**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

# по дисциплине « Разработка приложений баз данных для информационных систем»

на тему: «Обработка HTTP запросов средствами ASP.NET Core. Сохранение состояния. Кэширование.»

Вариант № 25

Выполнила: студентка

группы ИТП-31

Рожина А.К.

Проверил: доцент

Асенчик О. Д.

Гомель 2023

**Лабораторная работа № 3**

**Цель работы:** ознакомиться c методами обработкой HTTP средствами ASP.NET Core, методами сохранения состояния приложения и повышение производительности приложений путем использования разных видов кэширования.

**ЗАДАНИЕ**

1. Используя ранее разработанные объектную модель для доступа к данным в заданной предметной области разработать простое ASP.NET Core приложение.

2.1. С использование методов Run, Map и Use разработать:

1. компоненты промежуточного уровня (middleware) и встроить их в конвейер обработки HTTP запроса с целью кэширования 20 записей из каждой таблицы базы данных заданной предметной области с помощью встроенного инструмента кэширования - объекта IMemoryCache. Данные в кэше хранить неизменными в течение 2\*N+240 секунд, где N- номер вашего варианта.

2. собственную систему маршрутизации входящих запросов:

• если URL адрес входящего запроса содержит \info – выводить в выходной поток для отображения браузером информацию о клиенте и выходить из конвейера обработки запроса;

• если URL адрес входящего запроса содержит \table (где table – имя таблицы из базы данных) – выводить в выходной поток для отображения браузером с использование метода Response.WriteAsync кэшированную информацию из соответствующей таблицы базы данных и выходить из конвейера обработки запроса;

• если URL адрес входящего запроса содержит \searchform1 или \searchform2 – выводить в выходной поток для отображения браузером с использование метода Response.WriteAsync формы для поиска информации из базы данных и выходить из конвейера обработки запроса;

форма должна содержать, как минимум: одно поле, одного поле со списком, один список, одну кнопку;

• в противном случае (URL адрес входящего запроса не содержит перечисленных выше элементов) - продолжать обрабатывать другие компоненты конвейера обработки запросов и передавать управление системе маршрутизации MVC фреймворка;

2.2. Реализовать сохранение состояния элементов одной формы одной страницы с использованием куки (\searchform1).

2.3. Реализовать сохранение состояния элементов одной формы одной страницы в виде одного объекта специальной структуры с использованием объекта Session (\searchform2).

2.4. Осуществить заполнение элементов формы при их загрузке данными ранее сохранненными в объекте Session и куки (\searchform1, \searchform2 ).

2.5. С использованием средств разработчика браузера (Chrome, Firefox) продемонстрировать ускорение обработки запроса при наличии кэширования с использованием MemoryCache.

**Ход работы**

1. Результат кэширования таблицы *Apartments* представлен на рисунке 1.

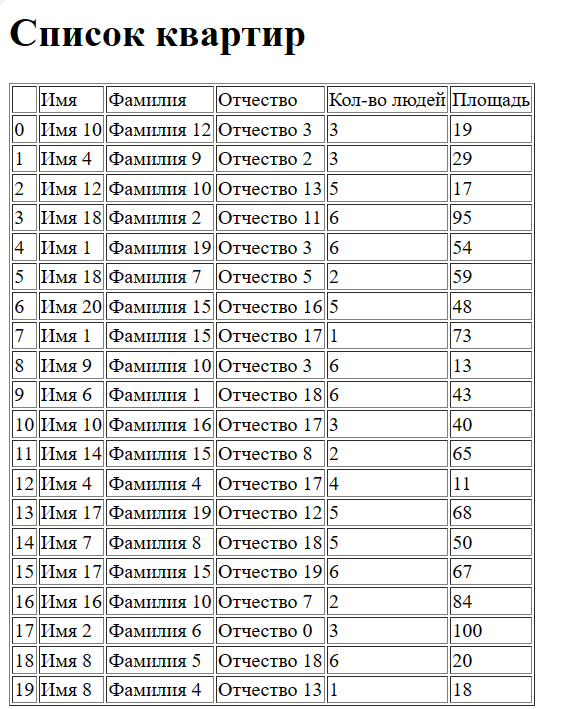


Рисунок 1 – Таблица *Apartments*

2. Отображение информации о клиенте представлено на рисунке 2.

