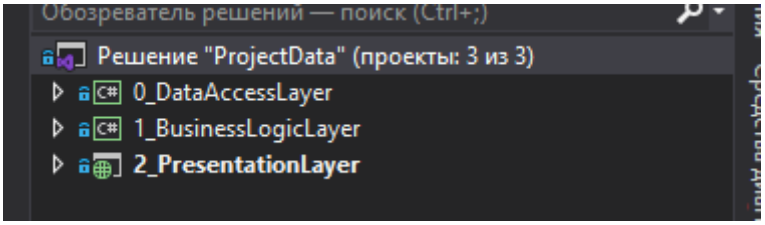
Соблюдение требований

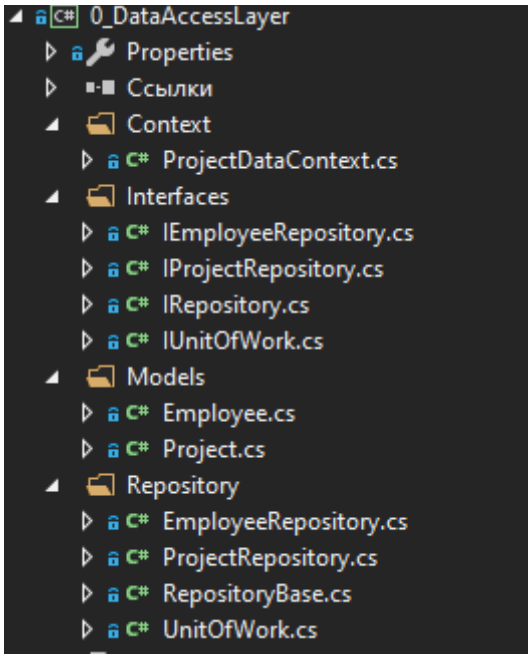
Архитектура проекта

Трёхуровневая архитектура проекта:

- уровень доступа к данным,
- уровень логики,
- уровень представления



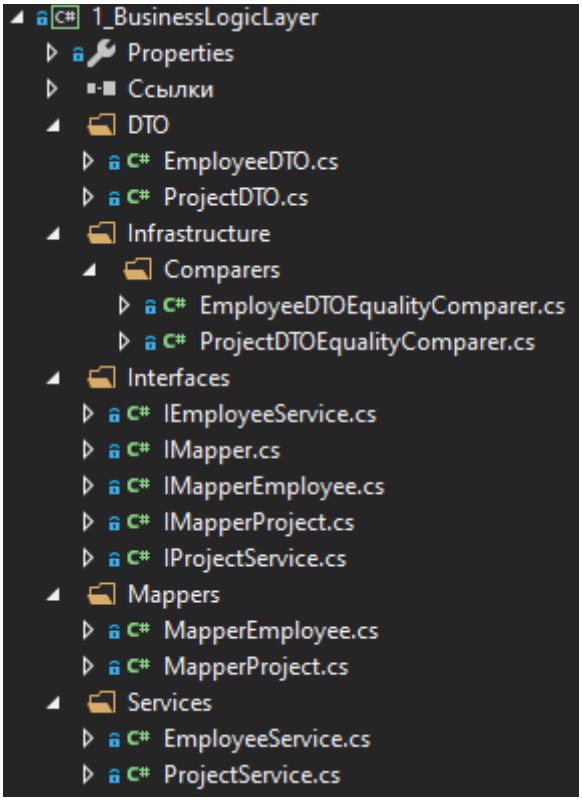
Уровень доступа к данным



На данном уровне определен контекст данных, описаны используемые модели, реализованны репозитории, а также UnitOfWork.

