|  |
| --- |
| **Documentazione tecnica**  ***Istat - WebClientCachingService***  Versione 2.0 del 11/09/2015 |

|  |  |  |
| --- | --- | --- |
| **Redatto da** | Massimiliano Gianni | IT Engineer |
| **Revisionato da** | <Nominativo> | <Qualifica> |
| **Committente** | | |
| **Dipartimento** |  | |
| **Direzione** |  | |
| **Servizio** |  | |

Sommario

[***Istat - WebClientCachingService*** 1](#_Toc429987205)

[1. Introduzione 3](#_Toc429987206)

[2. Lista oggetti DB 4](#_Toc429987207)

[2.1 Diagramma tabelle 4](#_Toc429987208)

[2.2 Descrizione tabelle 4](#_Toc429987209)

[Descrizione tabelle DB 4](#_Toc429987210)

[2.3 Descrizione tabella SavedWidget 5](#_Toc429987211)

[Tabella SavedWidget 5](#_Toc429987212)

[2.4 Stored Procedures 5](#_Toc429987213)

[Lista Stored Procedures 5](#_Toc429987214)

[3. Architettura applicazione 7](#_Toc429987215)

[3.1 Progetto ISTAT.WebClient.WidgetComplements 7](#_Toc429987216)

[3.2 Progetto ISTAT.WebClient.WidgetEngine 8](#_Toc429987217)

[3.3 Progetto ISTAT.WebClient.CachingService 9](#_Toc429987218)

[3.3.1 Descrizione del file di configurazione app.config 9](#_Toc429987219)

[3.3.2 Descrizione del file di configurazione log4net.xml 9](#_Toc429987220)

[3.3.3 Descrizione del file di classe Program.cs 9](#_Toc429987221)

[3.3.4 Descrizione del file di classe CachingManager.cs 10](#_Toc429987222)

[3.3.4.1 Eccezioni CachingManager 10](#_Toc429987223)

[Lista eccezioni componente CachingManager 10](#_Toc429987224)

[4. Configurazione file Web.config di IstatWebClient 12](#_Toc429987225)

# Introduzione

Il presente documento, descrive l'implementazione tecnica del software ISTAT.WebClientCachingService.

Il software ISTAT.WebClientCachingService è un'applicazione di tipo Windows Service che effettua una copia cache dei dati utilizzati dall'applicativo Istat WebClient per il popolamento e rendering delle Dashboard. Questa funzionalità garantisce una maggiore velocità di visualizzazione dei dati, in quanto permette all'applicazione web di caricare i dati leggendoli dal DB invece di richiederli ogni volta tramite delle chiamate SOAP.

# Lista oggetti DB

Di seguito gli oggetti DB utilizzati per la realizzazione del software

## Diagramma tabelle



Figura - Diagramma tabelle

## Descrizione tabelle

|  |  |
| --- | --- |
| Descrizione tabelle DB | |
| **Nome tabella** | **Descrizione** |
| Dashboard | Memorizza le informazioni primarie della Dashboard |
| Dashboard\_row | Memorizza le informazioni relative alle righe utilizzate dalla Dashboard  Relaziona la Dashboard con il widget |
| Widgets | Memorizza i dati relativi ai Widget |
| Widget\_text | Memorizza i dati principali localizzati dei widget |
| SavedWidget | Questa tabella è stata aggiunta per la gestione del caching dei widget. Memorizza il contenuto dei widget in formato JSON o HTML |

## Descrizione tabella SavedWidget

Tabella utilizzata per il caching dei Widget

|  |  |  |
| --- | --- | --- |
| Tabella SavedWidget | | |
| **Nome campo** | **Tipo** | **Descrizione** |
| SWID | Int, Identity, not null, PK | Campo chiave della tabella |
| WDG\_ID | Int, FK | Campo di relazione verso la tabella Widget |
| WidgetData | Ntext | Questo campo memorizza il contenuto dei Widget in formato JSON o HTML |
| Locale | Varchar(2) | Identifica la localizzazione dei dati memorizzati |
| DTUpdate | Datetime | Data ultimo aggiornamento del widget |
| Deleted | Bit | Utilizzato per la cancellazione logica della riga |

