**EUCases Crawler Tools**

**Developed by Apis Hristovich EOOD**

**Build Instructions**

Table of Contents

1 Introduction 3

2 Software Requirements 3

3 Build Procedure 3

3.1 Obtaining source code 3

3.2 Building source code 3

3.3 Projects description 4

3.4 Crawler web service and crawler tools usage 5

# Introduction

APIS is responsible for the development of the crawler framework and tools for selected public legal portals in Bulgaria, Austria, Germany, France, United Kingdom as well as the EUR-Lex portal and the HUDOC database of the European Court of Human Rights. The crawler framework consisting of the program modules Crawler Web service and Crawlers is developed on Microsoft Visual Studio with .Net Framework 4.5, Entity Framework 6, Microsoft SQL Server 2012 standard edition, SQLite v3 database. For hosting of the web service Microsoft Internet Information Server is used.

# Software Requirements

In order to build the Crawler web service and the Crawlers modules the following software requirements have to be met:

|  |  |
| --- | --- |
| **Technology** | **Description** |
| .NET Framework | 4.5 version or later |
| Microsoft Visual Studio | Visual C# 2013 |
| SQL Server Database | Microsoft SQL Server 2012 standard edition, SQLite v3 |
| Operating system | Windows Server 2012 for Crawler web service, Windows based OS supporting .Net Framework 4.5 or later for Crawlers |
| Web Server | Microsoft Internet Information Server 7.0 or later |

# Build Procedure

# Obtaining source code

The source code is published on following site:

<SITE>

# Building source code

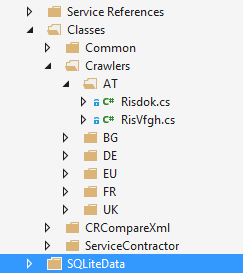
In order to build the source code you will need Microsoft Visual Studio 2013 and .Net Framework 4.5 or later.

All sources for the Crawler web service and Crawlers are included in the solution ‘CrawlerFramework’ which contains the following projects:

* CrawlerEUCases – crawlers developed by APIS
* CrawlerEUCases.Data – Entity Framework objects used by Crawlers to work with SQLite database
* MetainfoEUCases – contains objects describing MetaInfo xml for each downloaded document
* ServiceEUCases – Crawler Web service
* ServiceEUCases.Data – Entity Framework objects used by Crawler Web service to manipulate data in Microsoft SQL Server 2012

# Projects description

* + 1. **CrawlerEUCases** – This project is command line application and contains the source code for all crawlers developed by Apis. Depending on the application configuration different crawler will be executed. The project has the following structure:



* Directory **“Service References”** – contains reference to Crawler Web Service.
* Directory **“Classes”** – contains all units needed for Crawlers realisation. Each crawler is a single class in a separate unit and is positioned in a subdirectory with the name of the country from where documents are downloaded. For instance, all crawlers for Bulgaria are in directory “BG”.
* Directory **“SQLiteData”** – contains thе file:
* SQLiteCrawlerData.sqlite – the empty Database needed from crawlers to store information.

The project contains references to:

* **CrawlerEUCases.Data**
* **MetainfoEUCases**
* **Entity Framework 6 –** externalpackage
* **SharpZipLib** – external package for compressing documents into zip package for Crawlers Web Service
* **HtmlAgilityPack** – external package for processing crawled html documents
  + 1. **CrawlerEUCases.Data** – This project contains auto generated classes from Entity Framework 6 and ensures communication with the SQLite database.

The project contains a directory named “Scripts” with the file:

* SQLiteScript.txt – contains script for creating database structure from scratch.
  + 1. **MetainfoEUCases** – contains classes describing MetaInfoXML and enumeration for the change type of each document.
    2. **ServiceEUCases –** This project is realisation of the Crawler Web service and is developed on Windows Communication Foundation technology. Service export one method “Upload data” and ensures data validation and centralized storage of all submitted documents.

The project contains references to:

* **ServiceEUCases.Data**
* **MetainfoEUCases**
* **Entity Framework 6 –** externalpackage
* **DotNetZip –** externalpackage.
  + 1. **ServiceEUCases.Data –** This project contains auto generated classes from Entity Framework 6 and ensures communication with Microsoft SQL Server 2012 database

# Crawler web service and crawler tools usage

* + 1. **Publishing Crawler web service**

The user must install at least the IIS 7.0 from the Windows distribution kit and follow the instructions from Microsoft Website or other relevant source. Experienced dot Net developer and IIS administrator should create virtual directory pointing to directory containing result files from publishing of ServiceEUCases project.

* + 1. **Crawler Web service configuration**

The Crawler Web service reads the configuration from a web.config file.

* **Database configuration**

<add name="CrawlerFrameworkEntities" connectionString="metadata=res://\*/CrawlerFrameworkModel.csdl|res://\*/CrawlerFrameworkModel.ssdl|res://\*/CrawlerFrameworkModel.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=YourSqlServer;initial catalog=CrawlerFramework;persist security info=True;user id=YourUserId;password=YourPassword;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

* **Windows Communication Foundation** – in order to process large files (up to 20 MB) some custom configuration needs to be done.

