**Если требуется добавить новую сущность**

1. Создать класс сущности в Kokos\_Models

2. Добавить контекст в ApplicationDbContext

3. Создать миграцию add-migration

4. Обновить БД update-database

3. Добавить соответствующий интерфейс в IRepository

4. Добавить соответствующую реализацию Repository

5. Зарегистрировать службы в Startup.cs

**Добавление стороннего сервиса:**

1. Добавить все параметры в appsettings.json

2. В Kokos\_Utility добавить классы (при необходимости и интерфейсы) для работы с сервисом(пример Braintree)

3. Не забыть Зарегистрировать службы в Startup.cs, если используется внедрение зависимостей

services.Configure<BrainTreeSettings>(Configuration.GetSection("BrainTree"));

В этом случае при разрешении зависимостей поля объекта типа BrainTreeSettings будут автоматически заполняться значениями из appsettings.json

public BrainTreeGate(IOptions<BrainTreeSettings> options)

{

Options = options.Value;

}

Можно извлекать данные и внутри класса (как это сделано для сервиса MailJet)

private readonly IConfiguration \_configuration; // получение конфигурации через DI

…

MailJetSettings = \_configuration.GetSection("MailJet").Get<MailJetSettings>();

Первый вариант более правильный, т.к. вся регистрация настроек в одном месте

**Добавление стороннего сервиса аутентификации:**

1. Добавить компонент через Nuget

2. В Startup.cs добавить сервис.

**В проекте используется:**

1. Awesome – стилизованный шрифт

2. Mail Jet – сервис отправки Email

3. bootstrap

4. Toastr – стилизованные сообщения

5. Summernote – Расширенный редактор текста

6. Syncfusion

Ссылки на документацию по подключению:

https://ej2.syncfusion.com/aspnetcore/documentation/pdfviewer/getting-started/

https://ej2.syncfusion.com/aspnetcore/documentation/grid/getting-started-core/

https://ej2.syncfusion.com/aspnetcore/Grid/GridOverview#/material

7. DataTables

8. JQuery

9. Sweet alert – стилизованные сообщения