

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Смоленский государственный университет»

Кафедра прикладной математики и информатики

«Утверждаю»
Проректор по учебно-методической
работе
_____ Ю.А. Устименко
«23» июня 2022 г.

**Рабочая программа дисциплины
Б1.В.ДВ.04.02 Разработка web-сервисов**

Направление подготовки: **09.03.03 Прикладная информатика**
Направленность (профиль): **Прикладная информатика в логистике**
Форма обучения: очная
Курс – 4
Семестр – 8
Всего зачетных единиц – 2, часов – 72
Форма отчетности: зачет – 8 семестр

Программу разработал
кандидат технических наук, доцент Т.А.Самойлова

Одобрена на заседании кафедры
«16» июня 2022 г., протокол № 10

Заведующий кафедрой _____

Смоленск
2022

1. Место дисциплины в структуре ОП

Дисциплина «Разработка web – сервисов» относится к дисциплинам по выбору и является вспомогательной для производственной практики студентов на предприятиях. Она изучается в 8 семестре.

При изучении данной дисциплины необходимы компетенции студентов, сформированные при изучении таких дисциплин, как «Проектирование информационных систем», «Программная инженерия», «Базы данных», «Информационные системы и технологии в логистике». Рассматриваются возможные подходы к разработке веб-сервисов, предназначенных для построения корпоративных информационных систем современных организаций и предприятий, обеспечивающих возможность удаленного доступа к данным и коллективной работы сотрудников. Дисциплина «Разработка web – сервисов» обеспечивает проведение технологической практики студентов на предприятиях.

2. Планируемые результаты обучения по дисциплине

Компетенция	Индикаторы достижения
ПК-2. Способен проводить описание прикладных процессов и информационного обеспечения и проектировать информационные системы в логистике	Знать: основные принципы и методы описания и анализа прикладной области, информационных потребностей, формирования требований к информационным системам, методы формализации и структурирования данных, основные методы и технологии проектирования информационных систем, возможности типовых ИС, архитектуру, устройство и функционирование вычислительных сетей, коммуникационное оборудование и сетевые протоколы, теорию баз данных и основы программирования. Уметь: проводить анализ предметной области, выявлять информационные потребности и разрабатывать требования к информационным системам, формализовывать и структурировать полученную информацию, осуществлять сравнительный анализ и выбор информационно-коммуникационной технологии для решения поставленных задач, проектировать информационные системы. Владеть: навыками сбора и анализа информации, необходимой для решения поставленных производственных задач, навыками по формализации и структурированию данных, навыками работы с прикладным программным обеспечением для проектирования современных информационных систем.
ПК-3. Способен создавать прототипы логистических информационных систем, разрабатывать программный код информационной системы и баз данных информационной системы для управления логистическими процессами, создавать прикладное программное обеспечение.	Знать: современные языки программирования, их синтаксис, языки программирования и работы с базами данных, теорию баз данных, инструменты и методы тестирования характеристик ИС и прототипирования пользовательского интерфейса, возможности типовой ИС, ее устройство и функционирование, основы современных операционных систем, систем

	<p>управления базами данных.</p> <p>Уметь: кодировать на языках программирования, тестировать результаты прототипирования, тестировать разрабатываемую ИС (модульное, интеграционное тестирование), обнаруживать и устранять несоответствия и дефекты.</p> <p>Владеть: навыками по созданию прикладного программного обеспечения, разработке прототипов ИС, разработке кода ИС и баз даны ИС, тестирования ИС, устранения обнаруженных несоответствий и дефектов.</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3. Содержание дисциплины

В дисциплине «Разработка web – сервисов» рассматриваются следующие темы.

1. **Технологии разработки SOA. Платформы и средства создания веб - сервисов.** Понятие Service-Oriented Architecture (SOA). Технологии разработки SOA. Типы SOA. Причины появления. Платформы и средства создания веб-сервисов. Примеры информационных систем с использованием веб - сервисов. Протоколы веб - сервисов. Стандарт SOAP - протокол взаимодействия сервисов. Веб - сервис REST. HTTP-запросы в сервисах REST для реализации CRUD.
2. **Протокол REST. Использование служб REST.** Структура протокола HTTP. Использование HTTP-методов Get(), Post(), Put() и Delete() для обработки данных в сервисах REST. ORM-технологии. Преимущества ORM-технологий. Современные веб - сервера, их роль в работе веб - приложений. Примеры серверов. Примеры проектов с использованием веб-сервисов.
3. **Кроссплатформенное программное обеспечение SOA.** Поставляемое для SOA кроссплатформенное программное обеспечение: IBM, Microsoft, 1C, SAP. Основные требования к модели SOA. Проблемы разработки: организационные, экономические, ресурсные, технические. Ошибки в SOA - проектах.
4. **Среда разработки веб - сервисов ASP.NET Core.** Отличия от предыдущих версий ASP.NET. Основные типы проектов. Технология Core MVC. Маршрутизация в ASP.NET Core MVC. Модели и контроллеры MVC. Функции контроллера. Методы действий контроллера. Обработка данных пользователя средствами контроллера. Запросы HTTP GET и POST в контроллере. Доступ к данным в Core NET. MVC. Методы контроллера для выполнения CRUD. Пример проекта.
5. **Платформа NET Core MVC Entity Framework.** Преимущества. Модель Entity Data Model. Класс DbContext. Методы DbContext для управления запросами: выборка, добавление, изменение, удаление. Реализация SQL- запросов средствами Entity Framework. LINQ to Entities. Разработка контроллера для фильтрации данных. Разработка контроллера для групповых запросов.
6. **Технология MS WebAPI Core. Методы контроллеров VS WEBAPI.** Возвращаемые типы данных. Сериализация, десериализация. Пример WEBAPI – проекта для доступа к данным. Клиентские приложения баз данных, использующие службы REST. Средства обращения к методам службы из клиентских приложений. Базовые классы для отправки HTTP-запросов и получения HTTP-ответов. Публикация сервисов WebAPI на some.com.
7. **Доступ с веб - сервисам в десктоп, мобильных и веб-приложениях.** Информационные системы с использованием сервисов REST API и SOAP. Одностраничные клиентские приложения. Клиент Blazor WebAssembly. Доступ к WEBAPI из Blazor WebAssembly. CORS для браузерных клиентов. Асинхронная модель программирования. Примеры клиентских приложений, использующих службы REST для доступа к данным.

