Руководство по эксплуатации подсистемы импорта данных в LCR

Подсистема выполнена в виде SPA-приложения клиенткой части и RPC Web API со стороны сервера. Разработка велась в IDE MS Visual Studio 2019.

В корне репозитория расположены три sln-файла: для всех проектов, для back-end, для front-end.

# Back-end API

Все необходимые данные клиентской стороне отдаются через вызовы методов контроллеров.

## Стек технологий

Asp.net Core 2.2 (Web API, Hosted service)

## Используемые библиотеки

* Entity Framework Core – доступ к данным;
* NLog – логирование;
* Automapper – преобразование данных;
* ExcelDataReader – работа с Excel файлами;
* FluentValidation – валидация данных.

## Конфигурация

Все настройки серверной части находятся в файлах appsettings\*.json.

* appsettings.json – настройки не зависящие от окружения (PROD/TEST);
* appsettings.Staging.json – настройки для тестовой среды;
* appsettings.Production.json – настройки для «продакшн» среды.

Настройки логирования расположены в nlog\*.json по тому же принципу.

## Сборка и публикация

Для сборки и запуска проекта, необходимо установить .Net Core SDK 2.2.

Собирать версию, готовую к публикации можно двумя способами. Либо с помощью пункта меню Publish из MS Visual Studio, либо из консоли с помощью команды (в папке с проекта LCR.Import.Web.Api):



где CONFIG\_NAME – одно из значений Debug, Testing, Release

OUTPUT\_PATH – путь к директории, куда сохранить сборку

# Front-end SPA

Одностраничное JavaScript приложение, получающее данные по HTTP.

## Стек технологий

* Asp.net Core 2.2 MVC
* HTML/TypeScript/SCSS
* NodeJS

## Используемые библиотеки

* Aurelia – JavaScript фреймворк;
* TypeScript – «типизированная версия» JavaScript;
* SCSS – CSS препроцессор;
* Pikaday – календарь для выбора даты;
* UiKit – CSS фреймворк (стили, контролы);
* Webpack – сборщик JavaScript приложений.

## Конфигурация

Все настройки клиентской части находятся в файлах app-config\*.ts

* app-config.ts – настройки для dev-окружения;
* app-config-test.ts – настройки для тестовой среды;
* app-config-prod.ts – настройки для «продакшн» среды.

## Сборка и публикация

Для сборки и запуска проекта, необходимо установить:

* .Net Core SDK 2.2;
* NodeJS 12.17.0;
* Если потребуется отлаживать проект в Visual Studio, рекомендуется установить расширение “NPM task runner”.

Собирать версию готовую к публикации можно также с помощью меню Publish из MS Visual Studio, проект настроен таким образом, что при этом автоматически устанавливаются необходимые NPM-пакеты, запускается сборка клиентского кода с помощью webpack и потом это всё с помощью dotnet кладется в одну директорию.

Сборка проекта из консоли осуществляется с помощью команды:



где CONFIG\_NAME – одно из значений Debug, Testing, Release

OUTPUT\_PATH – путь к директории, куда сохранить сборку

Работать с проектом можно из любой среды разработки, для запуска проекта локально, нужно выполнить следующие команды в папке проекта LCR.Import.UI.Web:



## Структура БД (не финальная версия)

К сожалению, в наличии есть только скрипт предварительной версии БД.

