MetaDataManager - Installation manual

Release 1.9.0

ISTAT

Nov 11, 2022

CONTENTS

CHAPTER

ONE

INTRODUCTION

1.1 Aim of this document

Provide an installation guide of the base software, databases and web services needed to the tool Meta & Data Manager.

1.2 Acronyms and glossary

Definition / Acronym	Description
.NET	Microsoft software development framework.
.NET Core	Free and open source software development framework for different operating systems: Microsoft Windows, MacOS e Linux
AuthDB	Database used for the authentication and authorization. It is released by Eutostat and enhanced by Istat
DCAT-AP	Application Profile for data catalogue
DDB	Dissemination Data Base
DM API WS	Web Service for managing Data, Reference Metadata, authentication and authorization
IIS	Internet Information Services
MA API WS	Mapping Assistant API developed by Eurostat
MSDB	Mapping Store Data Base
NSI	National Statistical Institute
NSI WS	SDMX Web Service released by Eurostat
RMDB	Referential Metadata Data Base
SDMX	Statistical Data and Metadata eXchange
OS	Operating System

CHAPTER

TWO

SOFTWARE INSTALLATION

2.1 Pre-requisites

2.1.1 Operating system

The supported operating systems are the same ones supported by .NET Core. In the following table, the Microsoft Operating System supported are listed.

Windows Client	7 SP1+, 8.1	x64, x86
Windows 10 Client	Version 1607+	x64, x86
Windows Server	2008 R2 SP1+	x64, x86

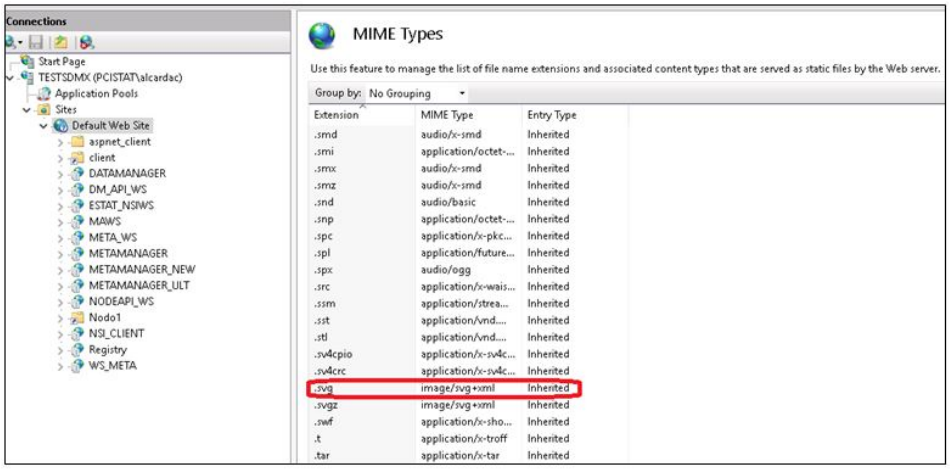
2.1.2 Sql Server

The supported database is Microsoft Sql Server ver.2012 or higher.

2.1.3 IIS

IIS has to be installed in a version supported by the used windows operating system. Please verify that the .svg e .json MIME types are available:

1. click on the IIS Web Site under which has to be installed the application;
2. double click the MIME Types;
3. verify the following mime types:
    - .svg image/svg+xml
    - .json application/json
4. If they are missing, add them by right clicking and selecting 'Add'



MIME Types

Use this feature to manage the list of file name extensions and associated content types that are served as static files by the Web server.

