Техническое задание на разработку парсера

Требуется создать парсер доски объявлений по разделу «Недвижимость», который должен выбрать все последние объявления, начиная с определенного объявления. Парсер должен быть написан на C# и выполнен в видеDLL библиотеки. Парсер получает в качестве входящих данных рубрику, регион, действие(продают, покупают и т. д.) и идентификатор объявления. Возвращаемый результат должен содержать все последние объявления размещенные на сайте после объявления, идентификатор которого передан входящим параметром. Если идентификатор последнего входящего объявления пустой, то возвращаемый результат - самое последнее объявление и его идентификатор.

Программа парсер – это **ОДНА** ваша библиотека, **ОДНА** ваша бд sqlite + **СОГЛАСОВАННЫЕ** сторонние библиотеки и бд.

Используемые бд:

- БД справочников регионов, рубрик, действий в формате Sqlite
- Справочники редактированию и изменению в одностороннем порядке без согласования не подлежат.
- Все возвращаемые библиотекой рубрики, регионы и действия должны соответствовать структуре справочников. Если структура каталогов ресурса отличается, то она должна быть приведена к структуре справочников.

Используемые технологии, библиотеки:

- VS 2012 + ReSharper
- .NET 4.5 (использование async/await)
- Managed Extensibility Framework (http://msdn.microsoft.com/ru-ru/library/dd460648(v=vs.110).aspx)
- HtmlAgilityPack – пакет из Nuget для парсинга Html
- servicestack.ormlite.sqlite32– пакет из Nuget для работы с Sqlite БД
- RT.Crawler - библиотека для получения контента с веба (вам предоставится исходный код)
- RT.ParsingLibs – базовые классы для парсинга, абстрактный модуль парсера (вам предоставится библиотека)

Требования к библиотеке парсера:

- Должна строиться по аналогии с примерами бибилотек RT.ParsingLibs.AbstractFirst и RT.ParsingLibs.AbstractSecond
- Парсер должен реализовать интерфейс IParsingModule из библиотеки RT.ParsingLibs
- Библиотека содержит следующие методы (IParsingModule):

1. **About** — возвращает информацию и координаты разработчика, а так же информацию о передаче исключительных прав. Текст о передаче исключительных прав будет выслан дополнительно.
2. **Sources** – метод получает «рубрика, категория, регион», проверяет обрабатывается ли такой набор исходных данных данной библиотекой и возвращает коллекцию названий ресурсов (сайтов) в случае возможной обработки.

3. **Keys**– возвращает коллекцию «рубрика, категория, регион» в структуре сервиса, которые умеет обрабатывать библиотека
4. **Result** – получает рубрику, категорию, регион, ID и DataTime и Хеш-MD5 последнего объявления. Результат содержит все последние объявления размещенные на сайте после ранее полученного последнего объявления, ID и DataTime и Хеш-MD5 которых является входящими параметрами метода . Если параметры последнего входящего объявления пусты, то возвращаемый результат - самое последнее объявление.

- Все выходные данные с библиотеки должны быть по максимуму заполнены
- Получения контента с веба вынести в библиотеку RT.Crawler (добавление, без изменения существующего кода)
- Должен быть создан отдельный проект с unit-тестами MSTest

- Вся кодировка устанавливается в UTF8.
- Все исходники библиотеки должны быть открыты.
- Разработчик библиотеки должен обладать на них авторскими правами и передать исключительные права на использование кодов.
- Если в качестве региона задана область, то объявления возвращаются по загородной недвижимости.
- Одним из условий работы парсера является требование к оптимизации алгоритма получения новых объявлений, т.е. необходимо делать минимальное количество запросов к ресурсу. Для этого парсер должен использовать форму расширенного поиска, чтобы получать выборку нужных объявлений. Затем отсеивать объявления, которые были до IdRes и использовать дальнейшие ссылки только для новых объявлений. Если в выборке полученной от расширенной формы нет объявления с IdRes, то возвращать не более CountAD объявлений и вернуть соответствующий код парсинга

Требования к вашей бд:

- Один файл в формате sqlite
- БД не должна по результатам работы программы увеличиваться в размерах.

Требования к сторонним библиотекам:

- По возможности загружать из хранилища Nuget

Пример реализации парсеров можно найти в библиотеках:

- RT.ParsingLibs.AbstractFirst (вам предоставится исходный код)
- RT.ParsingLibs.AbstractSecond (вам предоставится исходный код)

Требования к документации:

- Комментарии в коде
- Сложные алгоритмы должны быть описаны в отдельном документе docx

Термины

ID объявления - уникально в пределах ресурса и составляется самим ресурсом. Назовем его IdRes. Этот IdRes нам нужен для того, чтобы контролировать с какого ранее найденного объявления нам нужно получить более свежие объявления. Данный IdRes может быть числом, датой и временем размещения объявления или ссылкой. В пределах ресурса IdRes может быть скрыт от нас и нам не известен, поэтому мы должны сами идентифицировать эти объявления. Что выбрать в качестве IdRes на ресурсе определяет сам разработчик, который реализует парсинг ресурса. При первом вызове метода парсера из DLL не известно о IdRes объявления внутри ресурса, поэтому он передает значение NULL. В этом случае парсер должен выбрать самое последнее объявление и вернуть его IdRes ресурса. Если IdRes не найден, то парсер возвращает последнии CountAD объявлений (такая ситуация может получится, если размещенное объявление к следующей итерации будет удалено на ресурсе) Т.к. источников много и заранее не известен каким будет IdRes, то для IdRes зарезервировано 3 переменных ID и DataTime и Хеш-MD5 последнего объявления. Каким из них пользоваться решает разработчик.

Регион – это населенный пункт (мир, страна, федеральный округ, область, город, поселок …). Справочник регионов в конечном виде имеет древовидную структуру.

Рубрика (каталог) - содержит только те рубрики, которые сервис умеет обрабатывать

Действие (категория) – куплю, продам, обменяю, арендую и т.д.

Бинд – это тройка идентификаторов региона, рубрики и действия