# Теоретическая часть

* Понятие о фильтрах в ASP.NET Core MVC
* Порядок выполнения фильтров
* Создание синхронных и асинхронных фильтров действий
* Создание фильтров результатов
* Создание фильтров исключений
* Создание гибридного фильтра
* Пример простейшего сервиса
* Создание фильтров по типу при помощи атрибута **TypeFilter**
* Создание фильтров по сервису при помощи атрибута **ServiceFilter**
* Регистрация глобальных фильтров

# Практическая часть

Создать приложение ASP.NET Core MVC. Стилизацию приложения выполняйте при помощи фреймворка Bootstrap.

Примените страницу компоновки. Не забывайте создать модель для решения задачи. По GET-запросу **/Home/Index** выводите этот текст.

**Задача 1.** Туристическая фирма организует пешие маршруты по степным заказникам. Сведения о туристических маршрутах содержат: идентификатор маршрута, начальный пункт маршрута, обязательный промежуточный пункт маршрута, конечный пункт маршрута, протяженность маршрута в км (целое число), сложность маршрута (значение из ряда **А-, А, А+, В-, В, В+, С-, С, С+**; **А-** соответствует минимальной сложности, **С+** соответствует максимальной сложности), фамилия и инициалы инструктора маршрута (всего на фирме работает пять инструкторов, требуется редактировать сведения об инструкторах, удалять или добавлять инструктора не надо).

Сведения об инструкторах содержат: идентификатор инструктора, фамилия, имя, отчество, дата рождения, категория (**А, В, С**; А – низшая категория, С – высшая категория).

Данные о маршрутах и инструкторах хранить в файле **routes.json, instructors.json** в папке **App\_Data**. Реализуйте следующий функционал (критерии сортировки и выборки задавайте через переменные маршрута):

* Добавление маршрута (в форме на отдельной странице, требуется валидация на стороне сервера и на стороне клиента)
* Редактирование сведений о маршруте (в форме на отдельной странице, это может быть та же страница, что страница добавления маршрута, требуется валидация на стороне сервера и на стороне клиента)
* Удаление сведений о маршруте
* Вывод сведений о маршруте в порядке хранения в файле
* Вывод сведений о маршруте по убыванию протяженности маршрута
* Вывод сведений о маршруте по возрастанию протяженности маршрута
* Вывод сведений о маршруте по возрастанию сложности
* Вывод маршрутов с заданным начальным пунктом
* Вывод маршрутов, проходящих через заданный промежуточный пункт
* Вывод маршрутов с протяженностью, попадающей в заданный интервал, требуется валидация на стороне сервера и на стороне клиент
* Выводите сведения о маршруте, выбранном кнопкой (аналогично редактированию) в модальном «окне» на той же странице, что и сведения о маршрутах (используйте **AJAX**, возвращение **JSON** из метода действия контроллера)
* Редактирование сведений об инструкторе, требуется валидация на стороне сервера и на стороне клиента
* Вывод сведений об инструкторах в порядке хранения в файле
* Вывод сведений об инструкторах в порядке убывания категорий
* Вывод сведений об инструкторах в алфавитном порядке
* Вывод сведений об инструкторах с заданной категорией
* Вывод сведений об инструкторе, выбранном кнопкой (аналогично редактированию) в модальном «окне» на той же странице, что и сведения об инструкторах (используйте **AJAX**, возвращение **JSON** из метода действия контроллера)

**Задача 2.** В приложение по **задаче 1** добавьте возможность работы с коллекцией клиентов туристической фирмы. Сведения о клиенте содержат идентификатор, фамилию, имя, отчество, возраст клиента в годах, номер телефона, адрес электронной почты, пароль, признак постоянного клиента, фотографию клиента. При инициализации в коллекции клиентов должно быть не менее 12 записей.

Требуется хранить коллекцию сведений о клиентах в файле **clients.json** в папке **App\_Dat**a. Также надо добавлять, изменять и удалять сведения о клиенте, в том числе загружать фотографию клиента на сервер. При удалении сведений о клиенте его фотография также должна удаляться с сервера.

Для сведений о клиенте действуют следующие бизнес-правила:

* Все поля обязательны для ввода (кроме идентификатора, это поле генерируется в коде для обеспечения уникальности)
* Пароль должен быть от 8 до 28 символов в длину, хранить пароль в коллекции и в файле требуется в закодированном виде, для кодирования требуется применить операцию «Исключающее ИЛИ» каждого байта строки пароля и числа 42
* Возраст клиента должен быть от 18 до 65 лет

Для коллекции сведений о клиентах реализуйте следующий функционал (критерии сортировки и выборки задавайте через переменные маршрута):

* Вывод коллекции в порядке хранения в файле данных
* Вывод коллекции в алфавитном порядке
* Вывод коллекции по убыванию возраста
* Вывод сведений о клиентах с заданным диапазоном возраста (требуется валидация параметров на стороне клиента и на стороне сервера)
* Вывод сведений о клиентах по фамилии
* Вывод сведений о постоянных клиентах
* Вывод сведений о клиенте, выбранном кнопкой (аналогично редактированию) в модальном «окне» на той же странице, что и сведения о клиентах (используйте **AJAX**, возвращение **JSON** из метода действия контроллера)

*Разработайте сервис логирования, сохраняющий в файле* ***application.log*** *в папке* ***App\_Data*** *факты создания и редактирования сведений о клиентах: момент времени операции с точностью до секунды, собственно данные клиента (включая закодированный пароль).*

*Разработайте гибридный фильтр, фиксирующий при помощи сервиса логирования время работы методов действия и результатов. Задайте профилирование этим фильтром контроллер и результаты сведений об инструкторах.*

*Разработайте глобальный фильтр исключений – сервис с внедрением зависимостей, для проверки работы фильтра предусмотрите соответствующие действия в контроллерах маршрутов, инструкторов и клиентов. Параметры исключений записывайте в файл журнала при помощи сервиса логирования.*

# Дополнительно

Запись занятия можно скачать по [**этой ссылке**](https://cloud.mail.ru/public/nG5d/iMt1enrUb), материалы занятия в этом же архиве.