**Istat Registry Web**

Documentazione tecnica e funzionale

|  |  |
| --- | --- |
| **Version:** | 1.0 |
| **Date:** | 30/09/2015 |
| **Person in charge:** |  |
| **Approved by:** |  |

***Table of contents***

[1 Introduction 4](#_Toc440012749)

[2 Architecture of the application 7](#_Toc440012750)

[3 Technical features 9](#_Toc440012751)

[4 Architettura applicazione 12](#_Toc440012752)

[5 Progetto ISTATRegistry 13](#_Toc440012753)

[5.1 User Control 13](#_Toc440012754)

[5.1.1 AddText 13](#_Toc440012755)

[5.1.2 ArtefactDelete 14](#_Toc440012756)

[5.1.3 ControlAnnotations 15](#_Toc440012757)

[5.1.4 DuplicateArtefact 17](#_Toc440012758)

[5.1.5 FileDownload3 17](#_Toc440012759)

[5.1.6 GetCodelist 19](#_Toc440012760)

[5.1.7 GetConcept 20](#_Toc440012761)

[5.1.8 GetDataFlow 22](#_Toc440012762)

[5.1.9 GetDSD 23](#_Toc440012763)

[5.1.10 SearchBar 25](#_Toc440012764)

[5.2 Artefatti 27](#_Toc440012765)

[5.2.1 Codelist 27](#_Toc440012766)

[5.2.2 Data Structure Definition(DSD) 32](#_Toc440012767)

[5.2.3 Concept scheme 39](#_Toc440012768)

[5.2.4 Category scheme 42](#_Toc440012769)

[5.2.5 Dataflow 43](#_Toc440012770)

[5.2.6 Categorisation 46](#_Toc440012771)

[5.2.7 Hierarchical Codelist 46](#_Toc440012772)

[5.2.8 Agency scheme 47](#_Toc440012773)

[5.2.9 Data provider scheme 50](#_Toc440012774)

[5.2.10 Data Consumer scheme 53](#_Toc440012775)

[5.2.11 Organization unit scheme 56](#_Toc440012776)

[5.2.12 Content constraint 59](#_Toc440012777)

[5.3 StructureSet 63](#_Toc440012778)

[5.4 Importazione artefatti 64](#_Toc440012779)

[5.4.1 Logica funzionale 64](#_Toc440012780)

[5.4.2 Vincoli funzionali 65](#_Toc440012781)

[5.4.3 Abilitazione artefatti importabili 65](#_Toc440012782)

[5.4.4 Salvataggio artefatti 66](#_Toc440012783)

[5.5 Amministrazione utenti 67](#_Toc440012784)

[5.6 File di configurazione (Web.config) 67](#_Toc440012785)

[6 Progetto ISTATEntity 71](#_Toc440012786)

[7 Progetto ISTATEntityMapper 72](#_Toc440012787)

[8 Progetto ISTATWSDAL 76](#_Toc440012788)

# Introduction

The document describes the technical features and functionalities of the web GUI for creating,importing, managing, browsing and downloading the SDMX structural metadata (artefacts) , that is based on the use and enhancement of some modules of the Eurostat’s SDMX- Reference Infrastructure.

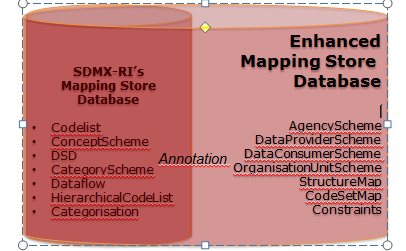
The artefacts that can be managed and browsed by the web GUI are those already managed in the SDMX-RI (CodeList, ConceptScheme, DSD, CategoryScheme, Dataflow, Categorization), and, in addition,

* *AgencyScheme,* ItemScheme artefact representing a set of Agencies.
* *DataProviderScheme,*ItemsScheme artefact representing a set of Data Provider organizations.
* *DataConsumerScheme,*ItemScheme artefact representing a set of Data Consumer organizations.
  + *OrganizationUnitScheme,*ItemScheme artefact representing a set of units inside an organization.
  + *Constraint,* artefact representing the selections of codes of the codelists linked to the components of a specific DSD or of a DSD referenced by a dataflow.
  + The *CodeSetMap,* artefact that allows to correlate two CodeLists by linking together their respective codes.
  + The *StructureMap,* artefact that allows to correlate two keyfamilies (DSDs) by linking together their respective dimensions and/or attributes

It’s possible to import artefacts in SDMX format and also to import items of the ItemSchemes(CodeLists,ConceptSchemes,CategorySchemes,AgencyScheme,DataProviderScheme,DataConsumerScheme) in csv format. It’s possible to export artefacts in SDMX format, and the items of the ItemSchemes in csv format; furthermore is possible to export CodeLists and DSDs (Data StructureDefinitions)in the xml format that can be used in order to create hypercubes and common dimensions inside the .Stat’s datawarehouse..

It’s also possiblepossibleto manage, view, browse and download the artefacts and the related items together with their annotations.

The SDMX artefacts are stored inside the new version of the SDMX-RI’s Mapping Store Database, published by Eurostat at the end of 2014, on the base of a joint work performed among ISTAT and Eurostat for the extension of the old database scheme with the artefacts foreseen for the management thorugh the web gui and the enhanced SDMX-RI web service, in order to create a real repository for SDMX structural metadata, as shown in the Figure 1



**Fig.1 – List of artefacts added in the new Mapping Store database schema**

The mapping store database is in fact an SDMX repository based on a relational DBMS (in ISTAT the Single Exit Point’s Mapping Store database has been implemented in the MS SQL Server 2008 RDBMS environment) in which are stored the structural metadata that are represented in the SDMX Information Model through some specific objects (artefacts) very well defined e described by the related XML schemas.Furthermore, in the Mapping Store are also stored the information related to the mapping of local data stored into a dissemination database (in the case of ISTAT, the I.Statdatawarehouse) against the SDMX data structures.

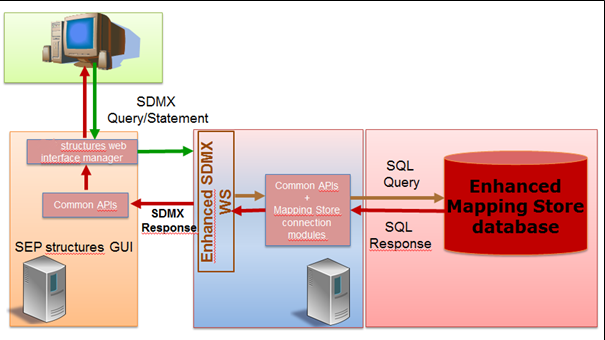
The software is based on the reuse and enhancement of some modules of the SDMX Reference Infrastructure (SDMX-RI): and particularly the Common APIs (SdmxSource .NET)

At present the web gui can work in two modes, according to the value of a specific parameter of the *web.config* file:

* .*Reading mode,* through which is possible to browse, search and download artefacts
* *Management mode,* through which, in addition to the reading functionalities, is possible to import, insert, update, delete artefacts, according with their state (i.e. final/not final for update; referenced/not referenced by other artefacts for deletion)

# Architecture of the application

The application architecture on which is based the web GUI is shown in the figure 2:

****

**Fig.2 – Application architecture of the web GUI**

In this architecture, the web GUI is based on the following layers:

* The *interface manager layer*, on which are based the interfaces for selecting artefacts, browsing the general information and the detail of a single artefact, downloading and uploading the artefacts
* The *SDMX query/statement generation layer*, through which the selections made by the users on the web GUI are translated in SDMX registry queries for retrieving artefacts on the base of the options selected (lists of artefacts; artefacts comprehensive of their details i.e. items; artefacts together with their referenced artefacts;…), or in SDMX statements for inserting or deleting artefacts.
* The *SDMX web service* to which the query/statementsare submitted. This web service is an extended version of the SDMX web service containing, in addition to the methods for querying data and structural metadata (SDMX Artefacts), also a method for submitting artefacts. Is possible to upload and download through the web service artefacts in SDMX v. 2.0 and 2.1.The queries/statements are then submitted to the web serviceand then processed according to the SDMX rules (i.e. is not possible to change a finalised artefact; is not possible to delete artefacts referenced by other artefacts, etc…)
* *Layer for artefacts handling*, that is based on the SDMX Common APIs (SdmxSource .NET) through which the artefacts returned by the SDMX web service are uploaded to SDMX objects in order to be presented or handled through the web GUI

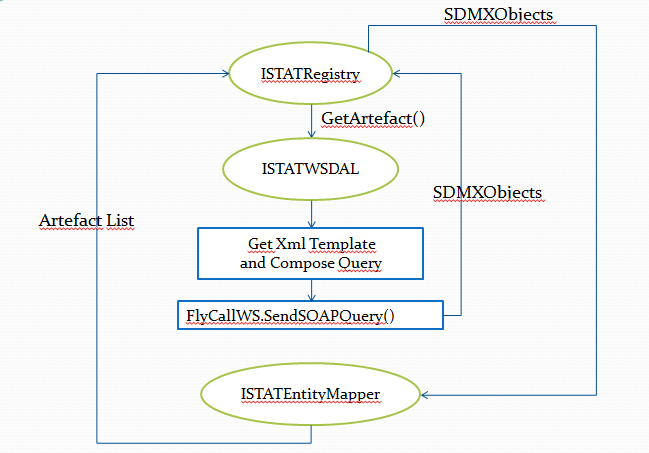
# Technical features

The application has been developed in C# language using the ASP. NET technology for 4.0 .NET framework. Some parts of the user interface have been developed by using also jquery technology (for instance the feature that, when is opened a pop-up window, the underlying page is blocked until the closure of the pop-up itself). The lists of artefacts and the lists of items of ItemSchemes are presented inside parametric grids defined in a general way and populated on the base of the user’s selections. Each selection produces a post-back command through which are generated the queries to be submitted to the SDMX web service and is managed the result through the SDMXObjects of the SdmxSource .NET; the content of the SDMXObjects is then passed to the interface objects (grid bindings text-box, combo-box, etc..)

As far as concerns the back-end modules, the two main functional flows for browsing and managing artefacts, are the following:

* *Artefact retrieval*

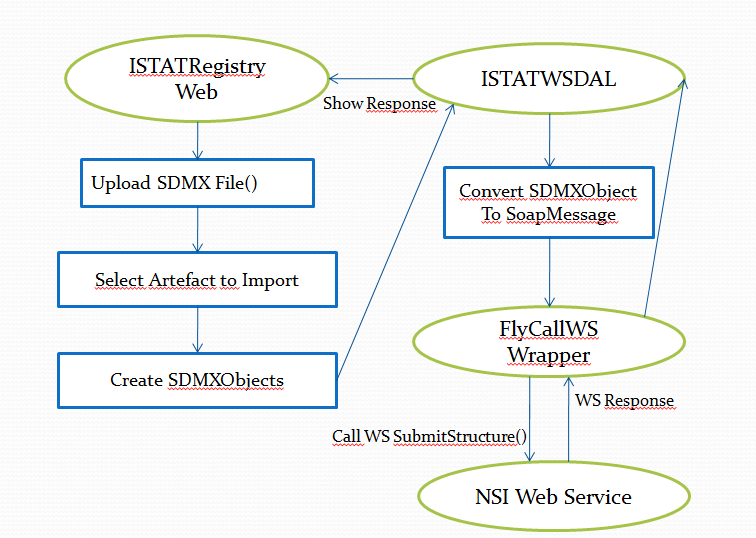
In order to retrieve the artefacts of interest, once selected from the GUI the type of artefact or a single artefact, the related parameters are passed to the ISTATWSDAL module, that, on the base of the SDMX-ML query templates, creates the SDMX-ML query and submit it to the web service through the FlyCallWS.SendSOAPQuery() function. The web service response is passed to the SDMXObiects module and, from it, uploaded to the IstatEntityMapper module, that maps the artefacts information contained in the SDMXObjects against the objects used for populating the web gui (as the grids containing the lists of artefacts or items).

****

**Fig.3 – Artefact retrieval modules workflow**

* *Submit Artefact*

After uploaded the SDMX file containing the artefacts to be imported, the web GUI populates an SDMXObiects in order to allow end users to select the artefacts to be imported, and then the selected artefacts through the SDMXObject are passed to the ISTATWSDAL (the class layer that “wraps” the interaction to the web service) that converts it in a XML stream and then submits this stream to the “SubmitStructure” method of the web service.

****

**Fig.4 – Submit artefact modules workflow**

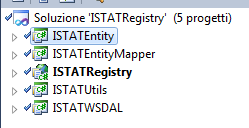
In applicative terms, the ASP.NET solution is composed by the following 5 projects:

* *ISTATEntity:* this project managesthe entities used by the Web Gui (implementedwith the only properties needed for visualization and storicization); it also contains the classes used for populating the grids (DataBind)
* *ISTATEntityMapper:* this project, that consists in a only one class named “EntityMapper”, that maps the data retrieved by the SdmxObjectsagainst the ISTATEntity objects.
* *ISTATRegistry:* the main ASP .NET project
* *ISTATUtils:* Project that contains Utility class for the localization of the application (LocalizedUtils)
* *ISTATWSDAL:* Project implementing the wrapper for accessing to the web service. It exposes the interfaces for the artefacts retrieval.

# Architettura applicazione

L'applicazione è stata sviluppata con tecnologie Microsoft su framework .NET versione 4.0, utilizza C# come linguaggio di programmazione. La soluzione si compone di 5 progetti:

* ISTATEntity
  + Contiene le Entity class su cui vengono rimappati i dati
* ISTATEntityMapper
  + Utilizzata per il mapping di dati SDMX su classi interne
* ISTATRegistry
  + Applicazione Web
* ISTATUtils
  + Contiene classi di utilità
* ISTATWSDAL
  + Libreria utilizzata per accedere al Web Service NSI



# Progetto ISTATRegistry

Questo è il progetto web della solution che permette la visualizzazione ed il CRUD degli artefatti presenti nel Registry DB.

