МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО»

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

дисциплина «Избранные главы информатики»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

«**Создание Web-сервиса с использованием WEB API ASP.NET MVC (ASP.NET Core MVC) для работы с информацией из базы данных**»

Вариант № 16

Выполнил:

студент группы ИП-31, Самусев Р.А.

Принял:

доцент Асенчик О.Д.

Гомель 2018

**Цель работы:** Ознакомиться с возможностями .NET Core по получению, отображению и изменение данных при помощи технологии WEB API.

Github:

**ЗАДАНИЕ**:

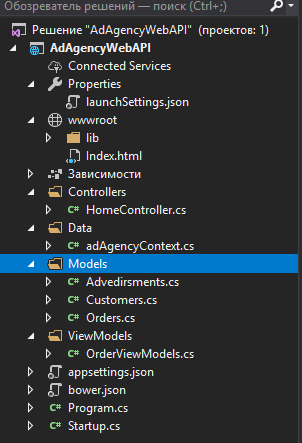
Создать с использованием технологии WEB API Web-приложение, в котором организовать получение, отображение и изменение данных.

Для этого необходимо:

1. Создать (использовать ранее разработанные в предыдущих лабораторных работах):
   1. Классы, моделирующие не менее чем три таблицы базы данных согласно вашему варианту. Одна из таблиц обязательно должна находиться на стороне отношения «многие» связи с другой таблицей в схеме базы данных.
   2. Класс контекста данных для доступа к базе данных.
2. Создать контроллер для операций с данными таблицы, стоящей на стороне отношения «многие».
3. Клиентское приложение в виде HTML страницы для визуального выполнения всех операций с данными выбранной в п.2 таблицы, использующих запросы к методам разработанного контроллера.

**Ход выполнения работы**

**Общий вид проекта**:



**Листинги исходных классов**

Исходные коды классов моделей

Advedirsments.cs

using System.Collections.Generic;

namespace AdAgencyWebAPI.Models

{

public class Advedirsments

{

public Advedirsments() { }

public Advedirsments(string AdType, string AdDescription)

{

this.AdType = AdType;

this.AdDescription = AdDescription;

}

public int AdvedirsmentsId { get; set; }

public string AdType { get; set; }

public string AdDescription { get; set; }

public virtual ICollection<Orders> Orders { get; set; }

}

}

Customers.cs

using System.Collections.Generic;

namespace AdAgencyWebAPI.Models

{

public class Customers

{

public Customers() { }

public Customers(string CustomerName, string CustomerAddress, string CustomerTelephone)

{

this.CustomerName = CustomerName;

this.CustomerAddress = CustomerAddress;

this.CustomerTelephone = CustomerTelephone;

}

public int CustomersId { get; set; }

public string CustomerName { get; set; }

public string CustomerAddress { get; set; }

public string CustomerTelephone { get; set; }

public virtual ICollection<Orders> Orders { get; set; }

}

}

Orders.cs

using System;

namespace AdAgencyWebAPI.Models

{

public class Orders

{

public Orders() { }

public Orders(DateTime DateOrder, DateTime DateBegin, DateTime DateEnd, string Location, int OrderCost)

{

this.DateOrder = DateOrder;

this.DateBegin = DateBegin;

this.DateEnd = DateEnd;

this.Location = Location;

this.OrderCost = OrderCost;

}

public int OrdersId { get; set; }

public DateTime DateOrder { get; set; }

public DateTime DateBegin { get; set; }

public DateTime DateEnd { get; set; }

public string Location { get; set; }

public int OrderCost { get; set; }

public bool PaymentStatus { get; set; }

public int? EmployeeId { get; set; }

public int? CustomersId { get; set; }

public int AdvedirsmentsId { get; set; }

public virtual Advedirsments Advedirsments { get; set; }

public virtual Customers Customers { get; set; }

}

}

OrderViewModel.cs

using System;

namespace AdAgencyWebAPI.ViewModels

{

public class OrderViewModels

{

public int OrdersId { get; set; }

public DateTime DateOrder { get; set; }

public string Location { get; set; }

public int OrderCost { get; set; }

}

}

Класс контекста данных:

using Microsoft.EntityFrameworkCore;

using AdAgencyWebAPI.Models;

namespace AdAgencyWebAPI.Data

{

public class adAgencyContext:DbContext

{

public DbSet<Advedirsments> Advedirsments { get; set; }

public DbSet<Customers> Customers { get; set; }

public DbSet<Orders> Orders { get; set; }

public adAgencyContext(DbContextOptions<adAgencyContext> options)