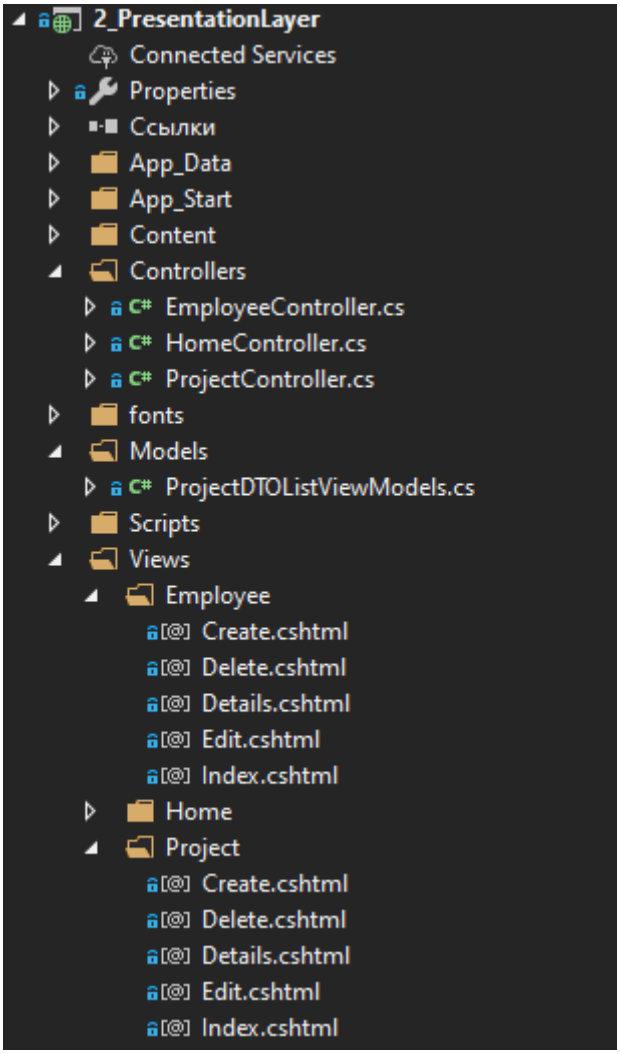
Используемые зависимости и NuGet паакеты: EntityFramework (v6.4.0)

Уровень логики



DTO для удобного представления модели, EqualityComparer для корректного сравнения, реализован интерфейс IMapper для маппинга из модели в DTO и наоборот, а также service (вся логика взаимодействия).

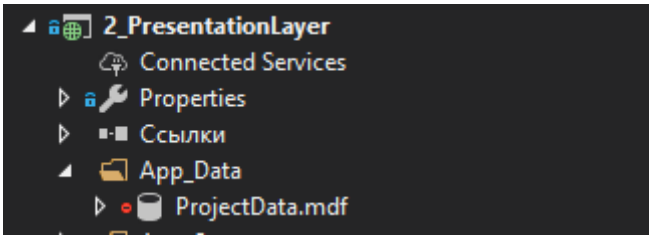
Уровень представления



Создано 2 контроллера: EmployeeController, ProjectController и их предситавления. Так же в папке Models есть ViewModels для проектов, т.к на них используются фильтры.

Используемые зависимости и NuGet паакеты: EntityFramework (v6.4.0) и стандартный набор, который устанавливается с MVC проектом.

Используемое хранилище



В папке App_Data создана локальная БД SQL server. Подключение к ней определено в Web.config:

<connectionStrings>

<add name="ProjectDataContext" connectionString="Data Source=(LocalDB)\MSSQLLocalDB; AttachDbFilename='|DataDirectory|\ProjectData.mdf';Integrated Security=True" providerName="System.Data.SqlClient" />

</connectionStrings>

Клиентский скрипт на JavaScript (файл Scripts/projectData.js)

```
$(document).ready(function () {
    // тултипы в таблицах
    let allTd = $('[data-toggle="tooltip"]');
    for (var i = 0; i < allTd.length; i++) {
        if (allTd[i].innerText.length > 10)
            // если явно указан title="ваш тултип" то ставим его
            allTd[i].title = allTd[i].title ? allTd[i].title : allTd[i].innerText;
    }
    allTd.tooltip();

    // uncheck radio кнопок
    $("[type='radio']").click(function (e) {
        if (e.ctrlKey) this.checked = false;
    })

    // обрабатываем сортировку таблиц
    let allTh = $('th');
    allTh.each(function (index) {
        // на пустых заголовках не работает
        if (allTh[index].innerText) {
            stateSort.push(-1);
            $(this).on("click", function () {
                sortTable(index, allTh);
            });
        }
    });
});
```

Он выполняет е функции: показывает тултипы в ячейках таблиц, где названия очень длинные, позволяет снимать руководителя проекта и сортирует таблицы.

```
/* сортировка таблиц */
var stateSort = [];// состояние сортировки для таблицы (-1 не сорт. 0 сорт. 1 инверсия от 0)
function sortTable(index, headers) {
    const table = $('table')[0];
    if (table) {
        let sortedRows = Array.from(table.rows).slice(1);
        if (stateSort[index] === -1) {
            sortedRows.sort((rowA, rowB) => rowA.cells[index].innerHTML > rowB.cells[index].innerHTML ? 1 : -1);
            stateSortChange(index, 0);
            headersChange(index, ' ↑', headers);
        }
        else if (stateSort[index] === 0) {
            sortedRows.reverse();
            stateSortChange(index, 1);
            headersChange(index, ' ↓', headers);
        }
        else if (stateSort[index] === 1) {
            sortedRows.reverse();
            stateSortChange(index, 0);
            headersChange(index, ' ↑', headers);
        }
        table.tBodies[0].append(...sortedRows);
    }
}

// изменение глобального состояния сортировки
function stateSortChange(changeIndex, newState) {
    for (var i in stateSort) {
        if (i == changeIndex) stateSort[i] = newState;
        else stateSort[i] = -1;
    }
}
// визуализация сортировки таблиц
function headersChange(currentIndex, mode, headers) {
    headers.each(function (i) {
        headers[i].innerText = headers[i].innerText.replace(' ↑', '').replace(' ↓', ''); //вырезаем символы
        if (i == currentIndex) headers[i].innerText = headers[i].innerText + mode;
    });
}
```