## User Control

Al fine di una maggiore modularità e riutilizzabilità del codice, sono stati sviluppati una serie di UserControl che possono essere inclusi nelle pagine .aspx.

Gli UserControl sono locati nella cartella 'UserControls' del progetto Web 'ISTATRegistry'

### AddText

Questo controllo permette l'inserimento del testo localizzato. Viene utilizzato nelle informazioni generali degli artefatti per inserire il Name e la Description e nelle Annotation per inserire il testo.

Il controllo include una GridView per la visualizzazione dei testi e un form che ne consente l'inserimento e la modifica.

Il controllo espone le seguenti proprietà pubbliche:

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Tipo*** | ***Funzionalità*** |
| ucOpenTabName | String | Questa stringa viene valorizzata con il nome del Tab(div html) che deve essere riaperto a seguito di un postback |
| ucOpenPopUpName | String | Questa stringa viene valorizzata con il nome della PopUp(div html) che deve essere riaperto a seguito di un postback |
| ucOpenPopUpWidth | Integer | Definisce la larghezza in pixel della PopUp che conterrà il controllo |
| ucEnableSuggest | Boolean | Questo parametro visualizza o meno i suggerimenti per le Annotation. Se valorizzato con true verrà incluso nel form la DropDownlList dei suggerimenti |
| ucEditMode | Boolean | Questo parametro visualizza il controllo in modalità di editing(true) o in read only(false) |
| ArtefactType | SdmxStructureEnumType | Enum che identifica la tipologia di artefatto(codelist, dsd, ecc...) che includerà il controllo |
| TType | TextType | Enum che identifica la tipologia di testo(Name o Description) |
| TextObjectList | IList<ITextTypeWrapperMutableObject> | Lista di riferimento che conterrà i testi inseriti |

### ArtefactDelete

Questo controllo consente l'eliminazione di un artefatto dal database. Una volta incluso e configurato correttamente può eliminare un artefatto tramite una chiamata soap.

Il controllo una volta renderizzato visualizza un link con un'icona raffigurante una X, cliccandoci sopra viene aperta una popup che riassume i dati principali dell'artefatto che si desidera cancellare, cliccando sul bottone Delete l'artefatto verrà eliminato fisicamente dal DB.

Il controllo espone le seguenti proprietà pubbliche:

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Tipo*** | ***Funzionalità*** |
| ucID | String | Questa stringa viene valorizzata con l'ID sdmx dell'artefatto |
| ucAgency | String | Questa stringa viene valorizzata con l'Agency ID sdmx dell'artefatto |
| ucVersion | String | Questa stringa viene valorizzata con la versione sdmx dell'artefatto |
| ucArtefactType | SdmxStructureEnumType | Enum che identifica la tipologia di artefatto(codelist, dsd, ecc...) che includerà il controllo |
| ucButtonClientId | String | Restituisce il Client ID del controllo Button che effettua il save |
| ucCanDeleteThis | Boolean | Questo booleano è valorizzato con true se l'artefatto può essere eliminato, altrimenti con false |

### ControlAnnotations

Questo controllo viene incluso nelle pagine .aspx per la gestione delle Annotation. Permette l'inserimento, la modifica e la cancellazione.

Il controllo include una GridView per la visualizzazione delle annotation e un form che ne consente l'inserimento e la modifica.

Per gestire l'inserimento dei testi il controllo include lo Usercontrol 'AddText'

Il controllo espone le seguenti proprietà pubbliche:

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Tipo*** | ***Funzionalità*** |
| AnnotationObjectList | IList<IAnnotationMutableObject> | Lista di oggetti AnnotationMutableObject che conterrà le annotation inserite |
| EditMode | Boolean | Questo parametro visualizza il controllo in modalità di editing(true) o in read only(false) |
| PopUpContainer | String | Questa stringa viene valorizzata con il nome della PopUp(div html) che deve essere riaperto a seguito di un postback |
| PopUpContainerWidth | Integer | Definisce la larghezza in pixel della PopUp che conterrà il controllo |
| AddText\_ucOpenTabName | String | Questa stringa viene valorizzata con il nome del Tab(div html) che deve essere riaperto a seguito di un postback |
| AddText\_ucOpenPopUpName | string | Questa stringa viene valorizzata con il nome della PopUp(div html) che deve essere riaperto a seguito di un postback |
| AddText\_ucOpenPopUpWidth | Integer | Definisce la larghezza in pixel della PopUp che conterrà il controllo |
| OwnerAgency | Boolean | Questo booleano è valorizzato con true se l'utente collegato è associato all'agenzia a cui l'artefatto appartiene. In caso positivo verrà visualizzata l'iconcina a forma di matita(edit), altrimenti l'icona a forma di paginetta(view). |

### DuplicateArtefact

Questo controllo permette la duplicazione di un artefatto. Visualizza tramite un form html i dati identificativi(id,agency,version) dell'artefatto sorgente(in read only) e i dati identificativi dell'artefatto di destinazione. L'utente collegato se abilitato all'editing potrà duplicare l'artefatto variando uno dei dati identificativi e cliccando sul tasto 'duplica'. Le agenzie presenti nella dropdownlist sono quelle associate all'utente.

I controlli form dell'artefatto di destinazione potranno essere abilitate tutti o parzialmente.

Il controllo espone le seguenti proprietà pubbliche:

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Tipo*** | ***Funzionalità*** |
| ucMaintanableArtefact | IMaintainableMutableObject | L'artefatto da duplicare |
| ucStructureType | SdmxStructureEnumType | Identifica la tipologia di artefatto |
| ucReadOnlyID | Boolean | Se valorizzato con true l'ID dell'artefatto di destinazione non potrà essere variato |
| ucReadOnlyAgency | Boolean | Se valorizzato con true l'AgencyID dell'artefatto di destinazione non potrà essere variato |
| ucReadOnlyVersion | Boolean | Se valorizzato con true la versione dell'artefatto di destinazione non potrà essere variato |
| ucDisable | Boolean | Abilita o disabilita il controllo |

### FileDownload3

Questo controllo permette il download di un artefatto nei seguenti formati:

* SDMX-ML versione 2.0
* SDMX-ML versione 2.1
* CSV
* Formato .STAT

Il controllo una volta renderizzato espone a video un imageButton che avvia il download, cliccandoci sopra viene aperta una popup che visualizza un form di selezione che contiene:

* una dropdownlist con le varie tipologie di download(sdmx, csv, ecc..)
* un checkbox per l'export in stub(solo i dati base dell'artefatto)
* una textbox per il separatore csv

Il controllo espone le seguenti proprietà pubbliche:

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Tipo*** | ***Funzionalità*** |
| ucID | String | Questa stringa viene valorizzata con l'ID sdmx dell'artefatto |
| ucAgency | String | Questa stringa viene valorizzata con l'Agency ID sdmx dell'artefatto |
| ucVersion | String | Questa stringa viene valorizzata con la versione sdmx dell'artefatto |
| ucArtefactType | SdmxStructureEnumType | Enum che identifica la tipologia di artefatto(codelist, dsd, ecc...) che includerà il controllo |

### GetCodelist

Il controllo permette di selezionare un artefatto di tipo codelist da una lista dello stesso tipo. Il form permette di filtrare i risultati tramite lo Usercontrol 'SearchBar'.

Effettuata l'operazione di selezione lo usercontrol effettua le seguenti operazioni:

1. Valorizza la proprietà 'text' del controllo 'TargetWebControl' passato in input con i dati identificativi dell'artefatto. La stringa restituita viene formattata nel formato: "ID,Agency,Version".
2. Genera un evento che restituisce un oggetto di tipo Datatable a cui aggiunge una riga contenente i dati identificativi dell'artefatto selezionato. Se il Datatable viene passato in input popolato viene inserita una ulteriore riga solo se i dati identificativi non sono già presenti. Questo evento viene utilizzato per gestire la selezione multipla.

Il controllo espone le seguenti proprietà pubbliche:

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Tipo*** | ***Funzionalità*** |
| TargetWebControl | ITextControl | Il controllo web di tipo ITextControl che verrà popolato con i dati identificativi dell'artefatto selezionato |
| ucTargetDataTable | DataTable | Il DataTable che verrà popolato con gli identificativi degli artefatti selezionati. Utilizzato per la multi selezione. |
| ucOpenTabName | String | Questa stringa viene valorizzata con il nome del Tab(div html) che deve essere riaperto a seguito di un postback |
| ucOpenPopUpName | String | Questa stringa viene valorizzata con il nome della PopUp(div html) che deve essere riaperto a seguito di un postback |
| ucOpenPopUpWidth | Integer | Definisce la larghezza in pixel della PopUp che conterrà il controllo |
| ucAddIconType | AddIconType | Enum che identifica la tipologia di icona da visualizzare(matita o croce) |
| ucIsFinalArtefact | Boolean | Valorizzato con true visualizza solo gli artefatti finalizzati, altrimenti tutti. |

### GetConcept

Il controllo permette la selezione di un artefatto di tipo ConcetpScheme o di un Concept ad esso associato. Il form permette di filtrare i risultati tramite lo Usercontrol 'SearchBar'.

Se configurato per la selezione di tipo ConceptScheme viene visualizzata una Gridview da cui si potrà effettuare la selezione.

Se configurato per la selezione di tipo Concept, dopo la selezione del ConceptScheme verrà selezionata un'ulteriore griglia che consentirà la selezione di uno dei concept da esso contenuti.

Restituisce i dati identificativi dell'artefatto in una stringa formattata nel formato: "ID,Agency,Version".

Il controllo espone le seguenti proprietà pubbliche:

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Tipo*** | ***Funzionalità*** |
| TargetWebControl | ITextControl | Il controllo web di tipo ITextControl che verrà popolato con i dati identificativi dell'artefatto selezionato |
| ucOpenTabName | String | Questa stringa viene valorizzata con il nome del Tab(div html) che deve essere riaperto a seguito di un postback |
| ucOpenPopUpName | String | Questa stringa viene valorizzata con il nome della PopUp(div html) che deve essere riaperto a seguito di un postback |
| ucOpenPopUpWidth | Integer | Definisce la larghezza in pixel della PopUp che conterrà il controllo |
| ucAddIconType | AddIconType | Enum che identifica la tipologia di icona da visualizzare(matita o croce) |
| IsConceptSelection | Boolean | Valorizzato con true permette la selezione del concept, con false invece la selezione si fermerà a livello di ConceptScheme |
| ucIsFinalArtefact | Boolean | Valorizzato con true visualizza solo gli artefatti finalizzati, altrimenti tutti. |

### GetDataFlow

Il controllo permette di selezionare uno o più artefatti di tipo Dataflow da una lista dello stesso tipo. Il form permette di filtrare i risultati tramite lo Usercontrol 'SearchBar'.

Effettuata l'operazione di selezione lo usercontrol effettua le seguenti operazioni:

1. Valorizza la proprietà 'text' del controllo 'TargetWebControl' passato in input con i dati identificativi dell'artefatto. La stringa restituita viene formattata nel formato: "ID,Agency,Version".
2. Genera un evento che restituisce un oggetto di tipo Datatable a cui aggiunge una riga contenente i dati identificativi dell'artefatto selezionato. Se il Datatable viene passato in input popolato viene inserita una ulteriore riga solo se i dati identificativi non sono già presenti. Questo evento viene utilizzato per gestire la selezione multipla.

Il controllo espone le seguenti proprietà pubbliche:

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Tipo*** | ***Funzionalità*** |
| TargetWebControl | ITextControl | Il controllo web di tipo ITextControl che verrà popolato con i dati identificativi dell'artefatto selezionato |
| ucOpenTabName | String | Questa stringa viene valorizzata con il nome del Tab(div html) che deve essere riaperto a seguito di un postback |
| ucTargetDataTable | DataTable | Il DataTable che verrà popolato con gli identificativi degli artefatti selezionati. Utilizzato per la multi selezione. |
| ucOpenPopUpName | String | Questa stringa viene valorizzata con il nome della PopUp(div html) che deve essere riaperto a seguito di un postback |
| ucOpenPopUpWidth | Integer | Definisce la larghezza in pixel della PopUp che conterrà il controllo |
| ucAddIconType | AddIconType | Enum che identifica la tipologia di icona da visualizzare(matita o croce) |
| ucIsFinalArtefact | Boolean | Valorizzato con true visualizza solo gli artefatti finalizzati, altrimenti tutti. |

### GetDSD

Il controllo permette di selezionare uno o più artefatti di tipo DSD da una lista dello stesso tipo. Il form permette di filtrare i risultati tramite lo Usercontrol 'SearchBar'.

Effettuata l'operazione di selezione lo usercontrol effettua le seguenti operazioni:

1. Valorizza la proprietà 'text' del controllo 'TargetWebControl' passato in input con i dati identificativi dell'artefatto. La stringa restituita viene formattata nel formato: "ID,Agency,Version".
2. Genera un evento che restituisce un oggetto di tipo Datatable a cui aggiunge una riga contenente i dati identificativi dell'artefatto selezionato. Se il Datatable viene passato in input popolato viene inserita una ulteriore riga solo se i dati identificativi non sono già presenti. Questo evento viene utilizzato per gestire la selezione multipla.

Il controllo espone le seguenti proprietà pubbliche:

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Tipo*** | ***Funzionalità*** |
| TargetWebControl | ITextControl | Il controllo web di tipo ITextControl che verrà popolato con i dati identificativi dell'artefatto selezionato |
| ucOpenTabName | String | Questa stringa viene valorizzata con il nome del Tab(div html) che deve essere riaperto a seguito di un postback |
| ucTargetDataTable | DataTable | Il DataTable che verrà popolato con gli identificativi degli artefatti selezionati. Utilizzato per la multi selezione. |
| ucOpenPopUpName | String | Questa stringa viene valorizzata con il nome della PopUp(div html) che deve essere riaperto a seguito di un postback |
| ucOpenPopUpWidth | Integer | Definisce la larghezza in pixel della PopUp che conterrà il controllo |
| ucAddIconType | AddIconType | Enum che identifica la tipologia di icona da visualizzare(matita o croce) |
| ucIsFinalArtefact | Boolean | Valorizzato con true visualizza solo gli artefatti finalizzati, altrimenti tutti. |

### SearchBar

Il controllo SearchBar espone un form atto a filtrare le liste di artefatti.