: base(options)

{

}

}

}

Класс контроллера HomeController.cs

using System;

using System.Collections.Generic;

using System.Linq;

using Microsoft.AspNetCore.Mvc;

using AdAgencyWebAPI.Data;

using AdAgencyWebAPI.Models;

using AdAgencyWebAPI.ViewModels;

using Microsoft.EntityFrameworkCore;

namespace AdAgencyWebAPI.Controllers

{

[Route("api/[controller]")]

public class HomeController : Controller

{

private readonly adAgencyContext \_context;

public HomeController(adAgencyContext context)

{

\_context = context;

}

//GET api/values

[HttpGet]

[Produces("application/json")]

public List<OrderViewModels> Get()

{

var item = \_context.Orders.Include(t => t.Advedirsments).Select(o =>

new OrderViewModels

{

OrdersId = o.OrdersId,

DateOrder = o.DateOrder,

Location = o.Location,

OrderCost = o.OrderCost,

});

return item.OrderBy(o => o.OrdersId).ToList();

}

// GET api/values

[HttpGet("{id}")]

public IActionResult Get(int id)

{

var item = \_context.Orders.FirstOrDefault(o => o.OrdersId == id);

if (item == null)

return NotFound();

return new ObjectResult(item);

}

// POST api/values

[HttpPost]

public IActionResult Post([FromBody]Orders order)

{

if (order == null)

{

return BadRequest();

}

\_context.Orders.Add(order);

\_context.SaveChanges();

return Ok(order);

}

// PUT api/values

public IActionResult Put([FromBody]Orders order)

{

if (order == null)

{

return BadRequest();

}

if (!\_context.Orders.Any(o => o.OrdersId == order.OrdersId))

{

return NotFound();

}

\_context.Update(order);

\_context.SaveChanges();

return Ok(order);

}

// DELETE api/values

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

Orders order = \_context.Orders.FirstOrDefault(o => o.OrdersId == id);

if (order == null)

{

return NotFound();

}

\_context.Orders.Remove(order);

\_context.SaveChanges();

return Ok(order);

}

}

}

Код файла index.html, реализующий получение, отображение и изменение данных

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width" />

<title>Advertising Agency</title>

<link href="lib/bootstrap/dist/css/bootstrap.css" rel="stylesheet" />

<script src="lib/jquery/dist/jquery.js"></script>

</head>

<body>

<h2>Заказы:</h2>

<form name="orderForm">

<input type="hidden" name="ordersId" value="0" />

<div class="form-group">

<label for="dateOrder">Дата заказа:</label>

<input class="form-control" name="dateOrder" />

</div>

<div class="form-group">

<label for="location">Месторасположение:</label>

<input class="form-control" name="location" />

</div>

<div class="form-group">

<label for="orderCost">Стоимость:</label>

<input class="form-control" name="orderCost" />

</div>

<div class="panel-body">

<button type="submit" class="btn btn-sm btn-primary">Сохранить</button>

<a id="reset" class="btn btn-sm btn-primary">Сбросить</a>

</div>

</form>

<table class="table table-condensed table-striped table-bordered">

<thead>

<tr>

<th>Id</th>

<th>Дата заказа</th>

<th>Месторасположения</th>

<th>Цена услуги</th>

<th></th>

</tr>

</thead>

<tbody></tbody>

</table>

<script>

// Получение списка всех заказов

function Getorders() {

$.ajax({

url: '/api/Home',

type: 'GET',

contentType: "application/json",

success: function (orders) {

var rows = "";