## Stored Procedures

|  |  |  |
| --- | --- | --- |
| Lista Stored Procedures | | |
| **Nome S.P.** | **Parametri** | **Descrizione** |
| Caching.ClearWidgetCache | Nessuno | Svuota la tabella SavedWidget. Elimina tutti i record in essa contenuti. |
| Caching.DeleteExpiredWidget | @SECONDS, Int, Input | Effettua una eliminazione logica dei widget scaduti della tabella SavedWidget. Imposta il campo Deleted a 1. |
| Caching.DeleteWidget | @WDG\_ID, Int, Input | Elimina fisicamente un widget memorizzato nella tabella SavedWidget |
| Caching.GetUpdateableWidgets | Nessuno | Restituisce un resultset contenente la lista dei widget aggiornabili(scaduti o non ancora inseriti nella cache) |
| Caching.GetWidget | @WDG\_ID, Int, Input  @LOCALE, Varchar, Input | Restituisce un widget memorizzato nella tabella SavedWidget in base ai parametri passati in input |
| Caching.InsertWidget | @WDG\_ID, Int, Input  @WIDGETDATA, Ntext, Input  @LOCALE, Varchar, Input | Inserisce un record nella tabella SavedWidget |
| Caching.IsCachedWidget | @WDG\_ID, Int, Input  @LOCALE, Varchar, Input  @EXIST, Bit, Output | Restituisce un booleano valorizzato a true se il widget richiesto è presente nella tabella SavedWidget, altrimenti false. |
| Caching.GetWidgetInfo | @LOCALE VARCHAR(5),  @WDG\_ID INT | Restituisce il titolo del Widget e della Dashboard |

# Architettura applicazione

Architettura della soluzione ISTAT.WebClientCaching

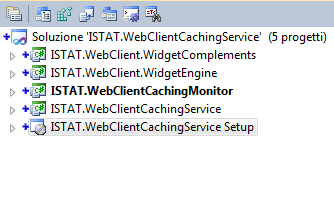


Figura - Soluzione software

L'applicativo di caching si compone di 5 progetti:

* ISTAT.WebClient.WidgetComplements
* ISTAT.WebClient.WidgetEgine
* ISTAT.WebClientCachingService
* ISTAT.WebClientCachingMonitor
* ISTAT.WebClientCachingService Setup

## Progetto ISTAT.WebClient.WidgetComplements

Questo progetto è parte integrante dell'applicativo Web ISTAT.WebClient, viene utilizzato dal software di Caching come strato di Model. Difatti questo assembly espone sia le funzionalità per l'accesso ai Web Services che quelle per l'accesso ai Database.

Il modulo software implementato: "CacheWidget.cs", svolge una funzione di wrapping verso il DB, espone i metodi per il richiamo delle Stored Procedures.

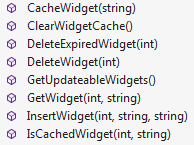


Figura - CacheWidget.cs, dettaglio metodi

## Progetto ISTAT.WebClient.WidgetEngine

Anche questo progetto deriva dell'applicativo Web ISTAT.WebClient, viene utilizzato dal software di Caching per rimappare i dati delle Dashboard derivanti dal Database in oggetti interni.

In questo progetto sono stati inseriti dei controlli per il recupero dei dati della cache.

Di seguito il flusso eseguito per il recupero dei dati:

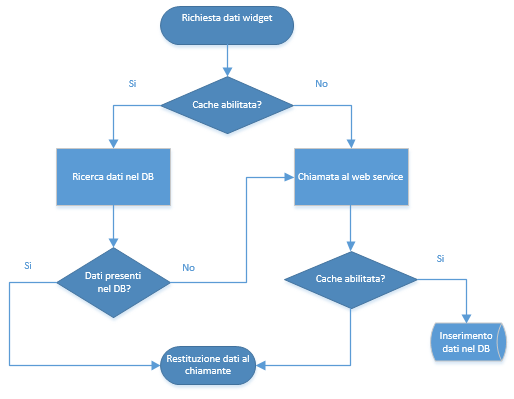


Figura - Flusso recupero dati

## Progetto ISTAT.WebClient.CachingService

L'ISTAT.WebClient.CachingService è il progetto che si occupa di creare una copia cache dei dati delle Dashboard utilizzate dall'applicativo web e di memorizzarle nel Database.