8. **Веб - сервисы gRPC.** Применение. Прото – файлы. Кодирование сообщений с помощью Protocol Buffers. Объекты реализации сервиса. Подключение gRPC в классах проекта. Клиенты gRPC. Стабы. Генерация прокси - сервера. Примеры проектов.
9. **Веб - сервисы с протоколом доступа OData.** Применение. Возможности. Документ метаданных OData. Операции отбора. Клиенты OData. Установка объекта OData Connected Service. Генерация прокси-службы в приложении клиента. Браузерный клиент Blazor Web Assembly для доступа к веб – сервису с помощью протокола OData. Библиотеки для работы с OData. Перспективы OData. Примеры проектов.

4. Тематический план

№ п/п	Разделы и темы	Всего часов	Формы занятий			
			лекции	практические занятия	лабораторные занятия	самостоятельная работа
1	Технологии разработки SOA. Платформы и средства создания веб - сервисов.	8	2	–	4	2
2	Протокол REST. Использование служб REST.	8	2	–	4	2
3	Кроссплатформенное программное обеспечение SOA.	8	2	–	4	2
4	Среда разработки веб - сервисов ASP.NET Core.	8	2	–	4	2
5	Платформа NET Core MVC Entity Framework.	8	2	–	4	2
6	Технология MS WebAPI Core. Методы контроллеров VS WebAPI.	8	2	–	4	2
7	Доступ с веб - сервисам в десктоп, мобильных и веб-приложениях.	8	2	–	4	2
8	Веб - сервисы gRPC.	8	2		4	2
9	Веб - сервисы с протоколом доступа OData.	8	2		4	2
ИТОГО		72	18	–	36	18

5. Виды образовательной деятельности

Лекции

1. **Технологии разработки SOA. Платформы и средства создания веб - сервисов.** Понятие о Service-Oriented Architecture (SOA). Технологии разработки SOA. Типы SOA. Причины появления. Платформы и средства создания веб - сервисов. Примеры информационных систем с использованием веб - сервисов. Протоколы веб - сервисов. Стандарт SOAP - протокол взаимодействия сервисов. Веб - сервис REST. HTTP-запросы в сервисах REST для реализации CRUD.
2. **Протокол REST. Использование служб REST.** Структура протокола HTTP. Использование HTTP-методов Get(), Post(), Put() и Delete() для обработки данных в

сервисах REST. ORM-технологии. Преимущества ORM-технологий. Современные веб - сервера, их роль в работе веб - приложений. Примеры серверов. Примеры проектов с использованием веб-сервисов.

