

Readme

Добрый день тут я объясняю, что это такое и с чем его едят :-D

Рабочая копия проекта находится по адресу - <http://worknest.ageron.info>

Как вы заметили это не простая программа, а решение в чем-то схожее на CMS? так вот вам не показалось :)

Говорю сразу, писал все я но для другого боевого проекта, для вашего проекта было изменено и дописано процентов 40%, в качестве основного подхода используется "инверсия зависимостей (IoC)" + библиотека для IoC от MS - Unity

А теперь пройдемся более подробно по системе

Все делалось в окружении "vasiliy.stan" это наследник от окружения "Debug" но персонализированное :)

- 1) проект поддерживает мультиязычность, т. е. все строки интерфейса так или иначе используют механизм библиотеки Translations
- 2) проекты Sources\Dal\Dal и Sources\Dal\Logs - библиотеки на основе EF6 и подхода CodeFirst для доступа к данным и логирования, через библиотеку log4net, соответственно :)
- 3) проект DataGenerator - проект для развертывания баз данных и инициализации их первичными значениями, сделано это потому, что постоянно переключаться на сервак и делать bak копии 2-3 баз меня задолбало + ведь не потащишь develop базы с ошметками тестов на production :) заказчик не поймет :) а тут 1 раз настроил корректно строки соединения и параметры для sql скриптов и забыл :), также она служит не просто развертыванию баз, а выполняет роль генератора первичных данных
- 4) модуль Sources\Frontend.Web.Core\Modules\Modules.WebApi, основной модуль WebApi системы асинхронен, имеет механизмы авторизации и работает на протоколе реализованном через json post запросы
- 5) Sources\Frontend.Web.Core\Libraries\Libraries.Core.Backend - основная бекенд библиотека для всей системы хранит в себе классы с базовой функциональностью
- 6) Sources\Frontend.Web.Core\Libraries\Libraries.Core.Frontend - основная фронтэнд библиотека по логике хранит в себе базовые модули на CSS и JQuery которые используются на фронтэнде
- 7) Sources\Frontend.Web.Core\Themes\Themes.Core - библиотека базовой темы сайта на основе Bootstrap
- 8) Sources\Frontend.Web.Core\Modules\Modules.Core - базовый модуль который хранит повторяемые страницы web сайта на подобии страницы входа, чтобы не верстать их заново в каждом новом приложении
- 9) Sources\Frontend.Web.Core\Modules\Modules.Layouts - базовый модуль который хранит повторяемые мастер страницы для SPA
- 10) Sources\Frontend.Web.Core\Modules\Modules.Site - основной модуль который хранит основные SPA для веб сайта
- 11) Sources\Frontend.Web.Core\Modules\Modules.WebApi.Shared - библиотека которая расшаривает классы запросов и ответов на основе которых и построен протокол WebApi текущего приложения

12) Sources\Frontend.Web - основное приложение построенное на ASP .NET MVC4 которое динамически связывает все остальные части

13) Sources\Translations - библиотека мультязычных переводов для сайта основанная на стандартном подходе с использованием *.resx файлов от MS

Для сборки проекта вам потребуются

- 1) Win10
- 2) MS VS 2017
- 3) MS SQL Server 2016
- 4) установленное расширение для MS VS 2017 - RazorGenerator
- 5) установленное расширение для MS VS 2017 - ResXManager

собирать нужно сразу полностью решение а не отдельные проекты

по возможности все "магические" элементы, как то строки или числа, были вынесены в файлы настроек

P. S. если у вас будут вопросы по проекту или просто пообщаться, то со мной можно связаться по email: vasiliyStankevich@gmail.com