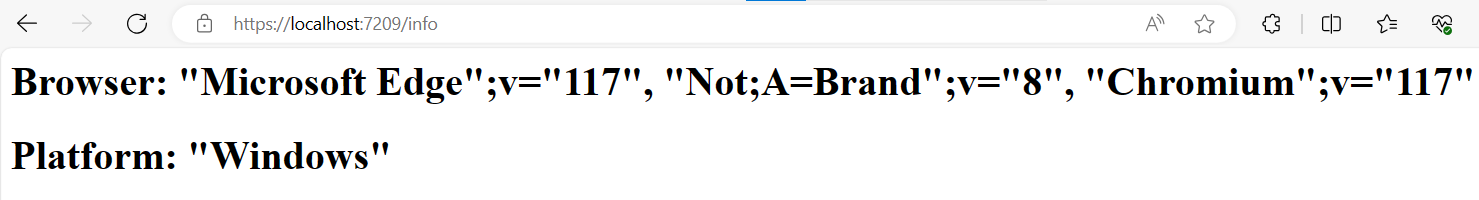


Рисунок 2 – Информация о клиенте

3 Отображение запроса, содержащего *URL* адрес с \cookiesSearchApartments и использованием *cookies,* представлено на рисунке 3.

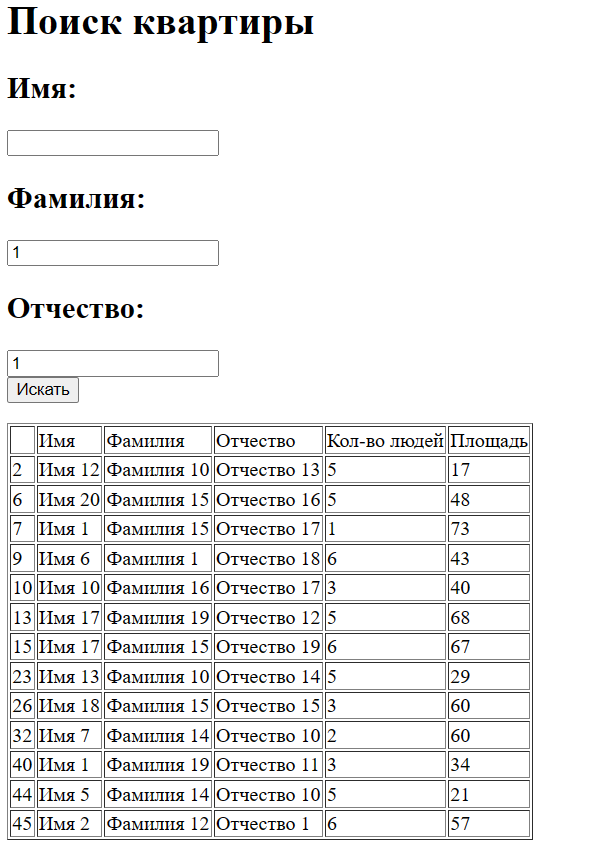


Рисунок 3 – Запрос, содержащий *URL* адрес с \cookiesSearchApartments

4. Отображение запроса, содержащего *URL* адрес с \sessionSearchApartments и использованием *Session,* представлено на рисунке 5

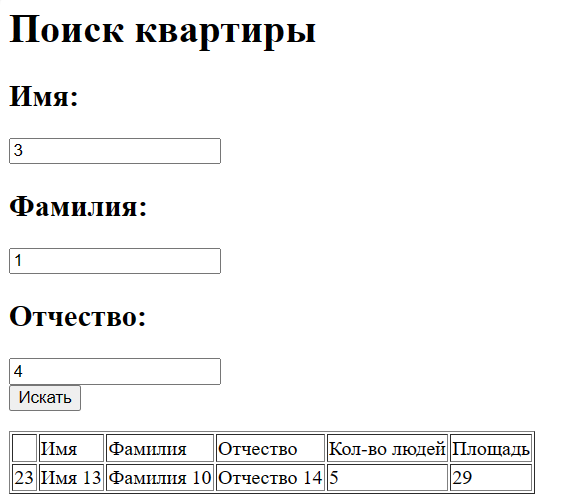


Рисунок 4 – Запрос, содержащий URL адрес с \sessionSearchDishes

5.1 Результат запроса по адресу \apartments при использовании кэширования показан на рисунке 4