3. **Кроссплатформенное программное обеспечение SOA.** Поставляемое для SOA кроссплатформенное программное обеспечение: IBM, Microsoft, 1C, SAP. Основные требования к модели SOA. Проблемы разработки: организационные, экономические, ресурсные, технические. Ошибки в SOA - проектах.
4. **Среда разработки веб - сервисов ASP.NET Core.** Отличия от предыдущих версий ASP.NET. Основные типы проектов. Технология Core MVC. Маршрутизация в ASP.NET Core MVC. Модели и контроллеры MVC. Функции контроллера. Методы действий контроллера. Обработка данных пользователя средствами контроллера. Запросы HTTP GET и POST в контроллере. Доступ к данным в Core NET. MVC. Методы контроллера для выполнения CRUD. Пример проекта.
5. **Платформа NET Core MVC Entity Framework.** Преимущества. Модель Entity Data Model. Класс DbContext. Методы DbContext для управления запросами: выборка, добавление, изменение, удаление. Реализация SQL- запросов средствами Entity Framework. LINQ to Entities. Разработка контроллера для фильтрации данных. Разработка контроллера для групповых запросов.
6. **Технология MS WebAPI Core. Методы контроллеров VS WebAPI.** Технология MS WebAPI Core. Методы контроллеров VS WEBAPI. Возвращаемые типы данных. Сериализация, десериализация. Пример WEBAPI – проекта для доступа к данным. Клиентские приложения баз данных, использующие службы REST. Средства обращения к методам службы из клиентских приложений. Базовые классы для отправки HTTP-запросов и получения HTTP-ответов. Публикация сервисов WebAPI на some.com.
7. **Доступ с веб - сервисам в десктоп, мобильных и веб-приложениях.** Информационные системы с использованием сервисов REST API и SOAP. Одностраничные клиентские приложения. Клиент Blazor WebAssembly. Доступ к WEBAPI из Blazor WebAssembly. CORS для браузерных клиентов. Асинхронная модель программирования. Примеры клиентских приложений, использующих службы REST для доступа к данным.
8. **Веб - сервисы gRPC.** Применение. Прото – файлы. Кодирование сообщений с помощью Protocol Buffers. Объекты реализации сервиса. Подключение gRPC в классах проекта. Клиенты gRPC. Стабы. Генерация прокси - сервера. Примеры проектов.
9. **Веб - сервисы с протоколом доступа OData.** Применение. Возможности. Документ метаданных OData. Операции отбора. Клиенты OData. Установка объекта OData Connected Service. Генерация прокси-службы в приложении клиента. Браузерный клиент Blazor Web Assembly для доступа к веб – сервису с помощью протокола OData. Библиотеки для работы с OData. Перспективы OData. Примеры проектов.

Лабораторные работы

№1-2. Создание простейшей веб-службы REST с помощью ASP.NET Web API Core фреймворка.

№3-4. Создание веб-службы REST для передачи изображений с помощью ASP.NET Web API Core фреймворка.

№5-6. Разработка клиентских приложений для чтения изображений.

№7-8. Создание веб-службы REST для передачи файлов с помощью ASP.NET Web API Core фреймворка.

№9-10. Разработка клиентских приложений для чтения файлов.

№11-12. Создание WebAPICore веб-службы REST с использованием EntityFramework для обработки базы данных.

№13-14. Создание клиентских приложений для созданной с помощью EntityFrameworkWebAPI веб – службы.

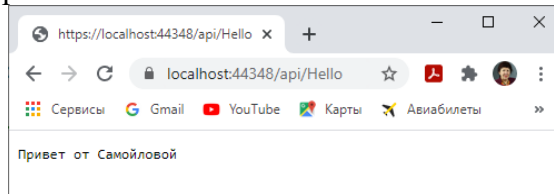
№15-16. Создание веб-службы gRPC с помощью ASP.NETCore.

№17-18. Создание веб - службы REST с протоколом доступа OData.

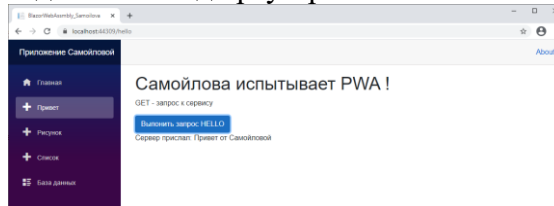
Задания для лабораторных занятий

Лабораторная работа 1-2. "Создание простейшей веб-службы REST с помощью ASP.NET WebAPICore фреймворка»

Задание 1. Создайте WEBAPICore – сервис для передачи клиенту простого сообщения. Результат тестирования запроса GET:

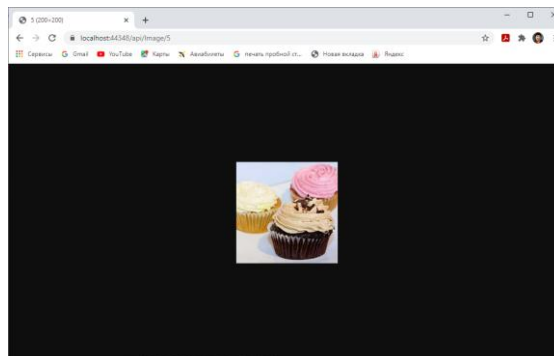


Задание 2. Создайте браузерный клиент BlazorWebAssembly для вызова GET- метода WEBAPI - сервиса, разработанного в задании 1. Вид браузера клиента:



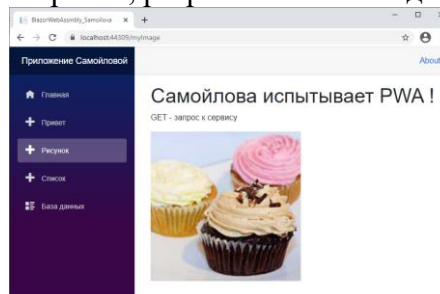
Лабораторная работа 3-4. "Создание веб-службы REST для передачи изображений с помощью ASP.NET WebAPICore фреймворка»

Задание 1. Создайте WEBAPICore – сервис для передачи клиенту изображения. Результат вызова метода:

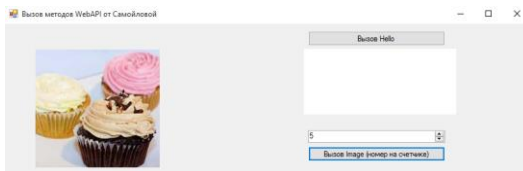


Лабораторная работа 5-6. "Разработка клиентских приложений для чтения изображений»