$.each(orders, function (index, order) {

// добавляем полученные элементы в таблицу

rows += row(order);

})

$("table tbody").append(rows);

}

});

}

// Получение одного заказа

function Getorder(id) {

$.ajax({

url: '/api/Home/' + id,

type: 'GET',

contentType: "application/json",

success: function (order) {

var form = document.forms["orderForm"];

form.elements["ordersId"].value = order.ordersId;

form.elements["dateOrder"].value = order.dateOrder;

form.elements["location"].value = order.location;

form.elements["orderCost"].value = order.orderCost;

}

});

}

// Добавление заказа

function CreateOrder(dateOrder, location, orderCost) {

$.ajax({

url: "api/Home",

contentType: "application/json",

method: "POST",

data: JSON.stringify({

dateOrder: dateOrder,

location: location,

orderCost: orderCost

}),

success: function (order) {

reset();

reset();

$("table tbody").append(generateRow(order));

}

})

}

// Изменение заказа

function Editorder(ordersId, dateOrder, location, orderCost) {

$.ajax({

url: "api/Home",

contentType: "application/json",

method: "PUT",

data: JSON.stringify({

ordersId: ordersId,

dateOrder: dateOrder,

location: location,

orderCost: orderCost

}),

success: function (order) {

reset();

$("tr[data-rowid='" + order.ordersId + "']").replaceWith(row(order));

}

})

}

// сброс формы

function reset() {

var form = document.forms["orderForm"];

form.reset();

form.elements["ordersId"].value = 0;

}

// Удаление заказа

function Deleteorder(id) {

$.ajax({

url: "api/Home/" + id,

contentType: "application/json",

method: "DELETE",

success: function (order) {

$("tr[data-rowid='" + order.ordersId + "']").remove();

}

})

}

// создание строки для таблицы

var row = function (order) {

return "<tr data-rowid='" + order.ordersId + "'><td>" + order.ordersId + "</td>" +

"<td>" + order.dateOrder + "</td>" +

"<td>" + order.location + "</td>" +

"<td>" + order.orderCost + "</td>" +

"<td><a class='editLink' data-id='" + order.ordersId + "'>Изменить</a> | " +

"<a class='removeLink' data-id='" + order.ordersId + "'>Удалить</a></td></tr>";

}

// сброс значений формы

$("#reset").click(function (e) {

e.preventDefault();

reset();

})

// отправка формы

$("form").submit(function (e) {

e.preventDefault();

var ordersId = this.elements["ordersId"].value;

var dateOrder = this.elements["dateOrder"].value;

var location = this.elements["location"].value;

var orderCost = this.elements["orderCost"].value;

if (ordersId == 0)

CreateOrder(dateOrder, location, orderCost);

else

Editorder(ordersId, dateOrder, location, orderCost);

});

// нажатие на ссылку Изменить

$("body").on("click", ".editLink", function () {

var id = $(this).data("id");

Getorder(id);

})

// нажатие на ссылку Удалить

$("body").on("click", ".removeLink", function () {

var id = $(this).data("id");

Deleteorder(id);

})

// загрузка данных для таблицы и ее формирование

Getorders();