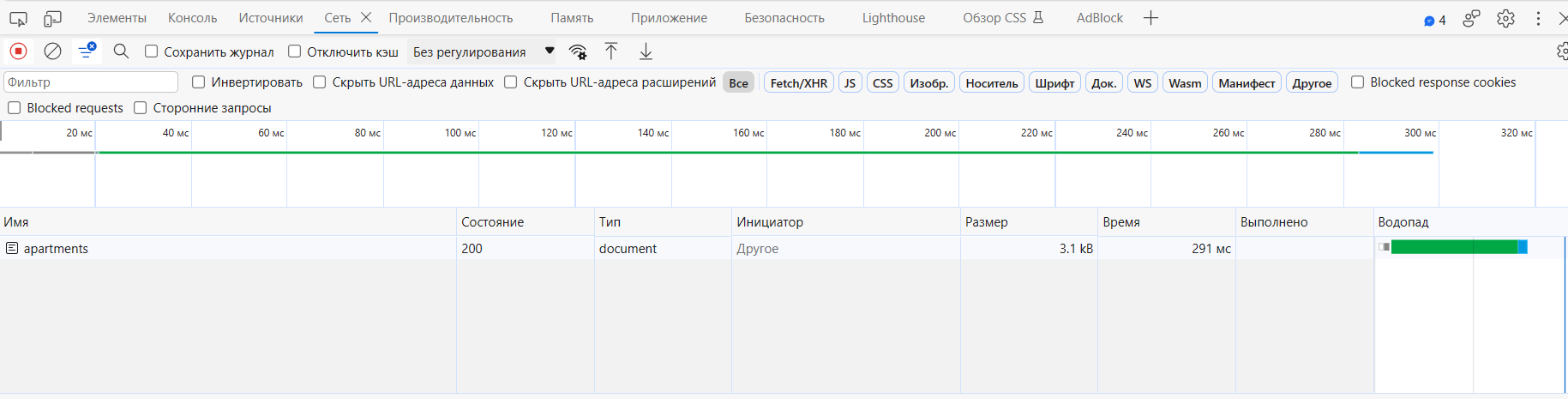
**

Рисунок 5 – запрос с использование кэширования

5.2 Результат запроса по адресу \dishes без использования кэширования показан на рисунке 6.

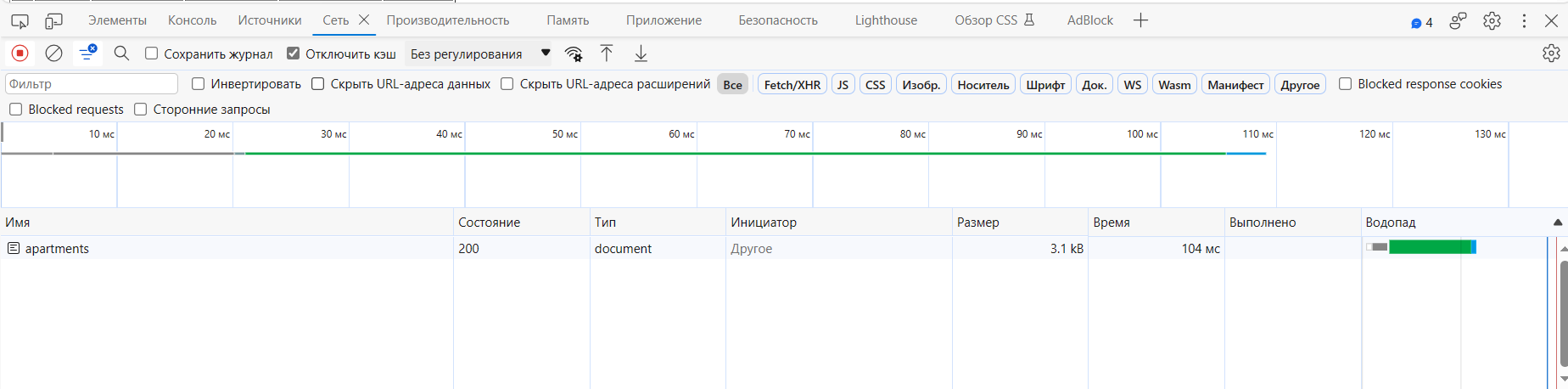


Рисунок 6 – запрос без использования кэширования

Ссылка на репозиторий проекта в *GitHub*:

**Вывод**: были ознакомлены c методами обработки HTTP средствами ASP.NET Core, методами сохранения состояния приложения и повышения производительности приложений путем использования разных видов кэширования.