Il controllo espone le seguenti proprietà pubbliche:

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Tipo*** | ***Funzionalità*** |
| FixedSearch | Boolean | Se impostato a True il form di ricerca sarà sempre visibile, valorizzato con False sarà chiuso |
| SearchBarID | String | Stringa da valorizzare per rendere il controllo univoco, nel caso in cui fossero presenti più istanze nella stessa pagina. |
| ucID | String | Stringa contenente l'ID dell'artefatto da ricercare |
| ucAgency | String | Stringa contente l'agenzia dell'artefatto da ricercare |
| ucVersion | String | Stringa contente la versione dell'artefatto da ricercare |
| ucName | Boolean | Stringa contente il name dell'artefatto da ricercare |
| BtnSearch | Button | Riferimento al bottone per l'avvio della ricerca. Da utilizzare per l'implementazione dell'Event Handler nella pagina chiamante, |

Codice di esempio per l'implementazione dell'event handler e filtro nella pagina chiamante:

protected void Page\_Load(object sender, EventArgs e)

{

SearchBar1.BtnSearch.Click += new EventHandler(btnSearch\_Click);

}

protected void btnSearch\_Click(object sender, EventArgs e)

{

WSModel wsModel = new WSModel();

ISdmxObjects sdmxInput;

ISdmxObjects sdmxFinal;

LocalizedUtils localizedUtils = new LocalizedUtils(Utils.LocalizedCulture);

IMutableObjects mutableObj = new MutableObjectsImpl();

sdmxInput = wsModel.GetDataStructure(

new ArtefactIdentity(SearchBar1.ucID,

SearchBar1.ucAgency,

SearchBar1.ucVersion), true,true);

// Codice per filtrare gli artefatti by name

foreach (IDataStructureObject ds in sdmxInput.DataStructures)

{

if (localizedUtils.GetNameableName(ds).ToUpper().Contains(

SearchBar1.ucName.Trim().ToUpper()))

mutableObj.AddDataStructure(ds.MutableInstance);

}

sdmxFinal = mutableObj.ImmutableObjects;

gridViewDataBind(sdmxFinal);

}

## Artefatti

### Codelist

La gestione dell'artefatto codelist si suddivide in 2 pagine .aspx: 'codelist.aspx' e 'codelistItemDetails.aspx'. La prima visualizza a video la lista delle codelist(in formato tabellare) e ne permette l'eliminazione(se l'utente è abilitato) e download. La seconda invece gestisce la visualizzazione del dettaglio e il CRUD dell'artefatto.

#### Codelist.aspx

La pagina codelist.aspx oltre alla visualizzazione delle codelist permette il download degli artefatti. Se l'utente è abilitato, nella griglia sarà presente l'icona per la cancellazione e a fondo pagina un bottone per la creazione di un nuovo artefatto. Nell'Event Handler dell'evento click associato al bottone verrà effettuato un redirect alla pagina CodelistItemDetail.aspx che ne permetterà la creazione. Il delete di un artefatto non è consentito se nel db è presente un vincolo di foreign key verso un'entità ad esso associata, in questo caso l'applicazione restituirà un errore.

##### Logica di esecuzione

Le operazioni che vengono svolte dal code behind della pagina per popolare la griglia sono le seguenti:

* Viene creata un'istanza dell'oggetto ISTAT.WSDAL.WSModel() (WSModel wsModel = new WSModel();)
* Viene richiamato il metodo GetCodeList() a cui viene passato un object entity di tipo ArtefactIdentity, questo oggetto contiene le proprietà 'ID','Agency' e 'Version' che se valorizzate filtreranno i risultati, in caso contrario verrà restituito l'intero resultset. I risultati saranno restituiti in un oggetto di tipo ISdmxObjects.
* Il risultato restituito verrà rimappato in una lista di oggetti di tipo ISTAT.Entity.CodeList. Il mapping verrà eseguito dal metodo GetCodeListList() della classe ISTAT.EntityMapper. EntityMapper() (List<ISTAT.Entity.CodeList> lCodeList = eMapper.GetCodeListList(\_sdmxObjects, Utils.LocalizedLanguage)
* Una volta ottenuta la lista di entità di tipo Codelist()(List<ISTAT.Entity.CodeList>) verrà effetuato il DataBind() della gridview.

##### Visualizzazione dettaglio

Cliccando su una delle iconcine del dettaglio presenti nella griglia, verrà effettuato un redirect alla pagina CodelistItemDetail.aspx, a cui verra passata in modalità 'GET' la chiave(id,agency e version) dell'artefatto richiesto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina Codelist.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| SearchBar | Utilizzato per filtrare la griglia dei risultati |
| ArtefactDelete | Permette di eliminare un artefatto previa conferma |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV * .Stat |

#### CodelistItemDetails.aspx

Questa pagina offre funzionalità di gestione su artefatti di tipo Codelist.

Funzionalità:

* CRUD Artefatto
* Duplicazione Artefatto
* Importazione Item da file CSV
* Inserimento puntuale su Codelist finalizzata
* Download Artefatto

##### CRUD Codelist

Il CRUD della Codelist è gestito in 2 TAB separati, il primo per la gestione delle informazioni principali dell'artefatto(ID, Agency,Version, Isfinal, ecc...), mentre il secondo di occupa della gestione degli Item code.

* Creazione nuova codelist:
  + In fase di creazione di una nuova Codelist, i dati inseriti tramite interfaccia grafica(form) vengono memorizzati in un oggetto di tipo ISdmxObjects. L'oggetto una volta popolato viene inserito nel DB tramite la chiamata del metodo SubmitStructure() della libreria ISTAT.WSDAL.WSModel.  
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
     wsModel.SubmitStructure(sdmxObjects);
* Modifica codelist esistente
  + In fase di modifica viene richiesto al Web service un artefatto di tipo Codelist che poi verrà utilizzato per popolare i form presenti nella pagina. La chiave dell'artefatto(ID,Agency e Version) viene passata in modalità GET dalla pagina chiamante e riletta tramite Request dalla pagina ricevente.   
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
    ArtefactIdentity artIdentity = Utils.GetIdentityFromRequest(Request);  
    ISdmxObjects sdmxObject = wsModel.GetCodeList(artIdentity);
  + Apportate le modifiche, verrà richiamato il metodo SubmitStructure()(analogamente alla creazione della Codelist) che si occuperà dell'aggiornamento dell'Artefatto nel DB.

##### Duplicazione artefatto

Se l'utente è abilitato sarà visibile lo User Control DuplicateArtefact che esporrà il bottone 'Duplicazione Artefatto'. Questo controllo deve essere inizializzato con la Chiave SDMX dell'artefatto.

##### Importazione item da file CSV

Nel TAB dei CODES è presente la funzionalità di importazione codici tramite file .csv.   
Di seguito la logica di esecuzione:

* Importazione del file csv
* Validazione del file
* La lista dei codici viene inserita nell'oggetto ICodelistMutableObject a sua volta memorizzato in un oggetto di sessione(Session[])

##### Inserimento puntuale su Codelist finalizzata

Normalmente tutti gli artefatti di tipo 'Final' non possono essere modificati, ma per gli Item della Codelist è presente un eccezione. La funzionalità di inserimento puntuale permette l'inserimento di un nuovo item nella Codelist. Per l'inserimento del nuovo code è presente un form contenente le informazioni di base, sul click del bottone submit questi dati vengono inviati al WS IRService che effettuerà un inserimento diretto nel DB.

##### Download Artefatto

Se l'utente è abilitato sarà visibile lo User Control FileDownload che visualizzerà a video un ImageButton che permetterà il download dell'artefatto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina CodelistItemDetails.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| DuplicateArtefact | Permette la duplicazione di un artefatto |
| ControlAnnotations | Gestisce il CRUD delle annotation sia a livello di artefatto che di Item |
| AddText | Gestisce il CRUD sui text(Name,Description) dell'artefatto e dell'Item |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV * .Stat |

### Data Structure Definition(DSD)

Tutte le informazioni che caratterizzano la DSD vengono memorizzate in un oggetto di tipo ISdmxObjects. L'oggetto una volta popolato viene inserito nel DB tramite la chiamata del metodo SubmitStructure() della libreria ISTAT.WSDAL.WSModel.

Esempio di codice:   
WSModel wsModel = new WSModel();  
wsModel.SubmitStructure(sdmxObjects);

La gestione della DSD si suddivide in 5 aree:

* Generale
  + Contiene le informazioni principali dell'artefatto
* Misura Principale
  + Form per la gestione della Primary Measure
* Dimensioni
  + Permette la visualizzazione delle dimensioni in formato tabellare
  + L'inserimento e la modifica di Dimensioni esistenti
* Gruppi
  + Permette la visualizzazione dei Dimension Group in formato tabellare
  + L'inserimento e la modifica di Dimension Group esistenti
* Attributi
  + Permette la visualizzazione degli Attributi in formato tabellare
  + L'inserimento e la modifica degli Attributi esistenti

#### Generale

La sezione Generale si occupa della gestione dei dati principali dell'artefatto.  
Nel dettaglio:

* ID: Identificativo dell'artefatto
* Agency: Identificativo dell'agenzia
* Version: Versione dell'artefatto
* URI: Uri dell'artefatto
* URN:Urn dell'artefatto
* Finalizzata: Checkbox che identifica se l'artefatto è finalizzato
* Valido da: Textbox associato ad un controllo di tipo Calendar che contiene la data inizio validità dell'artefatto
* Valido a: Textbox associato ad un controllo di tipo Calendar che contiene la data di fine validità dell'artefatto
* Nome: Usercontrol per la gestione dei Name
* Descrizione: Usercontrol per la gestione delle descrizioni
* Annotazioni: Usercontrol per la gestione delle Annotazioni

##### UserControl utilizzati

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| DuplicateArtefact | Permette la duplicazione di un artefatto |
| ControlAnnotations | Gestisce il CRUD delle annotation |
| AddText | Gestisce il CRUD dei text(Name,Description) |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati Gestisce il download dell'artefatto |

#### Misura principale

La sezione Misura principale espone all'utente i seguenti controlli:

* ID: ID della misura. E' una costante valorizzata con 'OBS\_VALUE'
* Concetto: UserControl per la selezione di un artefatto di tipo ConceptScheme
* Codelist: UserControl per la selezione di un artefatto di tipo Codelist
* Annotazioni: Usercontrol per la gestione delle annotazioni

##### UserControl utilizzati

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| ControlAnnotations | Gestisce il CRUD delle annotation |
| GetConcept | Permette la selezione di un artefatto di tipo ConceptScheme |
| GetCodelist | Permette la selezione di un artefatto di tipo Codelist |

#### Dimensioni

La sezione Dimensioni visualizza tramite un controllo di tipo GridView la lista di tutte le dimensioni appartenenti alla DSD. Nella griglia è presente un imagebutton che consente l'eliminazione di una dimensione previa conferma e un altro imagebutton che apre una popup contenente uno UserControl di tipo ControlAnnotations. I dati relativi alle dimensioni non possono essere modificati, per effettuare una modifica sarà necessario eliminare poi reinserire la dimensione.

Nella parte inferiore, sotto la gridview, è presente un imageButton che apre una popup contenente un form per l'inserimento di una nuova dimensione.  
Il form visualizza una dropdownlist che definisce la tipologia di dimensione.  
In base alla selezione di una delle voci della dropdownlist il form varierà il suo aspetto visualizzando le seguenti informazioni:

* NORMAL:
  + ID: Textbox editabile per l'inserimento di testo libero
  + Riferimento al concetto: Usercontrol di tipo GetConcept che consente la selezione del concetto
  + Lista codificata: Usercontrol di tipo GetCodelist
* FREQUENCY
  + ID: Textbox non editabile contenente il valore FREQ
  + Riferimento al concetto: Usercontrol di tipo GetConcept che consente la selezione del concetto
  + Lista codificata: Usercontrol di tipo GetCodelist
* MEASURE
  + ID: Textbox editabile per l'inserimento di testo libero
  + Riferimento al concetto: Usercontrol di tipo GetConcept che consente la selezione del concetto
  + Concept Scheme: Usercontrol di tipo GetConcept che consente la selezione del ConceptScheme
* TIME
  + ID: Textbox non editabile contenente il valore TIME\_PERIOD
  + Riferimento al concetto: Usercontrol di tipo GetConcept che consente la selezione del concetto

##### UserControl utilizzati

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| ControlAnnotations | Gestisce il CRUD delle annotation |
| GetConcept | Permette la selezione di un artefatto di tipo ConceptScheme |

#### Gruppi

La sezione Gruppi visualizza tramite un controllo di tipo GridView la lista di tutti i Gruppi appartenenti alla DSD. Nella griglia è presente un imagebutton che consente l'eliminazione di un gruppo previa conferma e un altro imagebutton che apre una popup contenente uno UserControl di tipo ControlAnnotations. I dati relativi ai gruppi non possono essere modificati, per effettuare una modifica sarà necessario eliminare poi reinserire il gruppo.

Nella parte inferiore, sotto la gridview, è presente un imageButton che apre una popup contenente un form per l'inserimento di un nuovo gruppo.

Nel form di inserimento sono presenti i seguenti controlli:

* ID: TextBox utilizzata per l'inserimento dell'identificativo del gruppo
* Dimensioni: Questa dropdownlist a selezione mutlipla viene popolata con la lista delle Dimensioni.

Cliccando sul tasto aggiungi viene creato un oggetto di tipo Org.Sdmxsource.Sdmx.SdmxObjects.Model.Mutable.DataStructure.GroupMutableCore() che verrà poi aggiunto all'oggetto IDataStructureMutableObject che contiene la DSD. Se l'inserimento del gruppo nella DSD dovesse generare un errore verrà rimosso nel catch e verrà visualizzato un errore all'utente.