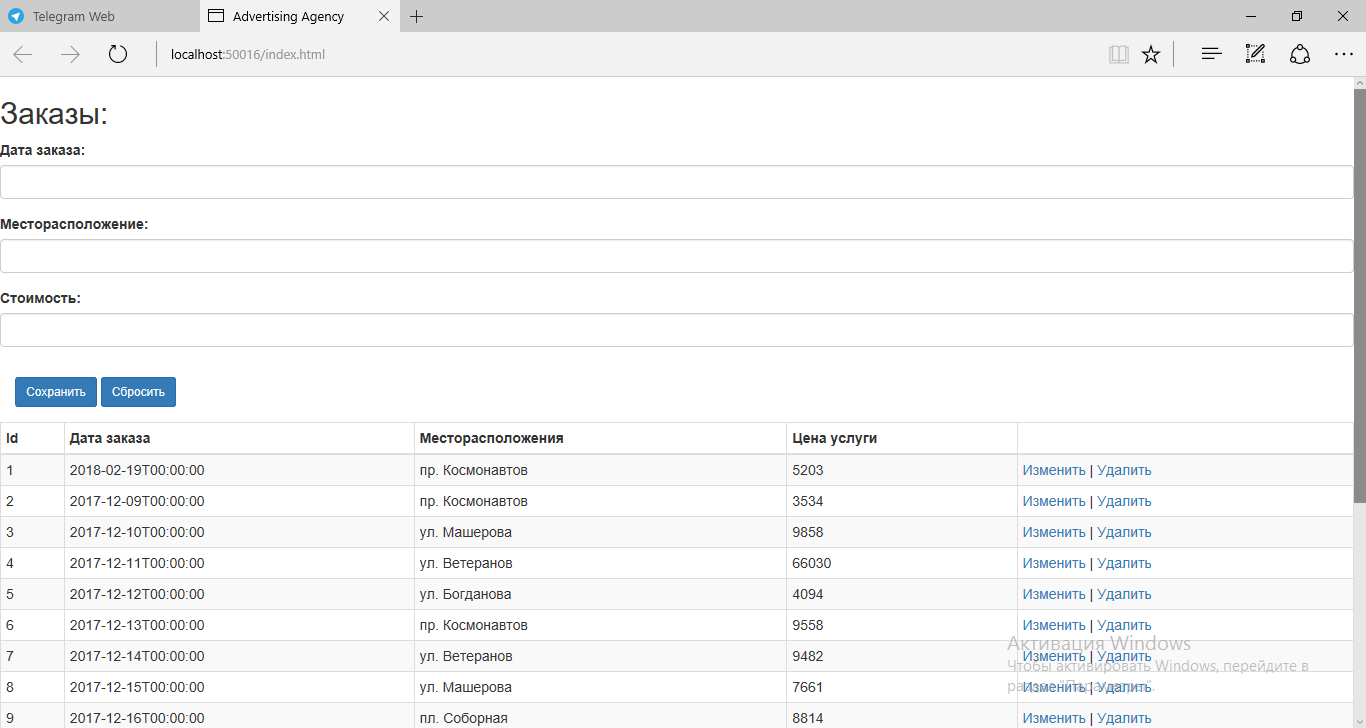
</script>

</body>

</html>

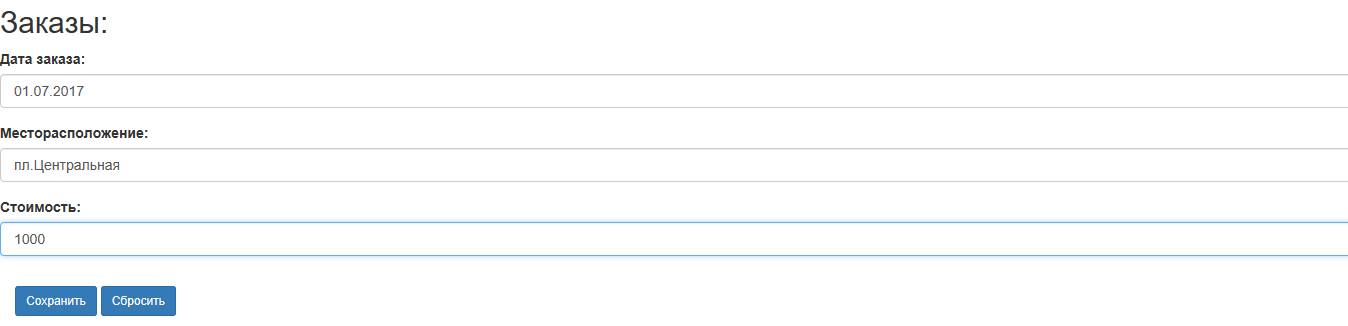
**Результаты выполнения запросов**

Получение всех записей при загрузке страницы:

