

# Тестовое задание при приеме на работу

## Задание

1. Проанализировать данные из Приложение 1(Data.xml), исходя из описания предметной области;
2. создать таблицы\бизнес-объекты в 3-й нормальной форме;
3. загрузить данные из Приложение 1(Data.xml) в таблицы;
4. реализовать службу WebApi со следующими функционалом:
  - a) приём файлов с аналогичной структурой в Data.xml, с последующей обработкой и записью данных в БД;
  - b) выдача файлов отчета с фильтром по полю «**Номер поезда**», сформировать отчет по данным одного натурального листа в формате Excel, базируясь на шаблоне Приложение 2(NL\_Template.xlsx);
  - c) выдача структуры натурального листа в формате **Json** с фильтром по полю «**Номер поезда**»;
5. дополнительный функционал, необязательный к исполнению, но являющийся конкурентным преимуществом:
  - a. использовать DI (Dependency Injection) с IoC-контейнером (Inversion of Control) на выбор кандидата;
  - b. на службе WebApi реализовать авторизацию, на выбор кандидата Basic, JWT и т.д.

**Все обработчики, методы имеют пояснения по назначению.**

## Создание таблицы\бизнес-объекты в 3-й нормальной форме

Исходя из структуры файла Data.xml создана объектная модель:

Модель данных создавалась на основе Entity Framework Core (6.0). Все модели и DataContext в /Models

Для удобства сопровождения и восприятия ВСЕ свойства в объектной модели совпадают с *Приложение 1.*

*Описание полей Data.xml:*

**NatSheet** содержит поля заголовочной части «Натурного листа»

**Detail** содержит поля детализации касательно вагонов

При составлении объектной модели учитывалось, что номер поезда не может быть уникальными в пределах ВСЕХ записей БД. Поэтому записи с одним и тем же номером поезда возможны при условии, что у них разные параметр *Дата и время операции над составом*.

Исходя из этого в качестве первичного ключа использовался алгоритм: `yyymmdd000000`.

Где `yy` = Год – 2000, `mm` месяц, `dd` день, `000000` заполняются номером поезда. Например: 1906300**2204**, 2307300**2226**, 1906300**2226**. Здесь 2204 и 2226 - номера поезда, а 230730и 190630 это отражение временного периода создания записи.

Также учитывалось, что номер поезда может включать символ, поэтому номер поезда (*TrainNumber*) хранится в `char(5)`

*TrainNumber* т.к. это тип `char(5)`, то при запросе по этому полю через **web сервис** выбирается последняя запись с этим значением. (См. `Document_NatSheet.Create_Model_NatSheet`)

## Загрузка данных из Data.xml в таблицы

При загрузке файла \*.xml используется проверка схемы по DataSchema.xsd

Создан специальный API: `Api_XML`

- `Conv_XMLdata_intoModels` основной обработчик файлов \*.xml (используется при загрузке любого файла)
- `Conv_XMLdata_intoModels_parsXML` парсинг формата xml into model (`Document_NatSheet`)
- `Ls_Err_parsXML_NatSheet` (`List<Err_parsXML_NatSheet>`) буфер документов не прошедших проверку, а также фиксация ошибок при конвертации.

Т.образом, если после обработки буфер Ls\_Err\_parsXML\_NatSheet имеет значение NULL, значит обработка состоялась иначе эти ошибки передаются далее по конвейеру обработки.

Завершающий этап Document\_NatSheet.Save\_LS\_DocumentsAsync .

Document\_NatSheet сложная модель, нацеленная на разный функционал. Каждый метод имеет соответствующее пояснение.

Сервис загрузки на уровне браузера:

```
WebserController.UploadXML_async ([HttpPost("upload")])
DetailNT(Razor page) по ссылке на Upload (Razor page)
```

Если при загрузке возникли ошибки : «Повторный ввод» или «Ошибка конвертации данных» буфер ошибок Ls\_Err\_parsXML\_NatSheet выводится в информационный <div>...</div>

Загрузить натурный лист в формате xml

DataNext.xml Загружен в БД

Выбрать файлы DataNext.xml

Загрузить файл

Back

Выполнена загрузка.

Загрузить натурный лист в формате xml

Error: null 2168 null Error converting to DateTime: 30.06.201914:49:00  
Error: 19063002236 2236 2019-06-30T14:07:00 Повторный ввод документа  
Error: 19063002434 2434 2019-06-30T14:50:00 Повторный ввод документа

Выбрать файлы DataNext\_base.xml

Загрузить файл

Back

```
Список не обработанных натурных документов
File xml: J:\VSprojects\Testwork_from_employers\SoftMaster\WebApi\WebSoftMast_02\
19063002236 2168 30.06.2019 14:07:00 Повторный ввод документа
19063002434 2434 30.06.2019 14:50:00 Повторный ввод документа
```

Загрузка не прошла проверку. Отображение сообщения в браузере и в файловой системе в журнале list\_write\_Doc\_existsNatSheet\_2023-4-5.log.