Esempio di codice:

IGroupMutableObject group = new GroupMutableCore();

try{

group.Id = txtGroupID.Text;

foreach (ListItem li in lbGroupDimension.Items)

{

if (li.Selected)

group.DimensionRef.Add(li.Value);

}

\_dsdMutable.AddGroup(group);

}  
catch (Exception ex)

{

\_dsdMutable.Groups.Remove(group);

Utils.ShowDialog(ex.Message, 600, Resources.Messages.err\_add\_group);

}

##### UserControl utilizzati

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| ControlAnnotations | Gestisce il CRUD delle annotation |

#### Attributi

La sezione Attributi visualizza tramite un controllo di tipo GridView la lista di tutti gli Attributi che fanno parte della DSD. Nella griglia è presente un imagebutton che consente l'eliminazione di un attributo previa conferma, un secondo per la modifica del gruppo, e l'ultimo che apre una popup contenente uno UserControl di tipo ControlAnnotations.

Nella parte inferiore, sotto la gridview, è presente un imageButton che apre una popup contenente un form per l'inserimento di un nuovo attributo.

Nel form di inserimento/modifica sono presenti i seguenti controlli:

* Attribute ID: TextBox utilizzata per l'inserimento dell'identificativo dell'attributo
* Concept: Usercontrol per la selezione di un concetto
* Codelist: Usercontrol per la selezione di una codelist
* Assignment Status: Dropdownlist popolata con i valori: 'Conditional' e 'Mandatory'
* Attachment level: Dropdownlist popolata con i valori: 'Dataset', 'DimensionGroup', 'Observation', 'Group'

Nell'event Handler 'OnChange' della ddl degli Attachement Level viene modificata l'interfaccia del form in base all'item scelto:

* DataSet
  + i controlli al di sotto della ddl non saranno visibil
* DimensionGroup
  + Viene visualizzato un controllo di tipo ListBox popolato con la lista delle Dimensioni
* Observation
  + i controlli al di sotto della ddl non saranno visibili
* Group:
  + Viene visualizzata la ddl popolata con gli identificativi dei gruppi

Cliccando sul tasto aggiungi viene creato un oggetto di tipo Org.Sdmxsource.Sdmx.SdmxObjects.Model.Mutable.DataStructure. AttributeMutableCore () che verrà poi aggiunto all'oggetto IDataStructureMutableObject che contiene la DSD. Se l'inserimento del nuovo attributo aggiunto alla DSD dovesse generare un errore verrà rimosso nel catch e verrà visualizzato un errore all'utente.

Esempio di codice:

IAttributeMutableObject attr = new AttributeMutableCore ();

.... codice per il popolamento dell'attributo

try{

\_dsdMutable.AddAttribute(attr);

}  
catch (Exception ex)

{

\_dsdMutable.Attributes.Remove(attr);

Utils.ShowDialog(ex.Message, 600, Resources.Messages. err\_add\_attribute);

}

##### UserControl utilizzati

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| ControlAnnotations | Gestisce il CRUD delle annotation |
| GetConcept | Permette la selezione di un artefatto di tipo ConceptScheme |
| GetCodelist | Permette la selezione di un artefatto di tipo Codelist |

### Concept scheme

La gestione dell'artefatto Concept Scheme si suddivide in 2 pagine .aspx: 'conceptschemes.aspx' e 'ConceptSchemeItemDetails.aspx'. La prima visualizza a video la lista dei Concept Scheme(in formato tabellare) e ne permette l'eliminazione(se l'utente è abilitato) e download. La seconda invece gestisce la visualizzazione del dettaglio e il CRUD dell'artefatto.

#### Conceptschemes.aspx

La pagina ConceptSchemes.aspx oltre alla visualizzazione dei conceptscheme permette il download degli artefatti. Se l'utente è abilitato, nella griglia sarà presente l'icona per la cancellazione e a fondo pagina un bottone per la creazione di un nuovo artefatto. Nell'Event Handler dell'evento click associato al bottone verrà effettuato un redirect alla pagina ConceptSchemeItemDetails.aspx, questa nuova pagina offrirà le funzionalità atte alla creazione di un nuovo artefatto della stessa tipologia. Il delete di un artefatto non è consentito se nel db è presente un vincolo di foreign key verso un'entità ad esso associata, in questo caso l'applicazione restituirà un errore.

##### Logica di esecuzione

Le operazioni che vengono svolte dal code behind della pagina per popolare la griglia sono le seguenti:

* Viene creata un'istanza dell'oggetto ISTAT.WSDAL.WSModel() (WSModel wsModel = new WSModel();)
* Viene richiamato il metodo GetConceptScheme() a cui viene passato un object entity di tipo ArtefactIdentity, questo oggetto contiene le proprietà 'ID','Agency' e 'Version' che se valorizzate filtreranno i risultati, in caso contrario verrà restituito l'intero resultset. I risultati saranno restituiti in un oggetto di tipo ISdmxObjects.
* Il risultato restituito verrà rimappato in una lista di oggetti di tipo ISTAT.Entity.ConceptScheme. Il mapping verrà eseguito dal metodo GetConceptSchemeList() della classe ISTAT.EntityMapper. EntityMapper() (List<ISTAT.Entity.ConceptScheme> lCS = eMapper.GetConceptSchemeList(\_sdmxObjects, Utils.LocalizedLanguage)
* Una volta ottenuta la lista di entità di tipo ConceptScheme()(List<ISTAT.Entity.ConceptScheme>) verrà effetuato il DataBind() della gridview.

##### Visualizzazione dettaglio

Cliccando su una delle iconcine del dettaglio presenti nella griglia, verrà effettuato un redirect alla pagina ConceptSchemeItemDetail.aspx, a cui verra passata in modalità 'GET' la chiave(id,agency e version) dell'artefatto richiesto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina Codelist.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| SearchBar | Utilizzato per filtrare la griglia dei risultati |
| ArtefactDelete | Permette di eliminare un artefatto previa conferma |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

#### ConceptSchemeItemDetails.aspx

Questa pagina offre funzionalità di gestione su artefatti di tipo ConceptScheme.

Funzionalità:

* CRUD Artefatto
* Duplicazione Artefatto
* Download Artefatto

##### CRUD ConceptScheme

Il CRUD del ConceptScheme è gestito in 2 TAB separati, il primo per la gestione delle informazioni principali dell'artefatto(ID, Agency,Version, Isfinal, ecc...), mentre il secondo di occupa della gestione degli Item code.

* Creazione nuovo ConceptScheme:
  + In fase di creazione di un nuovo ConceptScheme, i dati inseriti tramite interfaccia grafica(form) vengono memorizzati in un oggetto di tipo ISdmxObjects. L'oggetto una volta popolato viene inserito nel DB tramite la chiamata del metodo SubmitStructure() della libreria ISTAT.WSDAL.WSModel.  
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
     wsModel.SubmitStructure(sdmxObjects);
* Modifica conceptscheme esistente
  + In fase di modifica viene richiesto al Web service un artefatto di tipo ConceptScheme che poi verrà utilizzato per popolare i form presenti nella pagina. La chiave dell'artefatto(ID,Agency e Version) viene passata in modalità GET dalla pagina chiamante e riletta tramite Request dalla pagina ricevente.   
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
    ArtefactIdentity artIdentity = Utils.GetIdentityFromRequest(Request);  
    ISdmxObjects sdmxObject = wsModel.GetConceptScheme(artIdentity);
  + Apportate le modifiche, verrà richiamato il metodo SubmitStructure()(analogamente alla creazione del ConceptScheme) che si occuperà dell'aggiornamento dell'Artefatto nel DB.

##### Duplicazione artefatto

Se l'utente è abilitato sarà visibile lo User Control DuplicateArtefact che esporrà il bottone 'Duplicazione Artefatto'. Questo controllo deve essere inizializzato con la Chiave SDMX dell'artefatto.

##### Download Artefatto

Se l'utente è abilitato sarà visibile lo User Control FileDownload che visualizzerà a video un ImageButton che permetterà il download dell'artefatto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina CodelistItemDetails.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| DuplicateArtefact | Permette la duplicazione di un artefatto |
| ControlAnnotations | Gestisce il CRUD delle annotation sia a livello di artefatto che di Item |
| AddText | Gestisce il CRUD sui text(Name,Description) dell'artefatto e dell'Item |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

### Category scheme

### Dataflow

La gestione dell'artefatto Dataflow si suddivide in 2 pagine .aspx: 'dataflow.aspx' e 'dataflowItemDetails.aspx'. La prima visualizza a video la lista dei dataflow(in formato tabellare) e ne permette l'eliminazione(se l'utente è abilitato) e download. La seconda invece gestisce la visualizzazione del dettaglio e il CRUD dell'artefatto.

#### Dataflow.aspx

La pagina dataflow.aspx oltre alla visualizzazione dei dataflow permette il download degli artefatti. Se l'utente è abilitato, nella griglia sarà presente l'icona per la cancellazione e a fondo pagina un bottone per la creazione di un nuovo artefatto. Nell'Event Handler dell'evento click associato al bottone verrà effettuato un redirect alla pagina dataflowItemDetail.aspx che esporrà funzionalità di CRUD. Il delete di un artefatto non è consentito se nel db è presente un vincolo di foreign key verso un'entità ad esso associata, in questo caso l'applicazione restituirà un errore.

##### Logica di esecuzione

Le operazioni che vengono svolte dal code behind della pagina per popolare la griglia sono le seguenti:

* Viene creata un'istanza dell'oggetto ISTAT.WSDAL.WSModel() (WSModel wsModel = new WSModel();)
* Viene richiamato il metodo GetDataflow() a cui viene passato un object entity di tipo ArtefactIdentity, questo oggetto contiene la chiave dell'artefatto formata da 'ID','Agency' e 'Version' che se valorizzate filtreranno i risultati, in caso contrario verrà restituito l'intero resultset. I risultati saranno restituiti in un oggetto di tipo ISdmxObjects.
* Il risultato restituito viene rimappato in una lista di oggetti di tipo ISTAT.Entity.Dataflow. Il mapping verrà eseguito dal metodo GetDataflowList() della classe ISTAT.EntityMapper. EntityMapper() (List<ISTAT.Entity.Dataflow> lDataflow = eMapper.GetDataflowList(\_sdmxObjects, Utils.LocalizedLanguage)
* Una volta ottenuta la lista di entità di tipo Dataflow()(List<ISTAT.Entity.Dataflow>) verrà effetuato il DataBind() della gridview.

##### Visualizzazione dettaglio

Cliccando su una delle iconcine del dettaglio presenti nella griglia, verrà effettuato un redirect alla pagina DataflowItemDetail.aspx, a cui verra passata in modalità 'GET' la chiave(id,agency e version) dell'artefatto richiesto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina Dataflow.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| SearchBar | Utilizzato per filtrare la griglia dei risultati |
| ArtefactDelete | Permette di eliminare un artefatto previa conferma |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

#### DataflowItemDetails.aspx

Questa pagina offre funzionalità di gestione CRUD su artefatti di tipo ConceptScheme.

Funzionalità:

* CRUD Artefatto
* Duplicazione Artefatto
* Download Artefatto

##### CRUD ConceptScheme

Il CRUD del ConceptScheme è gestito in un unico form che contiene i dati principali dell'artefatto (ID, Agency,Version, Isfinal, ecc...) e lo UserControl GetDSD.

* Creazione nuovo Dataflow:
  + In fase di creazione di un nuovo Dataflow, i dati inseriti tramite interfaccia grafica(form) vengono memorizzati in un oggetto di tipo ISdmxObjects. L'oggetto una volta popolato viene inserito nel DB tramite la chiamata del metodo SubmitStructure() della libreria ISTAT.WSDAL.WSModel.  
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
     wsModel.SubmitStructure(sdmxObjects);
* Modifica Dataflow esistente
  + In fase di modifica viene richiesto al Web service un artefatto di tipo Dataflow che poi verrà utilizzato per popolare i form presenti nella pagina. La chiave dell'artefatto(ID,Agency e Version) viene passata in modalità GET dalla pagina chiamante e riletta tramite Request dalla pagina ricevente.   
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
    ArtefactIdentity artIdentity = Utils.GetIdentityFromRequest(Request);  
    ISdmxObjects sdmxObject = wsModel.GetDataflow(artIdentity);
  + Apportate le modifiche, verrà richiamato il metodo SubmitStructure()(analogamente alla creazione del Dataflow) che si occuperà dell'aggiornamento dell'Artefatto nel DB.

##### Duplicazione artefatto

Se l'utente è abilitato sarà visibile lo User Control DuplicateArtefact che esporrà il bottone 'Duplicazione Artefatto'. Questo controllo deve essere inizializzato con la Chiave SDMX dell'artefatto.

##### Download Artefatto

Se l'utente è abilitato sarà visibile lo User Control FileDownload che visualizzerà a video un ImageButton che permetterà il download dell'artefatto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina CodelistItemDetails.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| DuplicateArtefact | Permette la duplicazione di un artefatto |
| ControlAnnotations | Gestisce il CRUD delle annotation sia a livello di artefatto che di Item |
| AddText | Gestisce il CRUD sui text(Name,Description) dell'artefatto e dell'Item |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |
| GetDSD | Permette la selezione di un artefatto di tipo DSD |

### Categorisation

### Hierarchical Codelist

Per l’artefatto Hierarchical Codelist al momento non è prevista la funzionalità di CRUD ma solo la visualizzazione delle informazioni principali.Anche per le HCL la visualizzazione si suddivide in 2 pagine .aspx: 'hcl.aspx' e 'hclItemDetails.aspx'. La prima visualizza a video la lista delle HCL(in formato tabellare) e ne permette l'eliminazione(se l'utente è abilitato) e download. La seconda invece visualizza le informazioni principali dell'artefatto.