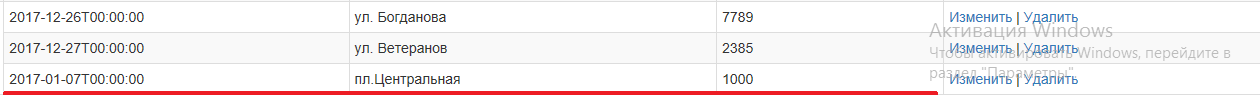


**Добавление новой записи**:

Заполняем форму, нажимаем «Сохранить» и обновляем страницу



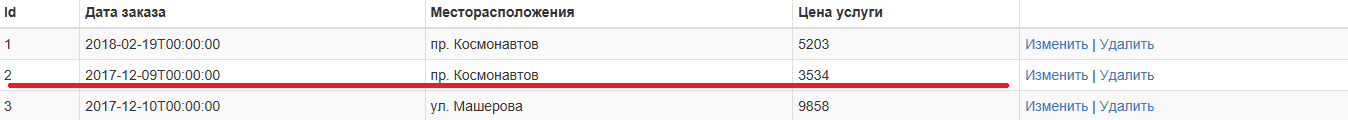
Запись добавилась в конец таблицы



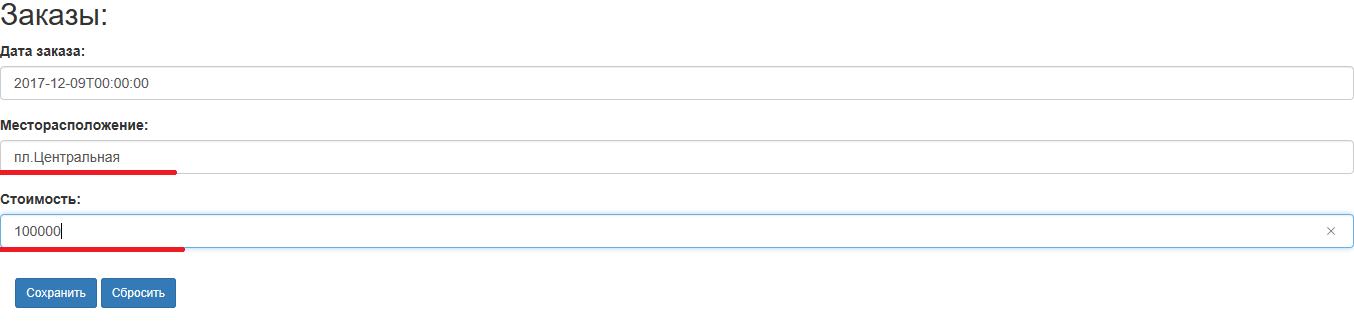
Как видим, запись добавляется правильно

**Изменение записи**:

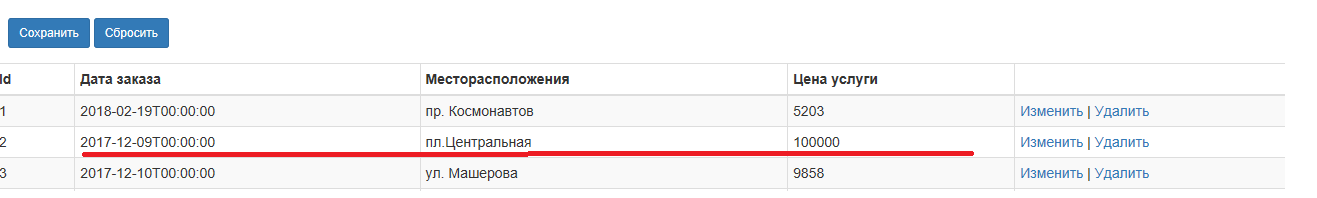
Изменим запись под номером 2



Изменим адрес на «пл.Центральная» и цену услуги на «100000»