Il progetto è composto dai seguenti moduli:

* app.config: file di configurazione dell'applicazione
* log4net.xml: file di configurazione per la scrittura dei log su file
* Program.cs: file che viene lanciato all'avvio del programma
* CachingManager.cs: file che contiene la logica del programma

## Descrizione del file di configurazione app.config

Il file app.config è un file xml di configurazione dell'applicazione ed è composto dalle seguenti informazioni:

* ***Section <connectionStrings>:*** questa sezione contiene la lista delle ConnectionString per l'accesso al Database, le credenziali di autenticazione e il tipo di provider utilizzato (System.Data.SqlClient). Al momento l'applicazione web è stata progettata per lavorare esclusivamente con DB MS SQL Server.
* ***EnableCaching:*** chiave di configurazione di tipo booleano utilizzata per attivare il caching, se viene valorizzata con True la funzionalità di cache viene attivata.
* ***EnableCacheWidget:*** analogamente alla chiave precedente se valorizzata con True abilita il caching dei widget.
* ***EnableCacheWidgetLog*:** se valorizzata con True abilita la scrittura del Log su file
* ***PollingCacheWidget:*** questa chiave di tipo Integer, contiene il numero di secondi che devono trascorrere per verificare se sul Database sono presenti dati di Widget da memorizzare nel DB
* ***RefreshCacheWidget:*** chiave di tipo Integer che viene valorizzata con il numero di secondi di validità del widget
* ***ParseSDMXAttributes:*** questa chiave di tipo booleano, se valorizzata con True abilita la richiesta degli attributi dati SDMX. La stessa chiave è presente anche nell'applicazione Web, **le 2 chiavi devono essere valorizzate con lo stesso valore**

## Descrizione del file di configurazione log4net.xml

Il file log4net.xml è un file di configurazione di tipo xml utilizzato dal componente **log4net** responsabile del logging su file.

Tra le varie opzioni di configurazione nel file è presente il percorso di destinazione dei file di log.

Ulteriori informazioni è possibile reperirle dal sito ufficiale: **https://github.com/apache/log4net.**

**Nota:**

L'attributo <level value="OFF"> deve rimanere valorizzato con OFF, l'abilitazione viene gestita da programma.

## Descrizione del file di classe Program.cs

In questo file è presente l'entry point dell'applicazione metodo "main". Una volta lanciata l'applicazione viene richiamato questo metodo si occupa di:

* Inizializzare il componente di log
* Istanziare il componente di caching CachingManager()
* Mantenere l'applicazione attiva fin quando il parametro di configurazione ***EnableCaching*** è valorizzato con True.

## Descrizione del file di classe CachingManager.cs

Questo è il file che contiene gli algoritmi c# per la memorizzazione della cache sul DB.

Come prima operazione viene creato un thread che verrà richiamato a intervalli regolari di X secondi(chiave ***PollingCacheWidget*** definita nell'app.config).

Nel thread(metodo RefreshWidget()) vengono effettuate le seguenti operazioni:

* Per ogni stringa di connessione(DB) presente nel file app.config vengono effetuate le seguenti operazioni:
* Viene richiamata la stored procedure **DeleteExpiredWidget** a cui viene passato il parametro @SECONDS(chiave ***RefreshCacheWidget*** definita nell'app.config).Questa procedura SQL effettua una cancellazione logica dei widget scaduti.
* Viene richiamata la stored procedure GetUpdateableWidgets() che restituisce la lista dei widget che dovrà essere aggiornata con nuovi dati.
* Per ogni widget da aggiornare:
  + Per i widget di tipo Chart, viene richiamato il metodo GetDataChart() a cui viene passato in input un oggetto di tipo ChartWidget. Le proprietà di configurazione presenti in questo oggetto(endpoint, localization, dataflow id ecc...) verranno utilizzate da GetDataChart() per effettuare una chiamata soap verso il web service dei dati che restituirà il risultato in formato JSON. Se la risposta del metodo sarà corretta, i dati presenti nel DB relativi al widget verranno sostituiti con i nuovi, altrimenti verranno inseriti se non presenti.
  + Per i widget di tipo Datatable, viene utilizzata la stessa logica dei widget di tipo Chart. Le uniche varianti sono che i metodi che vengono richiamati sono 2: GetData e StreamDataTable che si occupano rispettivamente di fare la chiamata Soap e di restituire il risultato che questa volta sarà in formato HTML.