#### Hcl.aspx

La pagina hcl.aspx oltre alla visualizzazione dei conceptscheme permette il download degli artefatti. Se l'utente è abilitato, nella griglia sarà presente l'icona per la cancellazione. Nell'Event Handler dell'evento click associato al bottone verrà effettuato un redirect alla pagina hclItemDetails.aspx, la pagina appena aperta visualizzerà a video le informazioni di base dell’artefatto selezionato.

### Agency scheme

La gestione dell'artefatto Agency Scheme si suddivide in 2 pagine .aspx: 'AgencySchemes.aspx' e 'AgencySchemeItemDetails.aspx'. La prima visualizza a video la lista degli Agency Scheme(in formato tabellare) e ne permette l'eliminazione(se l'utente è abilitato) e download. La seconda invece gestisce la visualizzazione del dettaglio e il CRUD dell'artefatto.

#### AgencySchemes.aspx

La pagina AgencySchemes.aspx oltre alla visualizzazione degli AgencyScheme permette il download degli artefatti. Se l'utente è abilitato, nella griglia sarà presente l'icona per la cancellazione e a fondo pagina un bottone per la creazione di un nuovo artefatto. Nell'Event Handler dell'evento click associato al bottone verrà effettuato un redirect alla pagina AgencySchemeItemDetails.aspx, questa nuova pagina offrirà le funzionalità atte alla creazione di un nuovo artefatto della stessa tipologia. Il delete di un artefatto non è consentito se nel db è presente un vincolo di foreign key verso un'entità ad esso associata, in questo caso l'applicazione restituirà un errore.

##### Logica di esecuzione

Le operazioni che vengono svolte dal code behind della pagina per popolare la griglia sono le seguenti:

* Viene creata un'istanza dell'oggetto ISTAT.WSDAL.WSModel() (WSModel wsModel = new WSModel();)
* Viene richiamato il metodo GetAgencyScheme () a cui viene passato un object entity di tipo ArtefactIdentity, questo oggetto contiene le proprietà 'ID','Agency' e 'Version' che se valorizzate filtreranno i risultati, in caso contrario verrà restituito l'intero resultset. I risultati saranno restituiti in un oggetto di tipo ISdmxObjects.
* Il risultato restituito verrà rimappato in una lista di oggetti di tipo ISTAT.Entity. AgencyScheme. Il mapping verrà eseguito dal metodo GetAgencySchemeList() della classe ISTAT.EntityMapper. EntityMapper() (List<ISTAT.Entity.AgencyScheme> lCS = eMapper.GetAgencySchemeList(\_sdmxObjects, Utils.LocalizedLanguage)
* Una volta ottenuta la lista di entità di tipo AgencyScheme ()(List<ISTAT.Entity.AgencyScheme>) verrà effetuato il DataBind() della gridview.

##### Visualizzazione dettaglio

Cliccando su una delle iconcine del dettaglio presenti nella griglia, verrà effettuato un redirect alla pagina AgencySchemeItemDetail.aspx, a cui verra passata in modalità 'GET' la chiave(id,agency e version) dell'artefatto richiesto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina AgencyScheme.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| SearchBar | Utilizzato per filtrare la griglia dei risultati |
| ArtefactDelete | Permette di eliminare un artefatto previa conferma |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

#### AgencySchemeItemDetails.aspx

Questa pagina offre funzionalità di gestione su artefatti di tipo AgencyScheme.

Funzionalità:

* CRUD Artefatto
* Duplicazione Artefatto
* Download Artefatto

##### CRUD AgencyScheme

Il CRUD dell' AgencyScheme è gestito in 2 TAB separati, il primo per la gestione delle informazioni principali dell'artefatto(ID, Agency,Version, Isfinal, ecc...), mentre il secondo di occupa della gestione degli Item code.   
Da notare che L'agencyScheme non è un artefatto finalizzabile.

* Creazione nuovo AgencyScheme:
  + In fase di creazione di un nuovo AgencyScheme, i dati inseriti tramite interfaccia grafica(form) vengono memorizzati in un oggetto di tipo ISdmxObjects. L'oggetto una volta popolato viene inserito nel DB tramite la chiamata del metodo SubmitStructure() della libreria ISTAT.WSDAL.WSModel.  
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
     wsModel.SubmitStructure(sdmxObjects);
* Modifica AgencyScheme esistente
  + In fase di modifica viene richiesto al Web service un artefatto di tipo AgencyScheme che poi verrà utilizzato per popolare i form presenti nella pagina. La chiave dell'artefatto(ID,Agency e Version) viene passata in modalità GET dalla pagina chiamante e riletta tramite Request dalla pagina ricevente.   
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
    ArtefactIdentity artIdentity = Utils.GetIdentityFromRequest(Request);  
    ISdmxObjects sdmxObject = wsModel.GetAgencyScheme (artIdentity);
  + Apportate le modifiche, verrà richiamato il metodo SubmitStructure()(analogamente alla creazione dell'AgencyScheme) che si occuperà dell'aggiornamento dell'Artefatto nel DB.

##### Download Artefatto

Se l'utente è abilitato sarà visibile lo User Control FileDownload che visualizzerà a video un ImageButton che permetterà il download dell'artefatto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina AgencySchemeItemDetails.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| DuplicateArtefact | Permette la duplicazione di un artefatto |
| ControlAnnotations | Gestisce il CRUD delle annotation sia a livello di artefatto che di Item |
| AddText | Gestisce il CRUD sui text(Name,Description) dell'artefatto e dell'Item |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

### Data provider scheme

La gestione dell'artefatto DataProvider Scheme si suddivide in 2 pagine .aspx: ' DataProvider Schemes.aspx' e ' DataProvider ItemDetails.aspx'. La prima visualizza a video la lista dei DataProviderScheme (in formato tabellare) e ne permette l'eliminazione(se l'utente è abilitato) e download. La seconda invece gestisce la visualizzazione del dettaglio e il CRUD dell'artefatto.

#### DataProviderSchemes.aspx

La pagina DataProviderScheme.aspx oltre alla visualizzazione dei DataProviderScheme permette il download degli artefatti. Se l'utente è abilitato, nella griglia sarà presente l'icona per la cancellazione e a fondo pagina un bottone per la creazione di un nuovo artefatto. Nell'Event Handler dell'evento click associato al bottone verrà effettuato un redirect alla pagina DataProviderSchemeItemDetails.aspx, questa nuova pagina offrirà le funzionalità atte alla creazione di un nuovo artefatto della stessa tipologia. Il delete di un artefatto non è consentito se nel db è presente un vincolo di foreign key verso un'entità ad esso associata, in questo caso l'applicazione restituirà un errore.

##### Logica di esecuzione

Le operazioni che vengono svolte dal code behind della pagina per popolare la griglia sono le seguenti:

* Viene creata un'istanza dell'oggetto ISTAT.WSDAL.WSModel() (WSModel wsModel = new WSModel();)
* Viene richiamato il metodo Get DataProviderScheme() a cui viene passato un object entity di tipo ArtefactIdentity, questo oggetto contiene le proprietà 'ID','Agency' e 'Version' che se valorizzate filtreranno i risultati, in caso contrario verrà restituito l'intero resultset. I risultati saranno restituiti in un oggetto di tipo ISdmxObjects.
* Il risultato restituito verrà rimappato in una lista di oggetti di tipo ISTAT.Entity. DataProviderScheme. Il mapping verrà eseguito dal metodo Get DataProviderScheme List() della classe ISTAT.EntityMapper. EntityMapper() (List<ISTAT.Entity.DataProviderScheme> lCS = eMapper.Get DataProviderSchemeList(\_sdmxObjects, Utils.LocalizedLanguage)
* Una volta ottenuta la lista di entità di tipo DataProviderScheme() (List<ISTAT.Entity. DataProviderScheme >) verrà effetuato il DataBind() della gridview.

##### Visualizzazione dettaglio

Cliccando su una delle iconcine del dettaglio presenti nella griglia, verrà effettuato un redirect alla pagina DataProviderSchemeItemDetail.aspx, a cui verra passata in modalità 'GET' la chiave(id,agency e version) dell'artefatto richiesto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina DataProviderScheme.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| SearchBar | Utilizzato per filtrare la griglia dei risultati |
| ArtefactDelete | Permette di eliminare un artefatto previa conferma |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

#### DataProviderSchemeItemDetails.aspx

Questa pagina offre funzionalità di gestione su artefatti di tipo DataProviderScheme.

Funzionalità:

* CRUD Artefatto
* Duplicazione Artefatto
* Download Artefatto

##### CRUD DataProviderScheme

Il CRUD del DataProviderScheme è gestito in 2 TAB separati, il primo per la gestione delle informazioni principali dell'artefatto(ID, Agency,Version, Isfinal, ecc...), mentre il secondo di occupa della gestione degli Item code.   
Da notare che il DataProviderScheme non è un artefatto finalizzabile.

* Creazione nuovo DataProviderScheme:
  + In fase di creazione di un nuovo DataProviderScheme, i dati inseriti tramite interfaccia grafica(form) vengono memorizzati in un oggetto di tipo ISdmxObjects. L'oggetto una volta popolato viene inserito nel DB tramite la chiamata del metodo SubmitStructure() della libreria ISTAT.WSDAL.WSModel.  
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
     wsModel.SubmitStructure(sdmxObjects);
* Modifica DataProviderScheme esistente
  + In fase di modifica viene richiesto al Web service un artefatto di tipo DataProviderScheme che poi verrà utilizzato per popolare i form presenti nella pagina. La chiave dell'artefatto(ID,Agency e Version) viene passata in modalità GET dalla pagina chiamante e riletta tramite Request dalla pagina ricevente.   
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
    ArtefactIdentity artIdentity = Utils.GetIdentityFromRequest(Request);  
    ISdmxObjects sdmxObject = wsModel. DataProviderScheme(artIdentity);
  + Apportate le modifiche, verrà richiamato il metodo SubmitStructure()(analogamente alla creazione del DataProviderScheme) che si occuperà dell'aggiornamento dell'Artefatto nel DB.

##### Download Artefatto

Se l'utente è abilitato sarà visibile lo User Control FileDownload che visualizzerà a video un ImageButton che permetterà il download dell'artefatto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina DataProviderSchemeItemDetails.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| ControlAnnotations | Gestisce il CRUD delle annotation sia a livello di artefatto che di Item |
| AddText | Gestisce il CRUD sui text(Name,Description) dell'artefatto e dell'Item |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

### Data Consumer scheme

La gestione dell'artefatto DataConsumerScheme si suddivide in 2 pagine .aspx: ' DataConsumerScheme.aspx' e ' DataConsumerSchemeItemDetails.aspx'. La prima visualizza a video la lista dei DataConsumerScheme (in formato tabellare) e ne permette l'eliminazione(se l'utente è abilitato) e download. La seconda invece gestisce la visualizzazione del dettaglio e il CRUD dell'artefatto.

#### DataConsumerSchemes.aspx

La pagina DataConsumerScheme.aspx oltre alla visualizzazione dei DataConsumerScheme permette il download degli artefatti. Se l'utente è abilitato, nella griglia sarà presente l'icona per la cancellazione e a fondo pagina un bottone per la creazione di un nuovo artefatto. Nell'Event Handler dell'evento click associato al bottone verrà effettuato un redirect alla pagina DataConsumerSchemeItemDetails.aspx, questa nuova pagina offrirà le funzionalità atte alla creazione di un nuovo artefatto della stessa tipologia. Il delete di un artefatto non è consentito se nel db è presente un vincolo di foreign key verso un'entità ad esso associata, in questo caso l'applicazione restituirà un errore.

##### Logica di esecuzione

Le operazioni che vengono svolte dal code behind della pagina per popolare la griglia sono le seguenti:

* Viene creata un'istanza dell'oggetto ISTAT.WSDAL.WSModel() (WSModel wsModel = new WSModel();)
* Viene richiamato il metodo GetDataConsumerScheme() a cui viene passato un object entity di tipo ArtefactIdentity, questo oggetto contiene le proprietà 'ID','Agency' e 'Version' che se valorizzate filtreranno i risultati, in caso contrario verrà restituito l'intero resultset. I risultati saranno restituiti in un oggetto di tipo ISdmxObjects.
* Il risultato restituito verrà rimappato in una lista di oggetti di tipo ISTAT.Entity. DataConsumerScheme. Il mapping verrà eseguito dal metodo GetDataConsumerSchemeList() della classe ISTAT.EntityMapper. EntityMapper() (List<ISTAT.Entity.DataConsumerScheme> lCS = eMapper.Get DataConsumerSchemeList(\_sdmxObjects, Utils.LocalizedLanguage)
* Una volta ottenuta la lista di entità di tipo DataConsumerScheme () (List<ISTAT.Entity.DataConsumerScheme>) verrà effetuato il DataBind() della gridview.

##### Visualizzazione dettaglio

Cliccando su una delle iconcine del dettaglio presenti nella griglia, verrà effettuato un redirect alla pagina DataConsumerSchemeItemDetail.aspx, a cui verra passata in modalità 'GET' la chiave(id,agency e version) dell'artefatto richiesto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina DataConsumerScheme.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| SearchBar | Utilizzato per filtrare la griglia dei risultati |
| ArtefactDelete | Permette di eliminare un artefatto previa conferma |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

#### DataConsumerSchemeItemDetails.aspx

Questa pagina offre funzionalità di gestione su artefatti di tipo DataConsumerScheme.

Funzionalità:

* CRUD Artefatto
* Duplicazione Artefatto
* Download Artefatto

##### CRUD DataConsumerScheme

Il CRUD del DataConsumerSchemeè gestito in 2 TAB separati, il primo per la gestione delle informazioni principali dell'artefatto(ID, Agency,Version, Isfinal, ecc...), mentre il secondo di occupa della gestione degli Item code.   
Da notare che il DataConsumerScheme non è un artefatto finalizzabile.

