
  - PDF документ
- Model – основная логика и данные приложения
  - База данных
  - Бизнес-логика
  - Домейн
- Controller – посредник между Model и View
  - Обрабатывает входящие запросы
  - Может отвечать за авторизацию
  - Скромные

MVC Example

Подробнее про домейн (domain)

- Домен в архитектуре ПО – это основная область работы компании
- Еще его называет Предметная область
- Бизнес-логикой обычно называют именно логику домена

Параметры страницы

Razor Pages

Пример обработки параметра запроса

/catalog?parentId={id родительской категории}

```
public IActionResult CategoryList(int parentId)
{
    var cats = AllCategories.Where(it => it.parentId == parentId);
    return View(cats);
}
```

Генерация ссылки на страницу через ActionLink

```
http://localhost/catalog/CategoryList?parentId=Id
```

```
@Html.ActionLink(
  $"{category.Name} (Id: {category.Id})", //Текст ссылки
  "CategoryList", //Имя действия
  "Catalog", //Имя контроллера
  new { parentId = category.Id } //Параметры страницы
)
```

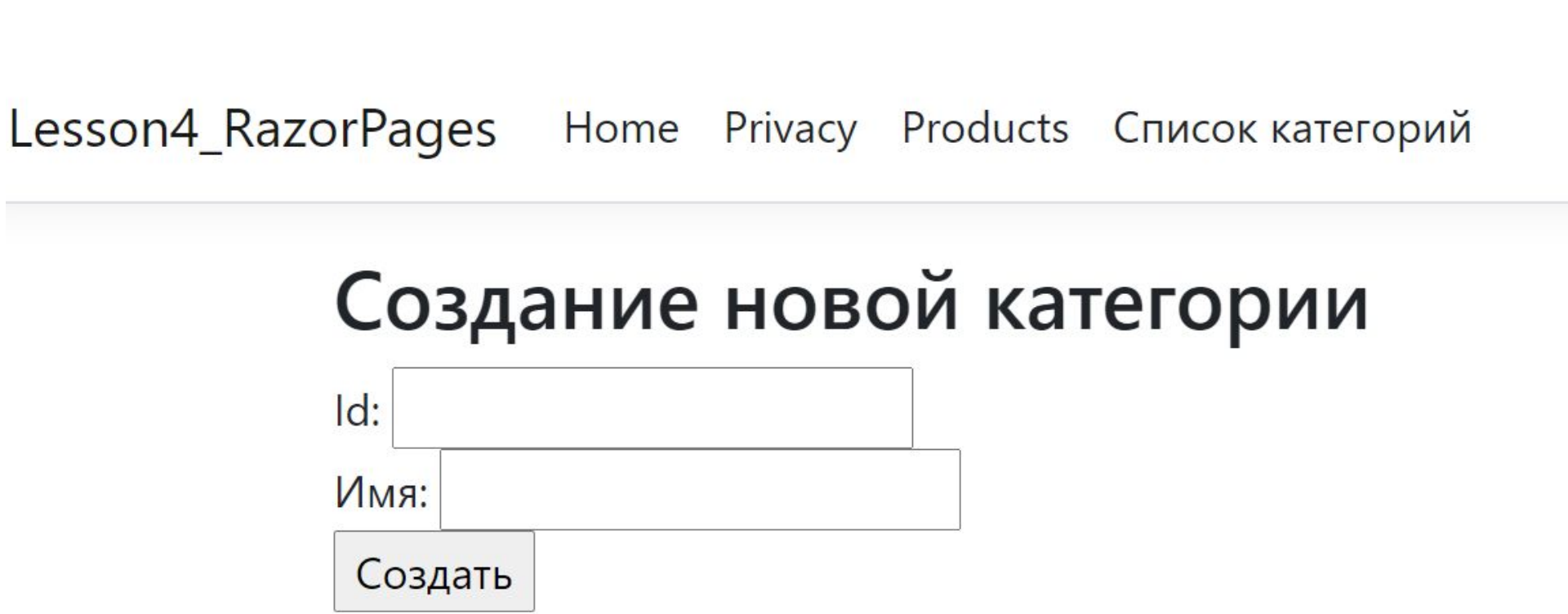
Продукты (Id: 1)

Формы

Razor Pages

Объявление формы

```
<form method="post">
  Id: <input type="text" name="id" required />
  <br/>
  Имя: <input type="text" name="name" />
  <br/>
  <input type="submit" name="submit" value="Создать" />
</form>
```


Lesson4_RazorPages
Home
Privacy
Products
Список категорий
Создание новой категории
Id:
Имя:
Создать

Обработка параметров формы

```
public IActionResult CatalogEditor()
{
    if (HttpContext.Request.Method == "POST")
    {
        var id = HttpContext.Request.Form["id"].ToString();
        var name = HttpContext.Request.Form["name"].ToString();
        categories.AddCategory(id, name);
    }
    return View();
}
```

Через параметры

```
public IActionResult CatalogEditor(int id, string name)
{
    if (HttpContext.Request.Method == "POST")
    {
        categories.AddCategory(id, name);
    }

    return View();
}
```

Через модель

```
[HttpPost]
public IActionResult CategoryAdding([FromForm] CategoryAddingModel model)
{
    if (!ModelState.IsValid)
    {
        return ValidationProblem();
    }
    _categories.Add(new Category(model.Id, model.ParentId, model.Name));
    ViewData["Message"] = "Категория добавлена";
    return View();
}
```

Модель данных

```
public class CategoryAddingModel
{
    [Range(0, long.MaxValue)]
    public long Id { get; set; }

    [Range(0, long.MaxValue)]
    public long ParentId { get; set; }

    [Required]
    public string Name { get; set; } = "";
}
```

Атрибуты валидации из `System.ComponentModel.DataAnnotations`

Атрибут	Устанавливает требование, что значение
`[Required]`	Должно быть задано (не null)
`[Range(MIN, MAX)]`	Должно находится в диапазоне от MIN и до MAX
`[Phone]`	Является номером телефона
`[EmailAddress]`	Является адресом эл. почты
`[Url]`	Является адресом Url
`[StringLength(N)]`	Не должно быть длиннее, чем N
`[RegularExpression]`	Соответствует регулярному выражению

Более полный список по ссылке.

Вопросы

Что проходили?

Что непонятно?

Домашнее задание

1. Создайте новый проект Web App (Model-View-Controller)

2. Выведите список товаров из каталога на странице catalog/products

3. * Добавьте поле с картинкой товара и отобразите товар с картинкой на странице каталога

4. Добавьте страницу добавления новых товаров в каталог.