Это отображение содержимого буфера Ls\_Err\_parsXML\_NatSheet .

Выдача файлов отчета с фильтром по полю «Номер поезда», сформировать отчет по данным одного натурного листа в формате Excel, базируясь на шаблоне Приложение 2(NL\_Template.xlsx)

Формат файла отчета не указан, поэтому на основе данных БД для браузера создана страница ListNT (Razor page) отображающая отчет в виде:

Список натурных документов				
TrainNumber	Index	DateOper	Station	
2012	196	30.06.2019 14:07:00	МЫС АСТАФЬЕВА (ЭКСП.)	Подробнее
2040	722	30.06.2019 14:50:00	НАХОДКА (ЭКСП.)	Подробнее
2874	331	30.06.2019 14:19:00	МЫС АСТАФЬЕВА (ЭКСП.)	Подробнее
2974	49	30.06.2019 14:06:00	НАХОДКА (ЭКСП.)	Подробнее

Выгрузить список в файл Excel

Загрузить XML



**NatSheet\_excel** наследованный класс для создания файла «Натурный лист» по заданному номеру поезда.

НАТУРНЫЙ ЛИСТ ПОЕЗДА						
Поезд №:		2226	Станция:		МЫС АСТАФЬЕВА (ЭКСП.)	
Состав №:		177	Дата :		43646.60764	
№	№ вагона	Накладная	Дата отправления	Груз	Вес по документам (т)	Последняя операция
1	63558985	ЭЛ403699	30-06-2019	УГОЛЬ КАМЕННЫЙ МАРКИ Д	74.900	ОТПРАВЛЕНИЕ ВАГОНА В СОСТАВЕ ПОЕЗДА СО СТАНЦИИ
2	60825163	ЭЛ403767	30-06-2019	УГОЛЬ КАМЕННЫЙ МАРКИ Т-ТОЩИЙ	70.000	ОТПРАВЛЕНИЕ ВАГОНА В СОСТАВЕ ПОЕЗДА СО СТАНЦИИ
46	62551874	ЭЛ314035	30-06-2019	УГОЛЬ КАМЕННЫЙ МАРКИ Т-ТОЩИЙ	74.950	ОТПРАВЛЕНИЕ ВАГОНА В СОСТАВЕ ПОЕЗДА СО СТАНЦИИ
47	62551874	ЭЛ314035	30-06-2019	УГОЛЬ КАМЕННЫЙ МАРКИ Д	74.950	ОТПРАВЛЕНИЕ ВАГОНА В СОСТАВЕ ПОЕЗДА СО СТАНЦИИ
	2			УГОЛЬ КАМЕННЫЙ МАРКИ Д	149.900	
	44			УГОЛЬ КАМЕННЫЙ МАРКИ Т-ТОЩИЙ	3 256.300	
	1			АНТРАЦИТ	74.950	
Всего: 47				3	3 481.150	

Список натуральных документов				
№	TrainNumber	Index	DateOper	Station
1	2012	196	30-06-2019	МЫС АСТАФЬЕВА (ЭКСП.)
2	2040	722	30-06-2019	НАХОДКА (ЭКСП.)
3	2082	017	30-06-2019	НАХОДКА (ЭКСП.)
4	2114	462	30-06-2019	НАХОДКА-ПЕРЕВАЛКА

**Реализовать авторизацию**

Аутентификация и авторизация создана на основе API ASP.NET Core Identity

Созданы инструменты управления «Ролями», «Пользователями», а также регистрация пользователя.

Т.к. для web служб не используется переход на страницу «Авторизации», поэтому требуется реализация другого сценария (в частности на основе Cookie набора):

Переход по адресу <http://localhost:5215/webclient.html>

**Данные ауторизации**

Логин

Пароль

- Get Data
- Log In
- Log Out

Error: 401: Unauthorized

Здесь клик на кнопке «**Get Data**» для не авторизованного пользователя

## Данные авторизации

Логин

Пароль

Get Data

Log In

Log Out

upload data  
ok

## Данные авторизации

Логин

Пароль

Get Data

Log In

Log Out

admin, admin@example.com  
alica, alica@example.com  
bob, bob@example.com  
joe, joe@example.com

Здесь клик на кнопке «**Get Data**» после авторизации. Вывод данных пользователей

Get Data

Log In

Log Out

Logged out

Здесь клик на кнопке «**Log Out**»

### Реализация для администратора:

По адресу в браузере: <http://localhost:5215/users/list>

Выполняется переход на авторизацию:

Log In

UserName

admin

Password

.....