* Creazione nuovo DataConsumerScheme:
  + In fase di creazione di un nuovo DataConsumerScheme, i dati inseriti tramite interfaccia grafica(form) vengono memorizzati in un oggetto di tipo ISdmxObjects. L'oggetto una volta popolato viene inserito nel DB tramite la chiamata del metodo SubmitStructure() della libreria ISTAT.WSDAL.WSModel.  
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
     wsModel.SubmitStructure(sdmxObjects);
* Modifica DataConsumerScheme esistente
  + In fase di modifica viene richiesto al Web service un artefatto di tipo DataConsumerScheme che poi verrà utilizzato per popolare i form presenti nella pagina. La chiave dell'artefatto(ID,Agency e Version) viene passata in modalità GET dalla pagina chiamante e riletta tramite Request dalla pagina ricevente.   
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
    ArtefactIdentity artIdentity = Utils.GetIdentityFromRequest(Request);  
    ISdmxObjects sdmxObject = wsModel. DataConsumerScheme(artIdentity);
  + Apportate le modifiche, verrà richiamato il metodo SubmitStructure()(analogamente alla creazione del DataConsumerScheme) che si occuperà dell'aggiornamento dell'Artefatto nel DB.

##### Download Artefatto

Se l'utente è abilitato sarà visibile lo User Control FileDownload che visualizzerà a video un ImageButton che permetterà il download dell'artefatto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina DataProviderSchemeItemDetails.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| ControlAnnotations | Gestisce il CRUD delle annotation sia a livello di artefatto che di Item |
| AddText | Gestisce il CRUD sui text(Name,Description) dell'artefatto e dell'Item |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

### Organization unit scheme

La gestione dell'artefatto OrganizationUnitScheme si suddivide in 2 pagine .aspx: ' OrganizationUnitScheme.aspx' e ' OrganizationUnitScheme SchemeItemDetails.aspx'. La prima visualizza a video la lista degli OrganizationUnitScheme (in formato tabellare) e ne permette l'eliminazione(se l'utente è abilitato) e download. La seconda invece gestisce la visualizzazione del dettaglio e il CRUD dell'artefatto.

#### OrganizationUnitSchemes.aspx

La pagina OrganizationUnitScheme.aspx oltre alla visualizzazione degli OrganizationUnitScheme permette il download degli artefatti. Se l'utente è abilitato, nella griglia sarà presente l'icona per la cancellazione e a fondo pagina un bottone per la creazione di un nuovo artefatto. Nell'Event Handler dell'evento click associato al bottone verrà effettuato un redirect alla pagina OrganizationUnitSchemeItemDetails.aspx, questa nuova pagina offrirà le funzionalità atte alla creazione di un nuovo artefatto della stessa tipologia. Il delete di un artefatto non è consentito se nel db è presente un vincolo di foreign key verso un'entità ad esso associata, in questo caso l'applicazione restituirà un errore.

##### Logica di esecuzione

Le operazioni che vengono svolte dal code behind della pagina per popolare la griglia sono le seguenti:

* Viene creata un'istanza dell'oggetto ISTAT.WSDAL.WSModel() (WSModel wsModel = new WSModel();)
* Viene richiamato il metodo GetOrganizationUnitScheme() a cui viene passato un object entity di tipo ArtefactIdentity, questo oggetto contiene le proprietà 'ID','Agency' e 'Version' che se valorizzate filtreranno i risultati, in caso contrario verrà restituito l'intero resultset. I risultati saranno restituiti in un oggetto di tipo ISdmxObjects.
* Il risultato restituito verrà rimappato in una lista di oggetti di tipo ISTAT.Entity. OrganizationUnitScheme. Il mapping verrà eseguito dal metodo GetOrganizationUnitSchemeList() della classe ISTAT.EntityMapper. EntityMapper() (List<ISTAT.Entity.OrganizationUnitScheme> lCS = eMapper.Get OrganizationUnitSchemeList(\_sdmxObjects, Utils.LocalizedLanguage)
* Una volta ottenuta la lista di entità di tipo OrganizationUnitScheme() (List<ISTAT.Entity.OrganizationUnitScheme >) verrà effetuato il DataBind() della gridview.

##### Visualizzazione dettaglio

Cliccando su una delle iconcine del dettaglio presenti nella griglia, verrà effettuato un redirect alla pagina OrganizationUnitSchemeItemDetail.aspx, a cui verra passata in modalità 'GET' la chiave(id,agency e version) dell'artefatto richiesto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina OrganizationUnitScheme.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| SearchBar | Utilizzato per filtrare la griglia dei risultati |
| ArtefactDelete | Permette di eliminare un artefatto previa conferma |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

#### OrganizationUnitSchemeItemDetails.aspx

Questa pagina offre funzionalità di gestione su artefatti di tipo OrganizationUnitScheme.

Funzionalità:

* CRUD Artefatto
* Duplicazione Artefatto
* Download Artefatto

##### CRUD OrganizationUnitScheme

Il CRUD dell' OrganizationUnitScheme è gestito in 2 TAB separati, il primo per la gestione delle informazioni principali dell'artefatto(ID, Agency,Version, Isfinal, ecc...), mentre il secondo di occupa della gestione degli Item code.   
Da notare che il OrganizationUnitScheme non è un artefatto finalizzabile.

* Creazione nuovo OrganizationUnitScheme:
  + In fase di creazione di un nuovo OrganizationUnitScheme, i dati inseriti tramite interfaccia grafica(form) vengono memorizzati in un oggetto di tipo ISdmxObjects. L'oggetto una volta popolato viene inserito nel DB tramite la chiamata del metodo SubmitStructure() della libreria ISTAT.WSDAL.WSModel.  
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
     wsModel.SubmitStructure(sdmxObjects);
* Modifica OrganizationUnitScheme esistente
  + In fase di modifica viene richiesto al Web service un artefatto di tipo OrganizationUnitScheme che poi verrà utilizzato per popolare i form presenti nella pagina. La chiave dell'artefatto(ID,Agency e Version) viene passata in modalità GET dalla pagina chiamante e riletta tramite Request dalla pagina ricevente.   
    Esempio di codice:   
    WSModel wsModel = new WSModel();  
    ArtefactIdentity artIdentity = Utils.GetIdentityFromRequest(Request);  
    ISdmxObjects sdmxObject = wsModel. OrganizationUnitScheme(artIdentity);
  + Apportate le modifiche, verrà richiamato il metodo SubmitStructure()(analogamente alla creazione del OrganizationUnitScheme) che si occuperà dell'aggiornamento dell'Artefatto nel DB.

##### Download Artefatto

Se l'utente è abilitato sarà visibile lo User Control FileDownload che visualizzerà a video un ImageButton che permetterà il download dell'artefatto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina OrganizationUnitSchemeItemDetails.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| ControlAnnotations | Gestisce il CRUD delle annotation sia a livello di artefatto che di Item |
| AddText | Gestisce il CRUD sui text(Name,Description) dell'artefatto e dell'Item |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

### Content constraint

Tutte le informazioni che caratterizzano le Content Constraint vengono memorizzate in un oggetto di tipo ISdmxObjects. L'oggetto una volta popolato viene inserito nel DB tramite la chiamata al metodo SubmitStructure() della libreria ISTAT.WSDAL.WSModel. L'artefatto Content Constraint viene gestito tramite la pagina ContentConstraintItemDetails.aspx.

Esempio di codice:   
WSModel wsModel = new WSModel();  
wsModel.SubmitStructure(sdmxObjects);

La gestione dei Content Constraint si suddivide in 3 aree:

* Generale
  + Contiene le informazioni principali dell'artefatto
* Elementi Constraint
  + Form per la gestione dei constraint
* Release Calendar
  + Form per la gestione dei Release Calnedar

#### Generale

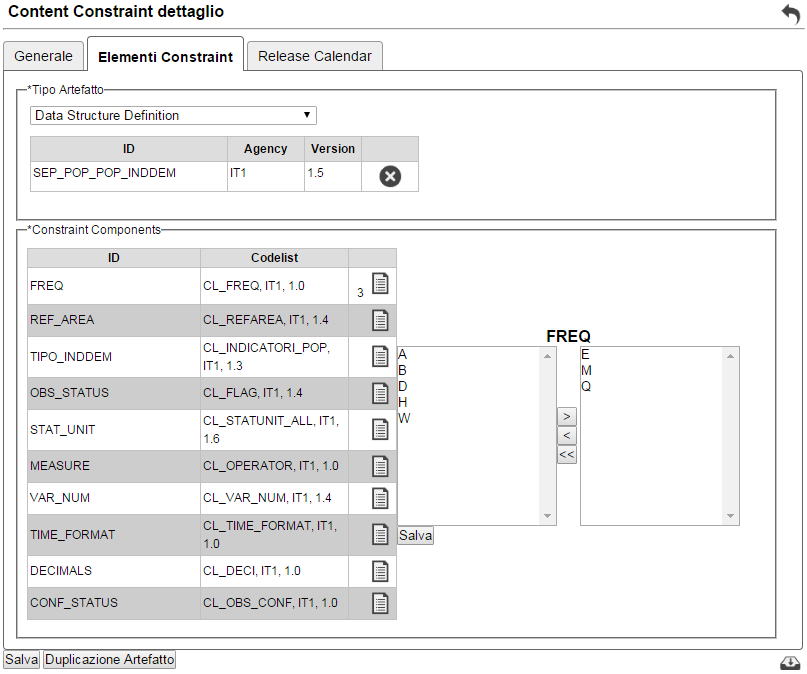
La sezione Generale si occupa della gestione dei dati principali dell'artefatto.  
Nel dettaglio:

* ID: Identificativo dell'artefatto
* Agency: Identificativo dell'agenzia
* Version: Versione dell'artefatto
* URI: Uri dell'artefatto
* URN:Urn dell'artefatto
* Finalizzata: Checkbox che identifica se l'artefatto è finalizzato
* Valido da: Textbox associato ad un controllo di tipo Calendar che contiene la data inizio validità dell'artefatto
* Valido a: Textbox associato ad un controllo di tipo Calendar che contiene la data di fine validità dell'artefatto
* Nome: Usercontrol per la gestione dei Name
* Descrizione: Usercontrol per la gestione delle descrizioni
* Annotazioni: Usercontrol per la gestione delle Annotazioni

##### UserControl utilizzati

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| DuplicateArtefact | Permette la duplicazione di un artefatto |
| ControlAnnotations | Gestisce il CRUD delle annotation |
| AddText | Gestisce il CRUD dei text(Name,Description) |
| FileDownload | Gestisce il download dell'artefatto |

#### Elementi Constraint

Questa sezione permette di selezionare dei vincoli in artefatti di tipo DataStructure e Dataflow.  
Tramite una dropdownlist è possibile selezionare uno di questi 2 artefatti, a seconda della selezione viene reso visibile lo UserControl GetDSD o GetDataflow. Selezionato l'artefatto, nel riquadro inferiore viene popolata una Gridview con la lista delle Codelist(Dimensioni e attributi) ad esso associate. Per ogni codelist sarà possibile includere degli item tramite 2 listbox a selezione multipla.  
  


##### Logica di esecuzione

* Nel'Event handler associato al SelectedIndexChanged della DropDownlist viene reso visibile lo usercontrol per la selezione dell'artefatto e successivamente viene popolata la Gridview delle codelist.
* Cliccando sull'iconcina del dettaglio della codelist scatterà l'evento 'RowCommand' della gridview, nel suo event handler verrà richiamato il Metodo 'GetCodeList' della libreria 'WSModel' a cui verrà passato in input l' 'ArtefactIdentity' della Codelist selezionata.
* Ogni codice della codelist verrà inserito nella Listbox di sinistra contenente gli Item associabili. Ogni riga della gridview contiene un controllo non visibile di tipo 'Label' che conserva informazioni sulla tipologia di componente: Dimension o Attribute. A seconda della tipologia di componente viene richiamato un metodo di classe. Per i componenti di tipo 'Dimension' viene richiamato il metodo 'MoveSelectedDimensions', per il tipo 'Attribute' invece 'MoveSelectedAttributes'. Questi 2 metodi svolgono la medesima operazione, ovvero, quella di popolare la listbox di destra con gli item vincolati. L'ultima operazione gestita dal metodo 'DeleteSelectedComponent()' provvede ad eliminare gli item della lista di sinistra(associabili) presenti nella lista di destra(associati).

##### UserControl utilizzati

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| GetDSD | Permette la selezione di un artefatto di tipo DSD |
| GetDataflow | Permette la selezione di un artefatto di tipo Dataflow |

#### Release Calendar

Questa sezione permette di gestire i dati relativi al Release Calendar.  
I valori inseriti servono a popolare l'oggetto delle Common API 'IContentConstraintMutableObjcet.ReleaseCalendar'   
Il Form contiene:

* 3 Textbox per l'inserimento di: Periodicità, Scostamento e Tolleranza, questi dati sono di tipo Integer e obbligatori.
* 3 Dropdownlist, una per ciascun textbox, popolate con la periodicità: ''Days,'Week', 'Month', 'Year

## StructureSet

La gestione dell'artefatto StructureSet si suddivide in 2 pagine .aspx: 'structureset.aspx' e 'StructureSetItemDetails.aspx'. La prima visualizza a video la lista degli StructureSet(in formato tabellare) e ne permette l'eliminazione(se l'utente è abilitato) e download. La seconda invece gestisce la visualizzazione del dettaglio e il CRUD dell'artefatto.

#### StructureSet.aspx

La pagina structureSet.aspx oltre alla visualizzazione degli StructureSet permette il download degli artefatti. Se l'utente è abilitato, nella griglia sarà presente l'icona per la cancellazione e a fondo pagina un bottone per la creazione di un nuovo artefatto. Nell'Event Handler dell'evento click associato al bottone verrà effettuato un redirect alla pagina StructureSetItemDetail.aspx che esporrà funzionalità di CRUD. Il delete di un artefatto non è consentito se nel db è presente un vincolo di foreign key verso un'entità ad esso associata, in questo caso l'applicazione restituirà un errore.