Демонстрация работы:

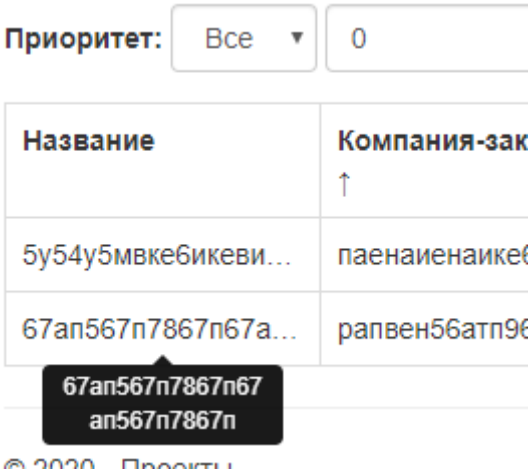


Рис. Тултипы.

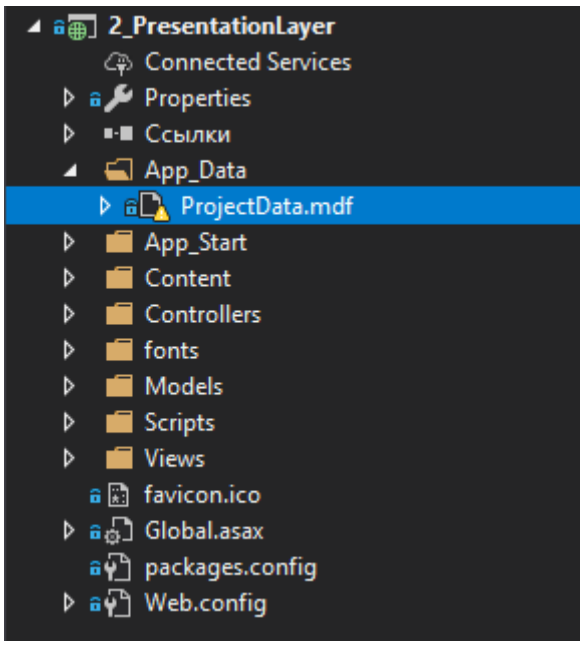
Название	Компания-заказчик	Компания-исполнитель ↑	Дата начала	Дата окончания	Приоритет	Сотрудники	Исполнители
5у54у5мвкеби...	паенаиенаике6ва	54вм5квке5ивкевм...	2020-04-06	2021-05-01	0	0	0
67ап567п7867п67а...	рапвен56атп96по7...	5ап6иаенаи56а56и...	2020-04-06	2021-04-06	0	0	0

Рис. Сортировка таблиц.

Конфигурация и сборка проекта:

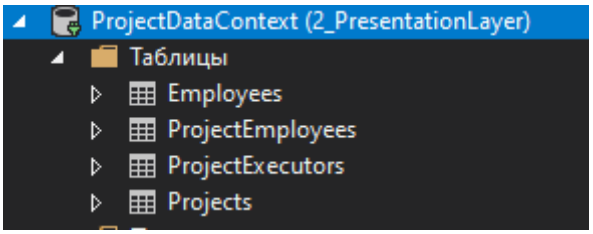
При скачивании проекта из https://github.com/AnatoliyCh/test-sibers-asp-mvc:

1. Восстановить локальную БД (удалить данный файл):



   Нужно создать SQL server с именем: ProjectData (правый клик - добавить - SQL server)

2. Выполнить в БД ProjectData sql скрипт (./Файлы/sql), который создаст все необходимые таблицы.



3. Одно из этих решений:
   a. Зайти в менеджер пакетов NuGet и восстановить все имеющиеся пакеты.
   b. Необходимо удалить и снова установить NuGet пакет(студия при запуске проекта хоть и подтягивает все зависимости, но почему-то папку roslyn не устанавливает): *Microsoft.CodeDom.Providers.DotNetCompilerPlatform*