<system.serviceModel>

   <services>

      <service name="ServiceEUCases.ServiceEUCases">

       <endpoint address="" binding="basicHttpBinding" bindingConfiguration="CustomConfig" contract="ServiceEUCases.IServiceEUCases" />

      </service>

    </services>

    <bindings>

      <basicHttpBinding>

        <binding name="CustomConfig" closeTimeout="00:10:00" openTimeout="00:10:00" sendTimeout="00:10:00" maxBufferPoolSize="2147483647" maxBufferSize="2147483647" maxReceivedMessageSize="2147483647" messageEncoding="Mtom">

          <readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647" maxArrayLength="2147483647" maxBytesPerRead="2147483647" maxNameTableCharCount="2147483647" />

        </binding>

      </basicHttpBinding>

    </bindings>

    <behaviors>

      <serviceBehaviors>

        <behavior>

          <!- To avoid disclosing metadata information, set the values below to false before deployment -->

          <serviceMetadata httpGetEnabled="true" httpsGetEnabled="true" />

         <!- To receive exception details in faults for debugging purposes, set the value below to true.  Set to false before deployment to avoid disclosing exception information -->

          <serviceDebug includeExceptionDetailInFaults="false" />

        </behavior>

      </serviceBehaviors>

    </behaviors>

    <protocolMapping>

      <add binding="basicHttpsBinding" scheme="https" />

    </protocolMapping>

    <serviceHostingEnvironment aspNetCompatibilityEnabled="true" multipleSiteBindingsEnabled="true" />

</system.serviceModel>

* **Database creation**

ServiceEUCases.Data project contains the file CrawlerFramework.sql in the subdirectory “Scripts”. This file contains all sql scripts needed for initial creation of database. Be careful, because the script will drop and recreate database with the name “CrawlerFramework”.

* + 1. **Crawler tools usage**

As mentioned above, all crawlers are in a single command line application and depending on the configuration file different site will be crawled. Crawlers read configuration from the app.conig file.

* **Main configuration –** the section “<appSettings>” contains the following keys:
  + DestinationFolder – working directory for each crawler
  + Language – language of the crawled documents
  + MaxDegreeOfParallelism – count of parallel documents to be downloaded
  + CrawlerName – name of the crawler to be executed. This is a very important key, because it specifies the name of the crawler class who is responsible for downloading the corresponding web site.

For instance:

<appSettings>

   <add key="DestinationFolder" value="D:\Crawler\Bundespatentgericht" />

    <add key="Language" value="DE" />

    <add key="MaxDegreeOfParallelism" value="2" />

    <add key="CrawlerName" value="DE.Bundespatentgericht" />

  </appSettings>

* **Service connection configuration –** with this settings the communication with the Crawler web service (timeout session, maximum document size, etc.) is being configured. Currently, the maximum volume of the documents that can submitted to the WCF web service is set to 20 MB.

<system.serviceModel>

    <bindings>

      <basicHttpBinding>

        <binding name="BasicHttpBinding\_IServiceEUCases" closeTimeout="00:10:00"

          openTimeout="00:10:00" receiveTimeout="00:10:00" sendTimeout="00:10:00"

          allowCookies="false" bypassProxyOnLocal="false" maxBufferPoolSize="2147483647"

          maxReceivedMessageSize="2147483647" useDefaultWebProxy="true"

          messageEncoding="Mtom" />

      </basicHttpBinding>

    </bindings>

    <client>

      <endpoint address="http://techno.eucases.eu/CrawlerService/ServiceEUCases.svc"

        binding="basicHttpBinding" bindingConfiguration="BasicHttpBinding\_IServiceEUCases"

        contract="ServiceEUCasesReference.IServiceEUCases" name="BasicHttpBinding\_IServiceEUCases" />

    </client>

  </system.serviceModel>

As long as all different crawlers logic is in the same executable, if more than one crawler is needed to start simultaneously on the same PC/server the following must be done: a copy of the application must be made in as many different folders as needed. For each copy amendments in the App.config file must be made according to the site to be downloaded. For further information see the bullet **Main Configuration** above.

When a crawler is started a check for existence of SQLite DBMS with the according structure is performed (SQLiteData\\SQLiteCrawlerData.sqlite). For further information see **Main configuration –** DestinationFolder. If the database does not exist an empty one is copied from SQLiteData\\SQLiteCrawlerData.sqlite.

After initial download all the documents will be processed and sent to the Crawler web service. For each document MD5 sum and other information is stored in SQLite DB. Depending on the frequency of document updates on the crawled web site crawlers must be run at appropriate time intervals. In each subsequent launch after the initial download of data the crawler performs downloading of all the documents and verification against the information stored in the database. MD5 sum comparison is performed and according to it the crawler decides if a document is in *Add*, *Update*, *Delete* or *None* mode (ADD, UPD, DEL, NONE) and this information is stored in the database.

After download and verification are finished all the documents are sent to the Crawler web service for further processing. It is to be noticed here that if there is no connection with the web service all relevant document information will be sent when the crawler is started next time.

It is recommended that each crawler tool is put as a task in „Task scheduler” of the Windows OS. Appropriate frequency of execution should be selected according to the web site data change frequency, for instance: once a day or once a week.