Задание 1. Разработайте браузерный клиент BlazorWebAssembly, содержащий страницу для вызова GET- методов WEBAPI - сервиса, разработанных в задании 1. Вид браузера клиента:

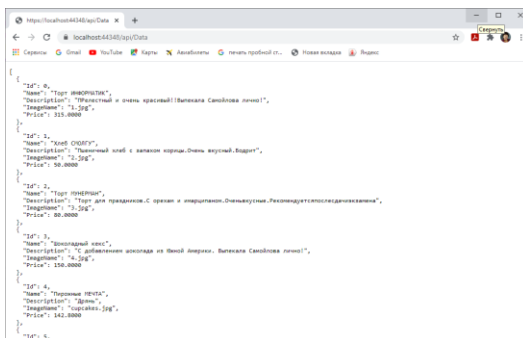


Задание 2. Создайте WindowsForms - клиента для вызова GET- метода WEBAPI – сервиса для передачи изображения. Вид клиента при вызовах метода службы:



Лабораторная работа 7-8. "Создание веб-службы REST для передачи файлов с помощью ASP.NET WebAPICore фреймворка»

Задание . Создайте WEBAPICore – сервис для передачи клиенту файла JSON. Результат вызова метода:

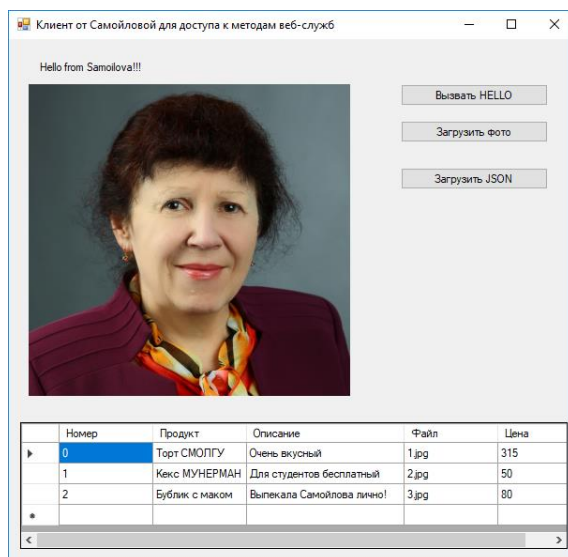


Лабораторная работа 9-10. "Разработка клиентских приложений для чтения файлов»

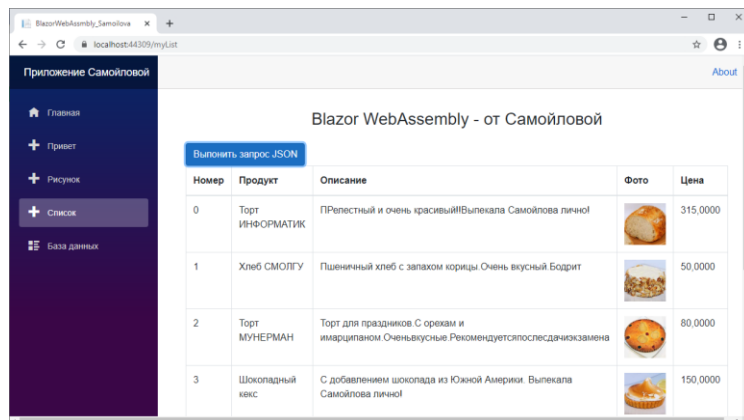
Задание 1. Создайте десктоп - приложение для вызова методов веб - сервиса и отправки клиенту:

- простого сообщения;
- изображения;
- файла JSON с табличными данными.

Вид формы клиента:

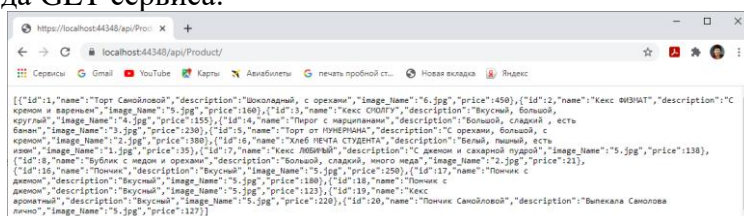


Задание 2. . Создайте браузерный клиент **BlazorWebAssembly** для вызова методов веб - сервиса, разработанных в ЛБ 5. Вид браузера клиента



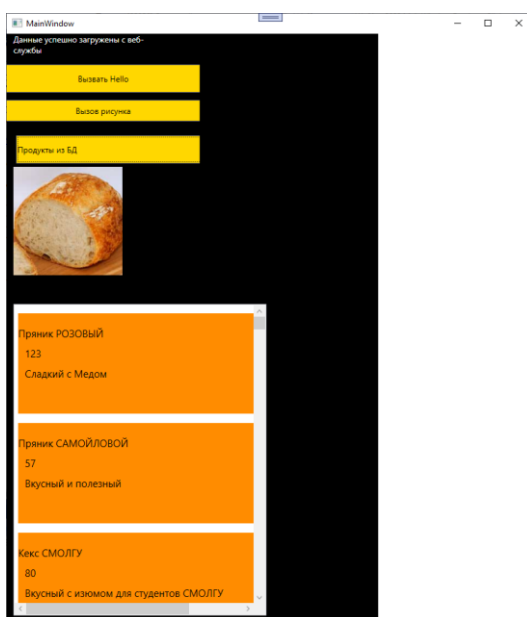
Лабораторная работа 11-12. «Создание WebAPICore веб-службы REST с использованием EntityFramework для обработки базы данных»