##### Logica di esecuzione

Le operazioni che vengono svolte dal code behind della pagina per popolare la griglia sono le seguenti:

* Viene creata un'istanza dell'oggetto ISTAT.WSDAL.WSModel() (WSModel wsModel = new WSModel();)
* Viene richiamato il metodo GetStructureSet() a cui viene passato un object entity di tipo ArtefactIdentity, questo oggetto contiene la chiave dell'artefatto formata da 'ID','Agency' e 'Version' che se valorizzate filtreranno i risultati, in caso contrario verrà restituito l'intero resultset. I risultati saranno restituiti in un oggetto di tipo ISdmxObjects.
* Il risultato restituito viene rimappato in una lista di oggetti di tipo ISTAT.Entity.StructureSet. Il mapping verrà eseguito dal metodo GetStructureSetList() della classe ISTAT.EntityMapper. EntityMapper() (List<ISTAT.Entity.StructureSet> lStructureSet = eMapper.GetStructureSetList(\_sdmxObjects, Utils.LocalizedLanguage)
* Una volta ottenuta la lista di entità di tipo StructureSet()(List<ISTAT.Entity.StructureSet>) verrà effetuato il DataBind() della gridview.

##### Visualizzazione dettaglio

Cliccando su una delle iconcine del dettaglio presenti nella griglia, verrà effettuato un redirect alla pagina StructureSetItemDetail.aspx, a cui verra passata in modalità 'GET' la chiave(id,agency e version) dell'artefatto richiesto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina StructureSet.aspx

|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| SearchBar | Utilizzato per filtrare la griglia dei risultati |
| ArtefactDelete | Permette di eliminare un artefatto previa conferma |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 * CSV |

#### StructureSetItemDetails.aspx

Questa pagina offre funzionalità di gestione CRUD su artefatti di tipo StructureSet.

Funzionalità:

* CRUD Artefatto
* Duplicazione Artefatto
* Download Artefatto

##### CRUD StructureSet

Il CRUD dello StructureSet suddivide in tre Tab:

* Generale
  + In questa sezione è possibile la gestione CRUD delle informazioni di base dell’Artefatto
* CodeListMap
  + In questa sezione è possibile la gestione CRUD della CodeListMap.
* StructureMap
  + Gestisce in CRUD dello StructureMap

##### Duplicazione artefatto

Se l'utente è abilitato sarà visibile lo User Control DuplicateArtefact che esporrà il bottone 'Duplicazione Artefatto'. Questo controllo deve essere inizializzato con la Chiave SDMX dell'artefatto.

##### Download Artefatto

Se l'utente è abilitato sarà visibile lo User Control FileDownload che visualizzerà a video un ImageButton che permetterà il download dell'artefatto.

##### UserControl utilizzati

Di seguito una tabella riassuntiva degli User Control utilizzati nella pagina CodelistItemDetails.aspx

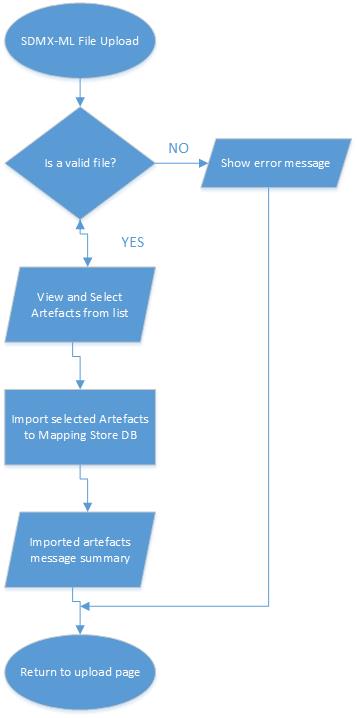
|  |  |
| --- | --- |
| ***Nome*** | ***Funzionalità*** |
| DuplicateArtefact | Permette la duplicazione di un artefatto |
| ControlAnnotations | Gestisce il CRUD delle annotation sia a livello di artefatto che di Item |
| AddText | Gestisce il CRUD sui text(Name,Description) dell'artefatto e dell'Item |
| FileDownload | Permette di scaricare un artefatto nei seguenti formati:   * SDMX 2.0 * SDMX 2.1 |
| GetDSD | Permette la selezione di un artefatto di tipo DSD |
| GetDataflow | Permette la selezione di un artefatto di tipo Dataflow |
| GetCodelist | Permette la selezione di un artefatto di tipo Codelist |

## Importazione artefatti

La funzionalità di importazione artefatti, consente l'inserimento o sostituzione di artefatti nel Mapping Store DB.  
Il codice relativo è presente nella pagina: 'UploadStructure.aspx'

### Logica funzionale

La funzionalità è accessbile attraverso la voce 'Importa Structure' del menu di navigazione. Se l'utente è abilitato, la pagina visualizzerà un form contenente un controllo asp.net di tipo FileUpload, e un pulsante per avviare l'upload del file verso il server. La tipologià di file caricabile deve essere di tipo SDMX-ML Compliant. Effettuato l'upload del file verrà visualizzata una finestra che mostra a video in formato tabellare tutti gli artefatti contenuti nel file, alla destra di ogni record è presente un checkbok per la selezione degli artefatti che si desidera importare. A fondo pagina troviamo un bottone che consente il salvataggio degli artefatti selezionati.



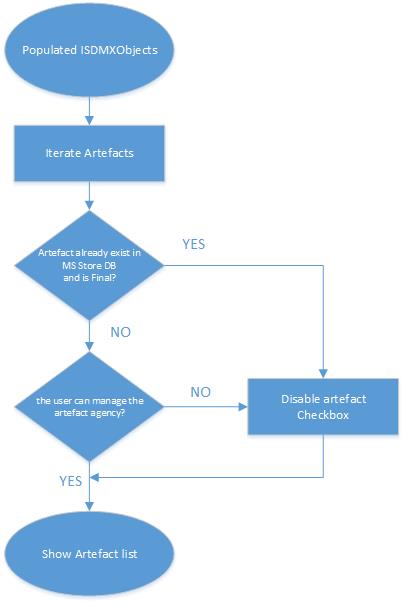
### Vincoli funzionali

Per effettuare l'inserimento di un artefatto dovranno essere rispettate le seguenti condizioni:

* Gli artefatti importabili non dovranno essere già presenti in base dati come finalizzati, in questo caso potranno solo essere visualizzati nella lista ma non importati.
* Potranno essere importati solo gli artefatti aventi 'Agency' gestibili dall'utente autenticato. La gestione delle agenzie è presente nella funzionalità di gestione degli Utenti

### Abilitazione artefatti importabili

Un file SDMX-ML quando si importa viene caricato in un oggetto di tipo ISDMXObjects, successivamente vengono iterate le liste di artefatti, per ogni elemento contenuto viene eseguita una chiamata al WS a cui viene passato un oggetto di tipo ArtefactIdentity. Se la chiamata al WS non restituisce nulla significa che l'artefatto non è presente nel DB e può essere importato, in caso contrario viene controllata la proprietà 'isFinal' che se risulta essere valorizzata a True(artefatto finalizzato) disabilita il checkbox relativo nella GridView. Oltre al controllo sulla finalizzazione dell'artefatto, ne viene effettuato un altro sull'utente che sta eseguendo l'operazione, viene controllato se l'utente è associato all'Agency dell'artefatto che si vuole importare. Questo controllo viene eseguito iterando la lista delle Agency associate all'utente, se non viene rilevata una corrispondeza il Checkbox della selezione artefatto viene disabilitato.



### Salvataggio artefatti

Gli artefatti selezionati dall'utente per poter essere importati nel DB vengono inseriti in un nuovo oggetto di tipo ISDMXObjects, quest' ultimo viene passato al metodo SubmitStructure() della classe WSModel che ne effetuerà il salvataggio.

## Autenticazione

L’autenticazione degli utenti può essere gestita in due diversi modi:

* Da file
* Tramite il Web Service ‘IstatRegistryWebService’.

E’ possibile definire la modalità di autenticazione tramite configurazione del file Web.Config.

### Autenticazione tramite File

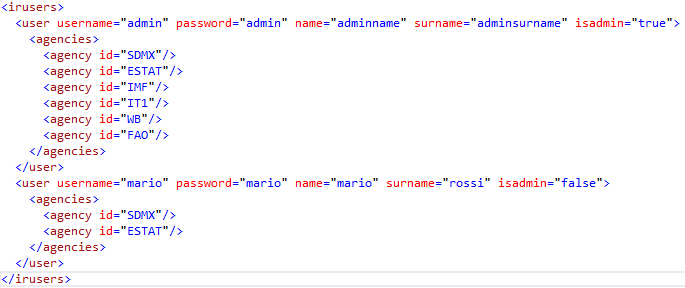
L’autenticazione tramite file è gestibile tramite la configurazione del file xml degli utenti.

Nella pagina master.Master è presente il form di autenticazione, una volta inserite le credenziali di autenticazione(username e password) vengono passate come parametri di input al metodo ‘GetUserFromFile’ presente nella stessa classe. Il metodo controlla se le credenziali inserite sono presenti nel File degli utenti, se trova una corrispondenza restituisce un oggetto di tipo ‘IRServiceReference.User’ popolato con le informazioni reperite dal file. Se l’oggetto restituito sarà diverso da null verrà memorizzato in una variabile di sessione e l’utente sarà autenticato, in caso contrario l’utente riceverà un messaggio di autenticazione fallita.

La struttura del file xml dovrà contenere le seguenti informazioni:

|  |  |  |
| --- | --- | --- |
| ***Attributo*** | ***Tipo*** | ***Descrizione*** |
| Username | String | Nome utente utilizzato per l’autenticazione |
| Password | String | Contiene la chave segreta(password) dell’utente |
| Name | String | Il nome di battesimo dell’utente |
| Surname | String | Il cognome dell’utente |
| IsAdmin | Boolean | Valorizzato a true se l’utente potrà usufruire dei privilegi di amministrazione, altrimenti a false. |
| Agencies | Sezione xml | La lista delle Agenzie che potrà gestire |

Esempio di file di configurazione utenti:



Metodi per l’autenticazione tramite File:

|  |  |  |
| --- | --- | --- |
| ***Metodo*** | ***Parametri di Input/Output*** | ***Descrizione*** |
| GetUserFromFile () | Input:   * UserName String * Password String   Output   * IRServiceReference.User | Restituisce un oggetto di tipo IRServiceReference.User Se le credenziali di accesso inserite sono corrette, altrimenti l’oggetto restituito non verrà inzializzato |

### Autenticazione tramite Web Service

Nella pagina master.Master è presente il form di autenticazione, una volta inserite le credenziali di autenticazione(username e password) vengono passate come parametri di input al metodo ‘GetUserByCredentials’ del Web service ‘IstatRegistryWebService’ il quale restituisce un oggetto di tipo IRServiceReference.User. Se l’oggetto restituito sarà diverso da null verrà memorizzato in una variabile di sessione e l’utente sarà autenticato, in caso contrario l’utente riceverà un messaggio di autenticazione fallita.

Metodi del Web Service ‘IstatRegistryWebService’ utilizzati

|  |  |  |
| --- | --- | --- |
| ***Web Method*** | ***Parametri di Input/Output*** | ***Descrizione*** |
| GetUserByCredentials() | Input:   * UserName String * Password String   Output   * IRServiceReference.User | Restituisce un oggetto di tipo IRServiceReference.User Se le credenziali di accesso inserite sono corrette, altrimenti l’oggetto restituito non verrà inzializzato |

## Amministrazione utenti

La pagina di amministrazione utenti può essere utilizzata per gestire le utenze quando è attiva l’Autenticazione tramite Web Service, non può essere utilizzata per l’autenticazione tramite File.

Tramite l’accesso al WebService ‘IstatRegistryWebService’ consente la gestione degli utenti.

Metodi del WebService ‘IstatRegistryWebService’ utilizzati:

|  |  |  |
| --- | --- | --- |
| ***Web Method*** | ***Parametri di Input/Output*** | ***Descrizione*** |
| GetAllUsers | Input:  Output   * Users Datatable | Restituisce un oggetto di tipo Datable contenente la lista di tutti gli User presenti in base dati. |
| GetUserById | Input:   * Int UserID   Output   * Datatable UserTable | Restituisce un oggetto di tipo Datable contenente i dati relativi alla UserID passata come parametro di Input. |
| DeleteUser | Input:   * Int UserID   Output   * Bool | Elimina un utente dal DB record. Restituisce true se l’eliminazione ha avuto successo. |
| GetCodesIdByUser | Input:   * Int UserID   Output   * String[] Codes | Restituisce un array di stringhe popolato con i codici delle agenzie associati alla UserId passata in input. |
| CheckIfUserExists | Input:   * Int UserID   Output   * Bool | Controlla l’esistenza di una UserID, restituisce true se trova una corrispondenza. |
| InsertUser | Input:   * String UserName * String Password * String Name * String Surname * Bool IsAdmin   Output   * Bool | Inserisce un nuovo utente nel DB e restituisce true se l’inserimento va a buon fine. |
| UpdateUser | Input:   * Int UserID * String UserName * String Password * String Name * String Surname * Bool IsAdmin   Output   * Bool | Aggiorna i dati di un utente. |
| CleanAgenciesRelationForUser | Input:   * Int UserID   Output | Elimina tutte le relazioni tra un Utente e le Agency |
| CreateUserAgenciesRelation | Input:   * Int UserID   Output   * String agencySchemeId * String agencySchemeAgency * String agencySchemeVersion * String agencyCode | Inserisce nel DB una relazione tra un Utente e una Agency. |

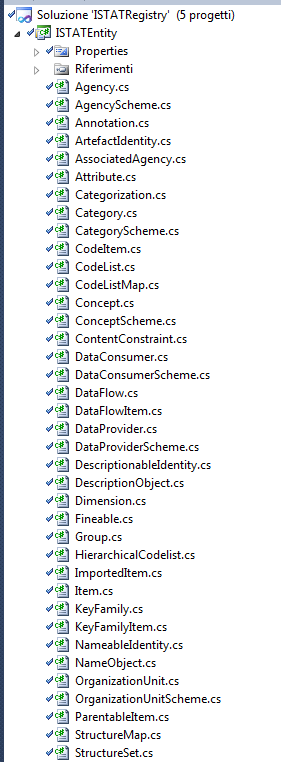
## File di configurazione (Web.config)

