Краткое описание проекта по локализации приложений.

Содержание

[1. Проект локализации приложений 2](#_Toc530558810)

[1.1. Общие положения по проекту 2](#_Toc530558811)

[1.2. Структура проекта 2](#_Toc530558812)

[1.3. Проверка работы проекта 2](#_Toc530558813)

[1.4. Описание проекта на Agular 2](#_Toc530558814)

[1.4.1. IDE 2](#_Toc530558815)

[1.4.2. Описание структуры папок 3](#_Toc530558816)

[1.4.3. Навигация и ленивая загрузка 3](#_Toc530558817)

[1.4.4. Полезные скрипты запуска 4](#_Toc530558818)

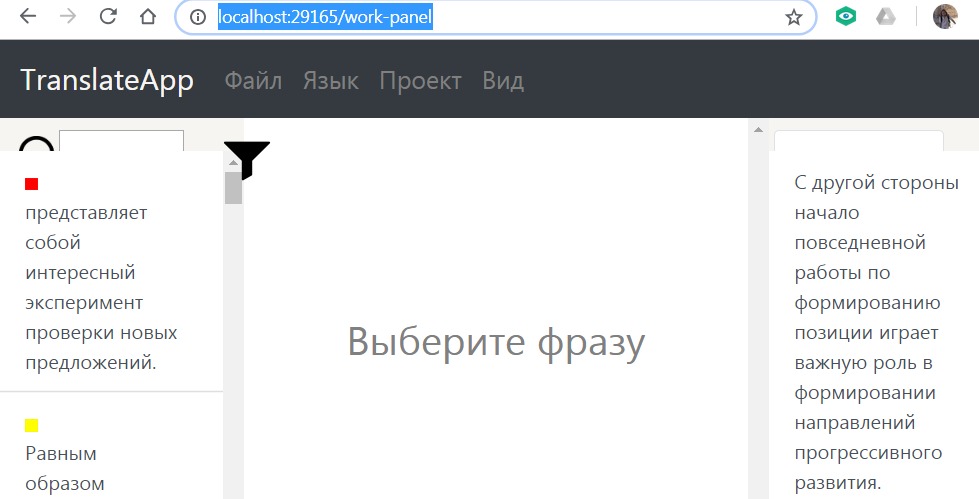
[1.5. Библиотеки Visual Studio Code 4](#_Toc530558819)

[1.6. Полезные ссылки 4](#_Toc530558820)

1. Проект локализации приложений
   1. Общие положения по проекту
2. Название проектов, объектов и так далее обсуждается.
3. Расположение объектов тоже обсуждается.
   1. Структура проекта
4. Localization – основное приложение.
5. Всю бизнес логику C# выносить в отдельный проект Models. В последствии будем писать мобильное приложение, которое будет использовать туже бизнес логику.
6. Утилиты работы с чем-то лучше выносить в Utilities, возможно оптом их будем использовать и для других приложений.
7. Скрипты работы с базой данных выносить в DAL.

* Предлогаю использовать Npgsql EF Core provider — Npgsql.EntityFrameworkCore.PostgreSQL (обсуждаеться).
* нельзя использовать Entity Framework. Пишут что у Entity Framework проблемы с быстродействием на больших проектов.
* ODBC – нельзя использовать, т.к. как заказчик потом хочет все на linux серверах использовать.
  1. Проверка работы проекта

1. Нужно откомлилять в VS или запустить ангуляр приложение и открыть страницу http://localhost:8880/work-panel (порт может быть любой). Результат на картинке ниже.



1. Сейчас работа над базовой папкой проекта
   1. Описание проекта на Agular
      1. IDE
2. Для работы с Angular можно использовать Visual Studio Code. Проект для открытия .\Localization\client.code-workspace.
   * 1. Описание структуры папок
3. Исходники находятся в папке . Localization\ClientApp\src\app
4. В проекте используется ленивая загрузка, то есть страницы загружаются только при обращении (как сделать будет описано ниже).
5. \Localization\ClientApp\src\app\app.module.ts – базовый модуль, туда подключать нельзя. Он подключает базовый модулю CoreModule .
6. CoreModule – главный модуль в котором подключаются все остальные модули.
7. Вспомогательные модули, библиотеки, тестовые далее и так далее лежат в в подпаках \Localization\ClientApp\src\app\: admin, entities, models, services, work-panel. В эту подпапку добавляйте свои разработки.
8. В core.module.ts – подключаете только необходимые вам import.
9. В core-routing.model.ts содержится объявление Routes – в рамках всей программы. Если вам нужно сделать навигацию в рамках свой странице можно посмотреть на admin-routing.module.ts.
10. Навигация и ленивая загрузка описана в разделе 1.1.3.
    * 1. Навигация и ленивая загрузка
11. Ленивая загрузка позволяет не загружать все js приложение за раз, а загружать только если пользователь просить открыть страницу. Для ее реализации необходимо разбить приложение на модули, которые и будут подгружаться при необходимости.
12. Для использования ленивой загрузки необходимо использовать ключевое слово loadChildren.
13. В core-routing.model.ts содержится объявление Routes – в рамках всей программы. Пример:

const routes: Routes = [

{

path: 'work-panel',

loadChildren: '../work-panel/work-panel.module#WorkPanelModule'

},

{

path: 'login',

component: LoginComponent

},

{

path: 'admin',

loadChildren: '../admin/admin.model#AdminModule'

},

{

path: '\*\*',

component: NotFoundComponent

}

];

1. Строка path: 'work-panel' говорить системе по какому url страница будет доступна. Пример url <http://localhost:8880/work-panel>
2. Строка '../work-panel/work-panel.module#WorkPanelModule' – показывать место расположения модуля.
3. Ключевое слово loadChildren говорит системе, что не нужно загружать сразу данный модуль.
   * 1. Полезные скрипты запуска

* \Localization\ClientApp\\_\_build\_prod.bat – собирает js для отправки на сервер приложений
* \Localization\ClientApp\0\_install\_node\_modules.bat – устанавливает библиотеки из node\_modules
* \Localization\ClientApp\1\_NPM setup.bat – обновляет библиотеки npm и node\_modules.
* \Localization\ClientApp\10\_start.bat – запускает приложение ангуляр.
* \Localization\ClientApp\4\_node\_version.bat – для проверки вырсии node и npm
* \Localization\ClientApp\7\_open\_url.bat – открывает урл <http://localhost:4200> в браузере
  1. Библиотеки Visual Studio Code

1. Библиотеки, которые я использую в Visual Studio Code (не уверен, что они вам все нужны)
   1. Полезные ссылки
2. Введение в модули Angular — корневой модуль (Root Module)

https://habr.com/post/351504/

1. Пример создания приложения SINGLE PAGE APPLICATION : ASP.NET MVC .NET CORE + ANGULAR

https://www.siv-blog.com/single-page-application-asp-net-mvc-net-core-angular-4-part-1/ - немного устарело

1. Angular Elements - How to use in ASP.NET Core using Angular CLI?

https://www.mithunvp.com/using-angular-elements-asp-net-core-angular-cli-visual-studio/

1. https://docs.microsoft.com/ru-ru/aspnet/core/client-side/spa-services?view=aspnetcore-2.1 Использование JavaScriptServices для создания одностраничных приложений ASP.NET Core (не знаю как переключить на оригинальный текст(английский)).
2. Ленивая загрузка: разделение кода NgModules с Webpack https://webformyself.com/lenivaya-zagruzka-razdelenie-koda-ngmodules-s-webpack/