Задание 1. Создайте WEBAPICore – сервис для обработки таблицы базы данных (имя таблицы содержит вашу фамилию). Реализуйте GET, POST, PUT, DELETE запросы к таблице для выполнения SQL- команд SELECT, INSERT, UPDATE, DELETE. Используйте EntityFramework. Пример вызова метода GET сервиса:

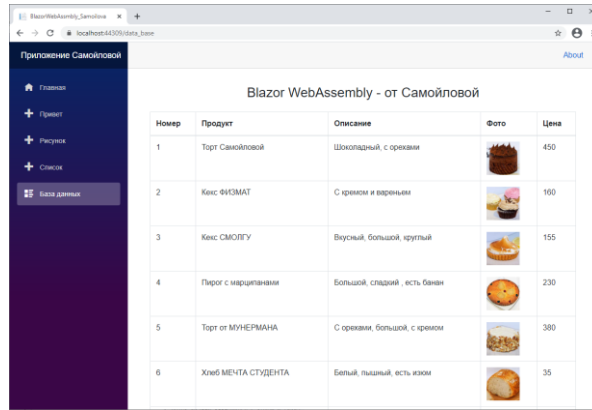


Лабораторная работа 13-14. «Создание клиентских приложений для созданной с помощью EntityFrameworkWebAPI веб – службы».

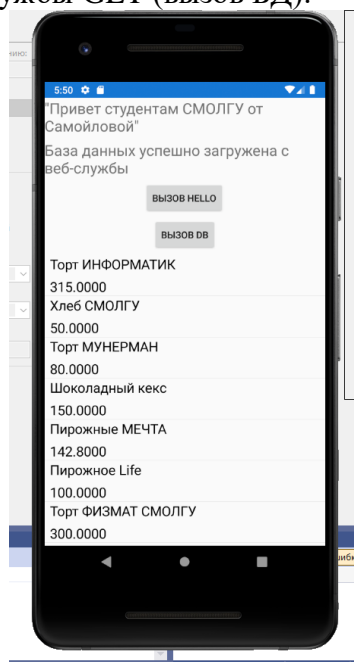
Задание 1. Создайте WPF - клиента, работающего на платформе Xamarin, для вызова методов веб - сервиса, разработанных в ЛБ 6. Форма клиента WPF:



Задание 2. . Создайте браузерный клиент BlazorWebAssembly для вызова методов веб - сервиса, разработанных в ЛБ 6. Вид браузера клиента

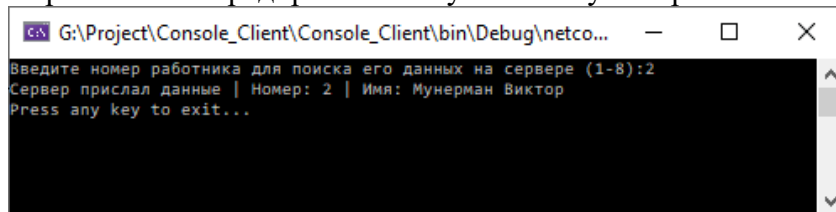


Задание 3. Создайте Android - клиента для вызова GET - метода WEBAPICore- сервиса, разработанного в ЛБб. Предварительно службу разместите на веб - хостинге или воспользуйтесь веб – службой преподавателя. Адрес - <http://www.samoilova.somee.com/api/data>. Вид клиента при вызове метода службы GET (вызов БД):

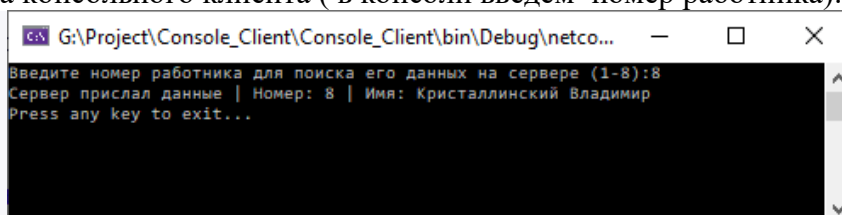


Лабораторная работа 15-16. «Создание веб-службы gRPC с помощью ASP.NETCore».