Di seguito una tabella riassuntiva delle sezioni configurate:

|  |  |
| --- | --- |
| ***Chiave/Sezione*** | ***Descrizione*** |
| ExportDotStatSettingsGroup | Questa sezione contiene informazioni di configurazione relative all'esportazione degli artefatti in formato .stat |
| EndPointSection | Questa sezione permette di configurare per ogni Endpoint i seguenti parametri:   * Name: La descrizione visualizzata nella dropdownlist * NSIEndPoint: End point del Web Service NSI * IREndPoint: End point del Web Service Istat Registry * PartialArtefact: Questo parametro di tipo booleano abilità o meno la visualizzazione dei nuovi artefatti nel menu di navigazione. Se valorizzato con true verranno visualizzati solamente gli artefatti gestiti dal WS di Eurostat, in caso contrario saranno presenti tutti. * EnableAuthentication: Parametro booleano che abilita l'autenticazione. * EnableAnnotationSuggest: Parametro booleano che abilità i suggerimenti sulle Annotation dei Dataflow. * EnableAdministration: Parametro booleano che abilita la pagina di amministrazione |
| AnnotationTypes | Questa sezione contiene gli elementi che andranno a popolare la dropdownlist degli Annotation Type. Le chiavi sono formate da un identificativo e un value, il primo apparirà come descrizione del combo, il secondo invece andrà a popolare la text area del type. |
| AnnotationValues | Questa sezione contiene gli elementi che andranno a popolare la dropdownlist degli Annotation Value. Le chiavi sono formate da un identificativo e un value, il primo apparirà come descrizione del combo, il secondo invece andrà a popolare la text area del Value. |
| Chiavi utilizzate dall'assembly FlyCallWS | Le seguenti chiavi di configurazione vengono utilizzate dall'assembly FlyCallWS, questa libreria fornisce funzionalità di accesso ai WebService.  NOTA: ATTUALMENTE LE SEGUENTI CHIAVI VENGONO UTILIZZATE PER TUTTI GLI END POINT.   * EndPoint20: Costante per gli endpoint di tipologia 2.0 * EndPoint21: Costante per gli endpoint di tipologia 2.1 * Prefix: * EnableHTTPAuthenication: Abilità l'autenticazione HTTP * Domain: Dominio utilizzato per l'autenticazione * Username: Username utilizzata per l'autenticazione * Password: Password utilizzata per l'autenticazione * EnableProxy: Abilita il proxy * UseSystemProxy: Usa un proxy * ProxyServer: L'indirizzo del proxy * ProxyServerPort: La porta del proxy * ProxyUsername: Username utilizzata dal proxy * ProxyPassword: Password utilizzata dal proxy * MaxOutputFileLength: Dimensione massima dei file di output espressa in Byte |
| ViewMode | Se valorizzato a true abilità la modalità 'Visualizzazione' del sito, il CRUD verrà disabilitato. |
| DefaultLanguageForResources | Linguaggio di default utilizzato dall'applicazione |
| Chiavi Gridview Default Result row | Le seguenti chiavi impostano il numero di righe dei risultati che verranno impostate di default per le Gridview:  GeneralCodelistGridNumberRow  DetailsCodelistGridNumberRow  GeneralConceptschemeGridNumberRow  DetailsConceptschemeGridNumberRow  GeneralKeyFamilyGridNumberRow  GeneralCategoryschemeGridNumberRow  GeneralDataflowGridNumberRow  GeneralCategorizationGridNumberRow  GeneralAgencySchemeGridNumberRow  DetailsAgencyschemeGridNumberRow  GeneralDataProviderSchemeGridNumberRow  DetailsDataProviderSchemeGridNumberRow  GeneralDataConsumerSchemeGridNumberRow  DetailsDataConsumerSchemeGridNumberRow  GeneralOrganizationUnitSchemeGridNumberRow  DetailsOrganizationUnitSchemeGridNumberRow  GeneralContentConstraintNumberRow  GeneralStructureSetNumberRow |

# Progetto ISTATEntity

In questo progetto sono presenti le classi di entità dove vengono rimappati i dati SDMX delle Common API di Eurostat. Vengono utilizzate principalmente per il Data Bind delle Gridview presenti nelle pagine .aspx.  
Namespace: ISTAT.Entity



# Progetto ISTATEntityMapper

Il progetto ISTATEntityMapper viene utilizzato dall'applicazione Web per il mapping dei dati SDMX delle Common API di Eurostat verso delle classi Interne(ISTATEntity). Il progetto si compone di un'unica classe: 'EntityMapper.cs'.   
La classe espone i seguenti metodi pubblici:

|  |  |  |
| --- | --- | --- |
| ***Nome metodo*** | ***Parametri di input e output*** | ***Descrizione*** |
| GetCodeListList() | Input: ISdmxObjects Output: List<ISTAT.Entity.CodeList> | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo Codelist e li rimappa in una lista di oggetti di tipo ISTAT.Entity.CodeList |
| GetConceptSchemeList() | Input: ISdmxObjects Output: List<ISTAT.Entity. ConceptScheme> | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo ConceptScheme e li rimappa in una lista di oggetti di tipo ISTAT.Entity.ConceptScheme |
| GetAgencySchemeList() | Input: ISdmxObjects Output: List<ISTAT.Entity. AgencyScheme> | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo AgencyScheme e li rimappa in una lista di oggetti di tipo ISTAT.Entity.AgencyScheme |
| GetDataProviderSchemeList() | Input: ISdmxObjects Output: List<ISTAT.Entity. DataProviderScheme > | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo DataProviderScheme e li rimappa in una lista di oggetti di tipo ISTAT.Entity. DataProviderScheme |
| GetOrganizationUnitSchemeList | Input: ISdmxObjects Output: List<ISTAT.Entity. OrganizationUnitScheme > | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo OrganizationUnitScheme e li rimappa in una lista di oggetti di tipo ISTAT.Entity. OrganizationUnitScheme |
| GetDataConsumerSchemeList | Input: ISdmxObjects Output: List<ISTAT.Entity. DataConsumerScheme> | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo DataConsumerSchemee li rimappa in una lista di oggetti di tipo ISTAT.Entity. DataConsumerScheme |
| GetCategorySchemeList | Input: ISdmxObjects Output: List<ISTAT.Entity. CategoryScheme > | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo CategoryScheme li rimappa in una lista di oggetti di tipo ISTAT.Entity. CategoryScheme |
| GetDataFlowList | Input: ISdmxObjects Output: List<ISTAT.Entity. DataFlow> | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo DataFlowli rimappa in una lista di oggetti di tipo ISTAT.Entity. DataFlow |
| GetKeyFamilyList | Input: ISdmxObjects Output: List<ISTAT.Entity. KeyFamily > | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo KeyFamily rimappa in una lista di oggetti di tipo ISTAT.Entity. KeyFamily |
| GetContentConstraintList | Input: ISdmxObjects Output: List<ISTAT.Entity. ContentConstraint > | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo KeyFamily rimappa in una lista di oggetti di tipo ISTAT.Entity. ContentConstraint |
| GetStructureSetList | Input: ISdmxObjects Output: List<ISTAT.Entity. StructureSet > | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo StructureSet rimappa in una lista di oggetti di tipo ISTAT.Entity. StructureSet |
| GetHclList | Input: ISdmxObjects Output: List<ISTAT.Entity. HierarchicalCodelist> | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo HierarchicalCodelistrimappa in una lista di oggetti di tipo ISTAT.Entity. HierarchicalCodelist |
| GetCodeItemList | Input: ISdmxObjects Output: List<ISTAT.Entity. CodeItem> | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli Item delle Codelist e li rimappa in una lista di oggetti di tipo ISTAT.Entity. CodeItem |
| GetConceptList | Input: ISdmxObjects Output: List<ISTAT.Entity. Concept> | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli Item delle Conceptscheme e li rimappa in una lista di oggetti di tipo ISTAT.Entity. Concept |
| GetCategoryList | Input: ISdmxObjects Output: List<ISTAT.Entity. Category> | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli Item dei Categoryscheme e li rimappa in una lista di oggetti di tipo ISTAT.Entity. Category |
| GetCategorizationList | Input: ISdmxObjects Output: List<ISTAT.Entity. Categorization > | Prende in input un oggetto di tipo ISdmxObjects, recupera i dati relativi agli artefatti di tipo Categorisation e li rimappa in una lista di oggetti di tipo ISTAT.Entity. Categorization |
| GetDataFlowItemList | Input: ISdmxObjects, ISdmxObjects Output: List<ISTAT.Entity. DataFlowItem> | Prende in input due oggetti di tipo ISdmxObjects, uno contenente oggetti di tipo Dataflow e uno contenente oggetti di tipo Categoryscheme recupera i dati che cartatterizzano un DataFlow e li rimappa in una lista di oggetti di tipo ISTAT.Entity. DataFlowItem |
| GetDimensionList | Input: IDataStructureObject Output: List<ISTAT.Entity. Dimension > | Prende in input un oggetto di tipo IDataStructureObject, recupera i dati relativi alle dimensioni contenute e li rimappa in una lista di oggetti di tipo ISTAT.Entity. Dimension |
| GetAttributeList | Input: IDataStructureObject Output: List<ISTAT.Entity. Attribute> | Prende in input un oggetto di tipo IDataStructureObject, recupera i dati relativi agli attributi e li rimappa in una lista di oggetti di tipo ISTAT.Entity. Attribute |
| GetGroupList | Input: IDataStructureObject Output: List<ISTAT.Entity. Group> | Prende in input un oggetto di tipo IDataStructureObject, recupera i dati relativi ai Gruppi e li rimappa in una lista di oggetti di tipo ISTAT.Entity. Group |
| GetCodeListMapList | Input: IStructureSetObject Output: List<ISTAT.Entity. CodeListMap> | Prende in input un oggetto di tipo IStructureSetObject, recupera i dati relativi agli Item delle CodeMapList e li rimappa in una lista di oggetti di tipo ISTAT.Entity. CodeListMap |
| GetStructureMapList | Input: IStructureSetObject Output: List<ISTAT.Entity. StructureMap > | Prende in input un oggetto di tipo IStructureSetObject, recupera i dati relativi agli Item delle StructureMapList e li rimappa in una lista di oggetti di tipo ISTAT.Entity. StructureMap |

# Progetto ISTATWSDAL

Il progetto ISTATWSDAL ha la funzionalità di wrapper verso il WS, espone le interfacce per l’accesso ai dati.

## Classe EndpointSettings

Questa classe viene utilizzata per rimappare le proprietà di configurazione utilizzate per la connessione verso il Web Service.

Espone le seguenti proprietà:

|  |  |
| --- | --- |
| ***Nome Proprietà*** | ***Tipo*** |
| Locale | String |
| Title | String |
| Domain | String |
| EnableHTTPAuthentication | String |
| EnableProxy | String |
| EndPoint | String |
| EndPointV20 | String |
| EndPointType | String |
| Password | String |
| Prefix | String |
| ProxyPassword | String |
| ProxyServer | String |
| ProxyServerPort | String |
| ProxyUserName | String |
| UseSystemProxy | Bool |
| UserName | String |
| Wsdl | String |
| Active | Bool |
| UseUncategorysed | Bool |
| UseVirtualDf | Bool |

## Classe WSConstants

La classe WSConstants contiene le costanti utilizzate dall’Assembly WSDAL.

Proprietà esposte:

|  |  |  |
| --- | --- | --- |
| ***Nome Proprietà*** | ***Tipo*** | ***Descrizione*** |
| wsOperation | Enum | La lista degli artefatti gestiti dalla libreria |
| xmlTemplate | String | Template XML utilizzato per caricare in un oggetto di tipo SDMXObjects la response restituita dal WS |
| wsEndPoint | String | L’End Point utilizzato per la connettersi al WS |
| MaxOutputFileLength | String | Il numero di massimo di byte che un file di output può contenere |

## Classe WSModel

Questa è la classe Entry Point della libreria WSDAL, espone i metodi pubblici per il recupero e il salvataggio degli artefatti.

### Metodi di retrieve degli artefatti

I metodi di Retrieve degli artefatti restituiscono un oggetto di tipo ISdmxObjects e prendono in input:

* ArtefactIdentity(ID,Agency,Version): l’identificativo degli artefatti da ricercare
* Booleano Stub: se valorizzato con true restituisce la versione ridotta degli artefatti(solo info di base), con false restituisce l’artefatto completo
* Booleano Like: se valorizzato con true esegue una ricerca degli artefatti per Like

### Metodi di submit degli artefatti

Il metodo pubblico SubmitStructure permette il salvataggio degli artefatti. Passato un oggetto di input(ISdmxobjcts o XmlDocument) effettuà la connessione al WS e gli passa l’oggetto da caricare nel DB.

## Classe WSUtils

La classe di utilità della libreria

Metodi esposti:

|  |  |  |
| --- | --- | --- |
| ***Metodo*** | ***Parametri di input/Output*** | ***Descrizione*** |
| ConvertToBytes | Input:   * XmlDocument   Output:   * byte[] | Converte un XmlDocument in un array di byte |
| GetSettings | Input:   * String   Output:   * WsConfigurationSettings | Prende in input una operation e restituisce un oggetto di tipo WsConfigurationSettings che contiene la configurazione per l’accesso al WS |
| GetXMLDocFromSdmxObjects | Input:   * ISdmxObjects * StructureOutputFormatEnumType   Output:   * XmlDocument | Converte un oggetto di tipo ISdmxObjects in XmlDocument |
| GetSdmxObjectsFromXML | Input:   * XmlDocument   Output:   * ISdmxObjects | Converte un oggetto di tipo XmlDocument in ISdmxObjects |