Log In

После чего:

User Administration			
ID	Name	Email	
14185a09-1f8b-4e80-bf04-2d07e787ff21	admin	admin@example.com	<a href="#">Edit</a> <a href="#">Delete</a>
9a749975-1f3f-4113-8bc8-04f83c3b33e3	alica	alica@example.com	<a href="#">Edit</a> <a href="#">Delete</a>
f154bdf4-6dfc-4f11-b870-0cde249244ea	bob	bob@example.com	<a href="#">Edit</a> <a href="#">Delete</a>
f2f0956e-2d2f-42e3-961b-a32a374ad159	joe	joe@example.com	<a href="#">Edit</a> <a href="#">Delete</a>

[Create](#)

Переход по <http://localhost:5215/roles/list>

Role Administration			
ID	Name	Members	
ab9695b8-04e9-4ca5-87cf-bbe84dfddcc1	manager	bob, joe	<a href="#">Edit</a> <a href="#">Delete</a>
bceb7589-b2ee-4e62-a1e9-26d3e66baaf4	Admins	admin	<a href="#">Edit</a> <a href="#">Delete</a>
fee9952d-3845-4a66-9350-9a10884a350e	developer	admin, alica, bob	<a href="#">Edit</a> <a href="#">Delete</a>

[Create](#)

Клик по кнопке «Edit» переход на страницу назначения роли «manager»

Edit Role: manager		
Members		
User	Email	
bob	bob@example.com	<a href="#">Change</a>
joe	joe@example.com	<a href="#">Change</a>
Non-Members		
User	Email	
admin	admin@example.com	<a href="#">Change</a>
alica	alica@example.com	<a href="#">Change</a>

[Back](#)

Управление процессом авторизации для Web служб конфигурируется в program:

```
builder.Services.AddAuthentication(opts =>
{
    opts.DefaultScheme =
        CookieAuthenticationDefaults.AuthenticationScheme;
    opts.DefaultChallengeScheme =
        CookieAuthenticationDefaults.AuthenticationScheme;
}).AddCookie(opts =>
{
    opts.Events.DisableRedirectForPath(e => e.OnRedirectToLogin,
        "/api", StatusCodes.Status401Unauthorized);
    opts.Events.DisableRedirectForPath(e => e.OnRedirectToAccessDenied,
        "/api", StatusCodes.Status403Forbidden);

});
```

Обработчик **CookieAuthenticationEvents** переконфигурирован в классе **CookieAuthenticationExtensions**

## Модульное тестирование xUnit

Тестирование	Длитель...
WebSoftMaster_02.Test (20)	11.9
WebSoftMaster_02.Test.API_XML_op...	2.7
Api_XML_loadData_test (2)	2.6
Api_XML_Conv_XMLdata_intoM...	2.5
Api_XML_Conv_XMLdata_intoM...	103 м
Api_XML_test (1)	111 м
Api_XML_Conv_XMLdata_intoM...	111 м
WebSoftMaster_02.Test.Excel_data (2)	3.4
CreateExcel_NatSheetDocument_t...	3.4
CreateExcel_List_NatSheetDocu...	63 м
CreateExcel_NatSheetDocument_...	3.3
WebSoftMaster_02.Test.FileOperations	< 1 м
Api_files_test (2)	< 1 м
Api_files_Path_downloads	< 1 м
Api_files_Path_uploads	< 1 м
WebSoftMaster_02.Test.ModelsApi (...)	2.9
Api_models_test (5)	2.7
Try_getMaxID_detail	2.7
Try_getMaxID_empty	< 1 м
Try_getMaxID_natSheet	1 м
Try_getMaxID_tmp_detail	4 м
Try_getMaxID_tmp_natSheet	20 м
Document_NatSheetGrData_test (...)	170 м
NatSheet_CreateKeyID_test (4)	21 м
WebSoftMaster_02.Test.NatSheetDo...	2.9
NatSheetDoc_testing (2)	2.9
Document_NatSheet_Create_Doc...	2.8
Document_NatSheet_Create_Doc...	110 м
NatSheets_existsData_test (1)	5 м
Document_NatSheet_existsData	5 м

Модульное тестирование (xUnit) затронуло практически весь созданный код.

Для удобства тестирования используется вывод ошибок в файл журнала (при тестировании загрузки \*.xml)

## Deployment

Заполнение БД происходит автоматически при первом старте см. Program:

**await SeedData.Load\_SeedDataAsync()**

```
if (context.NatSheets.AsNoTracking().Any()) return;
```

продолжение, если в БД нет данных

Заполнение начальными данными для роли и пользователя «admin»

**IdentitySeedData.CreateAdminAccount(app.Services, app.Configuration);**

**DB.zip** – архив файлов БД

**DataNext.xml** – используемый файл для загрузки \*.xml документа.