Задание 1. Создайте gRPCCore – приложение для передачи клиенту элемента коллекции, содержащей список работников предприятия. Результат запуска приложения клиента:

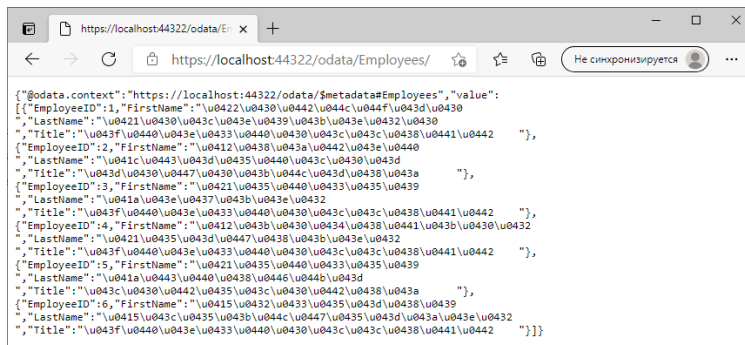


Задание 2. Создайте клиента для gRPCCore – сервиса для приема элемента коллекции. Результат запуска консольного клиента (в консоли введем номер работника):



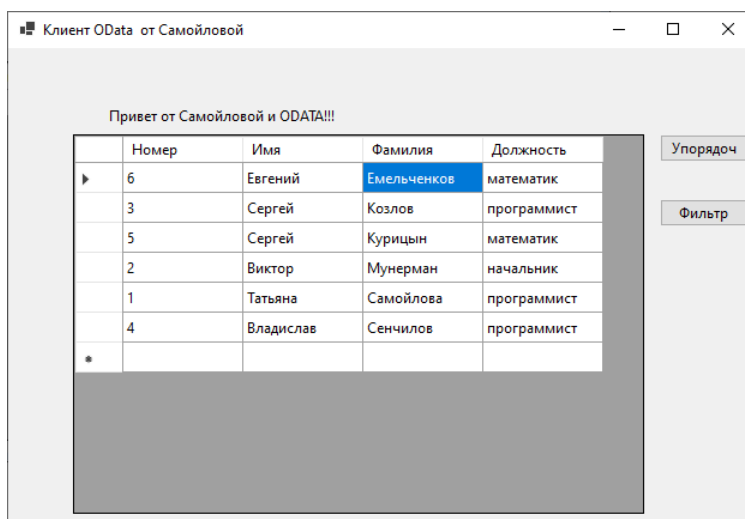
Лабораторная работа 17-18. "Создание веб - службы REST с протоколом доступа OData".

Задание 1. Создайте конечную точку OData веб – сервиса с помощью Core WEB-API ASP.NET. Разработайте GET, POST, PUT, DELETE - методы контроллера для обработки данных таблицы Employees = {EmployeeID, FirstName, LastName, Title}. Доступ к данным выполните средствами Entity Framework, включающей ручное создание модели. Пример выполнения GET-запроса к OData - веб-сервису (русские буквы не распознаны браузером, но клиентскому приложению они понятны):



```
["@odata.context": "https://localhost:44322/odata/$metadata#Employees", "value": [{"EmployeeID": 1, "FirstName": "\u0422\u0430\u0442\u044c\u044f\u0430\u0431\u043e", "LastName": "\u0421\u0430\u043c\u043e\u0439\u043b\u043e\u0432\u0430", "Title": "\u0430\u0434\u0430\u043c\u0438\u043d\u0438\u0441\u0442\u0440\u0430\u043d\u0438\u0439"}, {"EmployeeID": 2, "FirstName": "\u0412\u0438\u043a\u0442\u043e\u0440", "LastName": "\u043c\u0443\u043d\u0435\u0440\u043c\u0430\u043d", "Title": "\u043d\u0430\u0447\u0430\u043b\u043d\u0438\u043a"}, {"EmployeeID": 3, "FirstName": "\u0421\u0435\u0440\u0433\u0435\u0439", "LastName": "\u041a\u043e\u0437\u043b\u043e\u0432", "Title": "\u043f\u0440\u043e\u0433\u0440\u0430\u043c\u043c\u0438\u0441\u0442"}, {"EmployeeID": 4, "FirstName": "\u0412\u043b\u0430\u0434\u0438\u0441\u043b\u0430\u0432", "LastName": "\u0421\u0435\u043d\u0447\u0438\u043b\u043e\u0432", "Title": "\u043f\u0440\u043e\u0433\u0440\u0430\u043c\u043c\u0438\u0441\u0442"}, {"EmployeeID": 5, "FirstName": "\u0421\u0435\u0440\u0433\u0435\u0439", "LastName": "\u041a\u0443\u0440\u0438\u0446\u0438\u043d", "Title": "\u043c\u0430\u0442\u0435\u043c\u0430\u0442\u0438\u043a"}, {"EmployeeID": 6, "FirstName": "\u0415\u0432\u0433\u0435\u043d\u0438\u0439", "LastName": "\u0415\u043c\u0435\u043b\u044c\u0447\u0435\u043d\u043a\u043e\u0432", "Title": "\u043c\u0430\u0442\u0435\u043c\u0430\u0442\u0438\u043a"}]}
```

Задание 2. Создайте клиентское приложение WindowsForms для доступа к веб – сервису с помощью протокола OData. Разработайте GET – запрос для получения всех отсортированных по фамилии данных таблицы Employees = {EmployeeID, FirstName, LastName, Title}. Доступ к данным выполните средствами прокси – службы. Пример выполнения GET-запроса к веб – сервису OData:



Задания для лабораторных работ, размещены в системе дистанционного обучения СмолГУ (www.moodle.smolgu.ru). На занятиях для каждой работы задание предоставляется студентам в электронном виде.