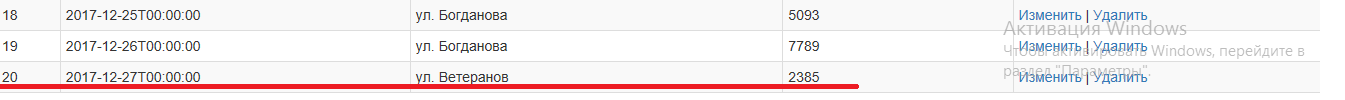
Нажимаем «Сохранить»



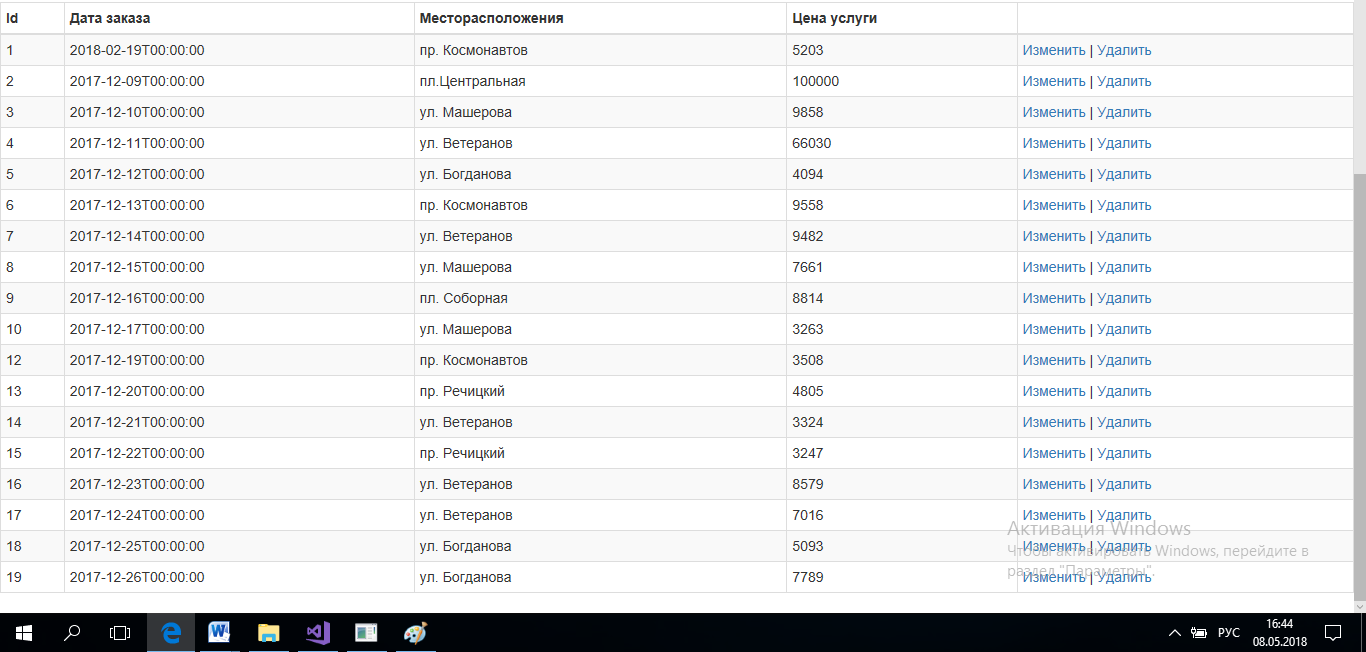
Как видим, запись изменилась верно

**Удаление записи:**

Удалим последнюю запись с id 20



После удаления таблица выглядит следующим образом:



Как видим, последняя запись таблицы с id равным 19, а значит удаление прошло успешно

**Вывод:** в ходе выполнения данной лабораторной работы, были получены навыки в возможностях .NET Core по получению, отображению и изменение данных при помощи технологии WEB API