## Eccezioni CachingManager

Durante le operazioni di caching si potrebbero verificare delle eccezioni dovute a problemi di connessione od altro. Di seguito una tabella descrittiva delle possibili eccezioni verificabili:

|  |  |
| --- | --- |
| Lista eccezioni componente CachingManager | |
| **N° eccezione** | **Descrizione** |
| 1 | Quando viene richiamato il metodo che effettua la chiamata soap che restituisce i dati, tra i vari parametri di configurazione è presente il parametro "locale" valorizzato con la lingua di interesse(en, it ecc...), nel caso in cui il metodo non dovesse trovare dati per la lingua richiesta restituirà le informazioni relative alla prima lingua trovata. |
| 2 | Tutti i dati visualizzati dai widget e memorizzati nel DB vengono restituiti attraverso chiamate di tipo SOAP. Nel caso in cui una chiamata verso il web service non vada a buon fine verranno conservati nella cache(DB) i dati relativi all'ultima chiamata avvenuta senza errori. Un errore verrà visualizzato nella console application e contestualmente inserito nel file di log(se attivo). |

* 1. **Progetto** **ISTAT.WebClientCachingMonitor**

Questo progetto di tipo eseguibile, viene installato parallelamente al Servizio Windows. Tramite una semplice interfaccia grafica permette all'utente la visualizzazione delle voci di Log del servizio, di visualizzarne lo stato corrente, di avviarlo o arrestarlo. L'eseguibile è presente come icona nella notification area del sistema operativo.

* + 1. **Il progetto si compone dei seguenti file**
* Program.cs
* StartupManager.cs
* TaskTrayApplicationContext.cs
  + - 1. **File Program.cs**

In questo file è presente l'entry point dell'applicazione metodo "main".

* + - 1. **File TaskTrayApplicationContext**

Questa classe eredita da 'ApplicationContext' e si occupa di:

* avvia l'applicazione nella notification area.
* aggiunge nel menu contestuale la voce 'Exit'
* sul doppio click apre l'interfaccia per la visualizzazione del log
  + - 1. **File ShowLog.cs**

Questo form rappresenta l'interfaccia utente per la visualizzazione del log e per la gestione del servizio.

Il form oltre a visualizzare il log in formato testuale permette di visualizzare lo stato corrente del servizio (running, stopped, not found) di avviarlo o arrestarlo. E' inoltre presente un checkbox per l'autorefresh del log a intervalli regolari di 2 secondi e un tasto per il refresh manuale.

La parte relativa alla gestione del servizio(start e stop) potrà essere utilizzata dall'utente solo se l'applicativo verrà lanciato in modalità amministratore.

* 1. **Progetto ISTAT.WebClientCachingService Setup**

Questo è il progetto di setup, si occupa di creare i file di installazione dell'applicazione.

* + 1. **Operazioni svolte durante l'installazione**

Il progetto di setup è configurato per svolgere le seguenti operazioni in fase di installazione:

* Installazione del servizio windows nel computer di destinazione
* Avvio del servizio
* Copia del file di monitoring(ISTAT.WebClientCachingMonitor.exe) nella directory del servizio
* Creazione del gruppo di programmi 'ISTAT' nel menu di Start
* Creazione e copia del collegamento del file di monitoring nella cartella 'ISTAT'
* Inserimento della chiave di registry per l'avvio automatico dell'applicazione di monitoring

# Configurazione file Web.config di IstatWebClient

Per la configurazione della cache nell'applicazione IstatWebClient è presente la chiave 'UseWidgetCache' sotto la sezione 'WebClientSettings' del file Web.config.

WebClientConfig.png

Figura - Chiave Web.config

Valorizzando la chiave con **true**, l'applicazione Web leggerà dal DB i dati per la popolazione dei Widget, se presenti. Nel caso in cui nel DB non siano presenti dati, verranno effettuate delle chiamate SOAP per il recupero degli stessi.

Se la chiave viene valorizzata con **false**, verrà ignorata la cache del DB.