Самостоятельная работа

Текущая самостоятельная работа студента направлена на углубление и закрепление знаний студентов, развитие практических умений. Она заключается в работе с лекционными материалами, поиске и обзоре литературы и электронных источников, информации по заданным темам курса, опережающей самостоятельной работе, в изучении тем, вынесенных на самостоятельную проработку, подготовке к лабораторным занятиям.

Самостоятельная внеаудиторная работа студентов включает:

- проработку лекционного материала, составление конспекта лекций по темам, вынесенным на самостоятельное изучение;
- выполнение домашних заданий;
- подготовку к защите лабораторных работ.

Темы для самостоятельного изучения

1. История развития SOA -программирования и веб-сервисов.
2. Среды разработки веб-сервисов.
3. Библиотеки разработок веб-сервисов в средах VisualStudio и Django (Flask).
4. Доступ сервисов к SQL-Server и PostgreSQL.
5. Технология WPF для клиентских десктоп - приложений.
6. Технология Web – Assembly для клиентских веб - приложений.

Консультирование студентов осуществляться в индивидуальном порядке на занятиях и во внеурочное время. Выполнение самостоятельной работы оценивается по электронным материалам, подготовленным студентами. Результаты деятельности накапливаются в индивидуальных портфолио студентов.

Учебно-методическое обеспечение для самостоятельной работы

1. Интернет-Университет Информационных Технологий, <http://www.intuit.ru>
2. Каталог образовательных Internet-ресурсов, <http://window.edu.ru>.
3. Библиотека разработчика Microsoft, <http://msdn.microsoft.com> .

6. Критерии оценивания результатов освоения дисциплины (модуля)

6.1. Оценочные средства и критерии оценивания для текущей аттестации

Теоретические вопросы

1. Понятие Service-Oriented Architecture (SOA).
2. Технологии разработки SOA. Типы SOA. Причины появления. Платформы и средства создания веб - сервисов.
3. Примеры информационных систем с использованием веб – сервисов.
4. Протоколы веб - сервисов. Стандарт SOAP - протокол взаимодействия сервисов.
5. Веб - сервис REST. HTTP-запросы в сервисах REST для реализации CRUD.
6. Протокол REST. Использование служб REST.
7. Структура протокола HTTP. Использование HTTP-методов Get(), Post(), Put() и Delete() для обработки данных в сервисах REST.
8. ORM-технологии. Преимущества ORM-технологий.
9. Современные веб - сервера, их роль в работе веб - приложений. Примеры серверов.
10. Примеры проектов с использованием веб-сервисов.
11. Кроссплатформенное программное обеспечение SOA. Поставляемое для SOA кроссплатформенное программное обеспечение: IBM, Microsoft, 1C, SAP.
12. Основные требования к модели SOA. Проблемы разработки: организационные, экономические, ресурсные, технические. Ошибки в SOA - проектах.
13. Среда разработки веб - сервисов ASP.NET Core. Отличия от предыдущих версий ASP.NET. Основные типы проектов.
14. Технология Core MVC. Маршрутизация в ASP.NET Core MVC. Модели и контроллеры MVC. Функции контроллера. Методы действий контроллера. Обработка данных пользователя средствами контроллера.
15. Запросы HTTP GET и POST в контроллере. Доступ к данным в Core NET. MVC.
16. Методы контроллера для выполнения CRUD. Пример проекта.
17. Платформа NET Core MVC Entity Framework. Преимущества.
18. Модель Entity Data Model. Класс DbContext. Методы DbContext для управления запросами: выборка, добавление, изменение, удаление.
19. Реализация SQL- запросов средствами Entity Framework. LINQ to Entities. Разработка контроллера для фильтрации данных.
20. Разработка контроллера для групповых запросов.
21. Технология MS WebAPI Core. Методы контроллеров VS WEBAPI. Возвращаемые типы данных.

22. Сериализация, десериализация. Пример WEBAPI – проекта для доступа к данным. Клиентские приложения баз данных, использующие службы REST.
23. Средства обращения к методам службы из клиентских приложений. Базовые классы для отправки HTTP-запросов и получения HTTP-ответов.
24. Публикация сервисов WebAPI на some.com.
25. Доступ с веб - сервисам в десктоп, мобильных и веб-приложениях.
26. Информационные системы с использованием сервисов REST API и SOAP. Одностраничные клиентские приложения. Клиент Blazor WebAssembly.
27. Доступ WEBAPI из Blazor WebAssembly. CORS для браузерных клиентов. Асинхронная модель программирования.
28. Примеры клиентских приложений, использующих службы REST для доступа к данным.
29. Веб - сервисы gRPC. Применение. Прото – файлы. Кодирование сообщений с помощью Protocol Buffers. Объекты реализации сервиса.
30. Подключение gRPC в классах проекта. Клиенты gRPC. Стабы.
31. Генерация прокси - сервера. Примеры проектов.
32. Веб - сервисы с протоколом доступа OData. Применение. Возможности.
33. Документ метаданных OData. Операции отбора. Клиенты OData. Установка объекта OData Connected Service.
34. Генерация прокси-службы в приложении клиента. Браузерный клиент Blazor Web Assembly для доступа к веб – сервису с помощью протокола OData.
35. Библиотеки для работы с OData. Перспективы OData. Примеры проектов.

