# Теоретическая часть

* Понятие о валидации данных, отправляемых на сервер
* Свойство **ModelState** контроллера для отслеживания состояния модели (валидно или не валидно)
* Непосредственная валидация состояния модели в коде метода действия
* Валидация на стороне сервера при помощи метаданных
* Валидация на стороне клиента при помощи ненавязчивого JavaScript – jQuery unobtrusive validation

# 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 пользователя. 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)

# Дополнительно

Запись занятия можно скачать по [**этой ссылке**](https://cloud.mail.ru/public/vk3c/GYVixkoQv), материалы занятия в этом же архиве.