Критерии оценивания теоретических вопросов

1. Нормы оценивания ответов на теоретические вопросы

№ п/п	Теоретический вопрос	Количество баллов (*)
1	Дан краткий ответ на поставленный вопрос	1 балл
2	Дан развернутый ответ на вопрос с анализом результатов	2 балла

(*) Возможна градация в 0,25 балла.

2. Шкала оценивания. Оценка «зачтено» за ответы на теоретические вопросы выставляется, если набрано не менее 3 баллов при ответе на три вопроса, в противном случае выставляется «не зачтено».

Задания для лабораторных занятий

Задачи по темам курса предложены к каждому лабораторному занятию.

Задания для лабораторных и самостоятельной работ, образцы решений основных типовых задач практики также размещены в системе дистанционного обучения СмолГУ (www.moodle.smolgu.ru).

Образец задания

1. Создайте WEBAPICore – сервис для передачи клиенту изображения.
2. Создайте WindowsForms - клиента для вызова GET- метода WEBAPI – сервиса для передачи изображения.
3. Создайте браузерный клиент Blazor Web Assembly для вызова методов веб - сервиса
4. Создайте WPF - клиента, работающего на платформе Xamarin, для вызова методов веб – сервиса.

Критерии оценивания выполнения лабораторных работ

1. Нормы оценивания каждой лабораторной работы:

№п/п	Структурная часть работы	Количество баллов (*)
1	Ответ на теоретические вопросы по теме лабораторной работы	1 балл

2	Демонстрация выполнения конкретного задания, предложенного для самостоятельного решения лабораторной работе	2 балла
---	-------------------------------------------------------------------------------------------------------------	---------

(*) с возможностью градации до 0,25 балла.

2. Шкала оценивания. Оценка «зачтено» за лабораторную работу выставляется, если набрано не менее 2 баллов, в противном случае за работу выставляется «не зачтено».

6.2. Оценочные средства и критерии оценивания для промежуточной аттестации

Зачетная работа (пример задания)

1. Разработайте клиентское приложение для доступа к своему сервису передачи файла JSON.
2. Установите средства сериализации и десериализации данных для взаимодействия с сервисом.

Критерии оценивания зачетной работы

1. Нормы оценивания работы

№ п/п	Структурная часть контрольной работы	Количество баллов (*)
1	Правильно реализован каждый метод решения	1 балл
2	Анализ результатов	2 балла

(*) Возможна градация в 0,25 балла.

2. Шкала оценивания работы:

п/п	Оценка	Количество баллов
1	Отлично	4,75-5
2	Хорошо	3,75-4,5
3	Удовлетворительно	3-3,5
4	Неудовлетворительно	менее 3

Критерий получения зачета

Зачет выставляется по результатам работы студента в течение семестра.

Для получения зачета студент должен:

- уметь отвечать на теоретические вопросы, рассмотренные на лекциях;
- уметь решать задачи, предложенные на лабораторных занятиях;
- уметь решать задачи, предложенные на зачетной контрольной работе.

7. Перечень основной и дополнительной учебной литературы

7.1. Основная литература

1. Полуэктова, Н. Р. Разработка веб-приложений : учебное пособие для вузов /— Москва : Издательство Юрайт, 2021. — 204 с. — (Высшее образование). — ISBN 978-5-534-13715-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/466449>
2. Сысолетин, Е. Г. Разработка интернет-приложений : учебное пособие для вузов /— Москва : Издательство Юрайт, 2022. — 90 с. — (Университеты России). — ISBN 978-5-9916-9975-4. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/415378>

7.2. Дополнительная литература

3. Руководство Microsoft по проектированию архитектуры приложений. Дата обновления: 01.01.2022. URL: <http://apparchguide.ms/Book>.

4. Ньюкомер, Эрик. Веб-сервисы. XML, WSDL, SOAP и UDDI: пер. с англ. / Э. Ньюкомер. – СПб. : Питер, 2020. – 256 с.

7.3.Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Система дистанционного обучения СмолГУ (moodle.smolgu.ru).
2. Интернет-Университет Информационных Технологий, <http://www.intuit.ru>
3. Каталог образовательных Internet-ресурсов, <http://window.edu.ru>.
4. Библиотека разработчика Microsoft, <http://msdn.microsoft.com>.

8. Материально-техническое обеспечение

Для занятий необходимы:

1. проектор;
2. интерактивная доска;
3. персональные компьютеры.

Для самостоятельной работы подготовлены аудитории № 224, 226, 230, 234 с выходом в Интернет, оснащенные компьютерами IBMPCс процессорами IntelCore 7 и оперативной памятью не менее 16 Гб.

9. Программное обеспечение

1. MicrosoftVisualStudio

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 03B6A3C600B7ADA9B742A1E041DE7D81B0
Владелец: Артеменков Михаил Николаевич
Действителен: с 04.10.2021 до 07.10.2022