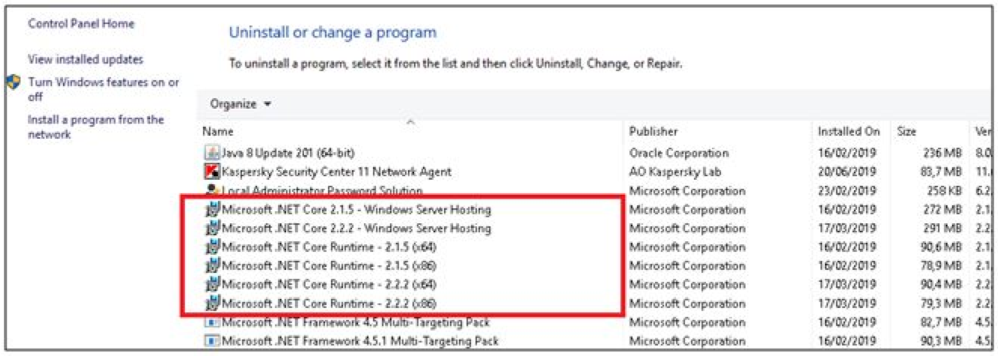
Extension	MIME Type	Entry Type
.smd	audio/x-smd	Inherited
.smi	application/octet-...	Inherited
.smx	audio/x-smd	Inherited
.smz	audio/x-smd	Inherited
.snd	audio/basic	Inherited
.snp	application/octet-...	Inherited
.spc	application/x-pkc...	Inherited
.spl	application/future...	Inherited
.spx	audio/ogg	Inherited
.src	application/x-wais...	Inherited
.ssm	application/strea...	Inherited
.sst	application/vnd....	Inherited
.stl	application/vnd....	Inherited
.sv4cpio	application/x-sv4c...	Inherited
.sv4crc	application/x-sv4c...	Inherited
.svg	image/svg+xml	Inherited
.svgz	image/svg+xml	Inherited
.swf	application/x-sho...	Inherited
.t	application/x-troff	Inherited
.tar	application/x-tar	Inherited

MIME Types

2.1.4 .NET Core

The **.NET Core Framework ver. 3.1.0** has to be installed together with **.NET Core hosting bundle for IIS**, verifying before the compliance with all the related prerequisites (as specified in https://docs.microsoft.com/it-it/dotnet/core/install/dependencies?pivots=os-windows&tabs=netcore31).

In order to verify if these modules have been already installed, access to: *Control Panel/Programs/Programs and functions*:


Control Panel Home
View installed updates
Turn Windows features on or off
Install a program from the network
Organize

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

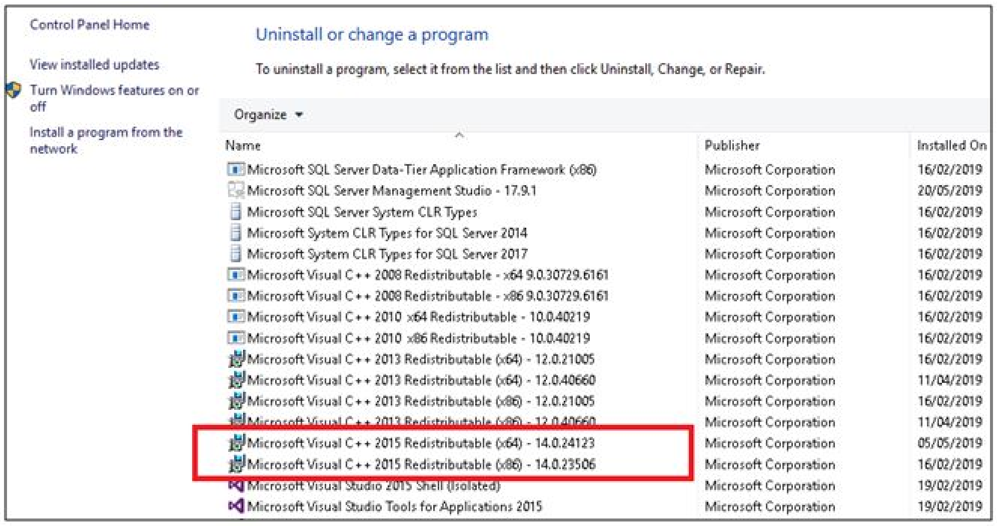
Name	Publisher	Installed On	Size	Ver
Java 8 Update 201 (64-bit)	Oracle Corporation	16/02/2019	236 MB	8.0
Kaspersky Security Center 11 Network Agent	AO Kaspersky Lab	20/06/2019	83,7 MB	11.
Local Administrator Password Solution	Microsoft Corporation	23/02/2019	258 KB	6.2
Microsoft .NET Core 2.1.5 - Windows Server Hosting	Microsoft Corporation	16/02/2019	272 MB	2.1
Microsoft .NET Core 2.2.2 - Windows Server Hosting	Microsoft Corporation	17/03/2019	291 MB	2.2
Microsoft .NET Core Runtime - 2.1.5 (x64)	Microsoft Corporation	16/02/2019	90,6 MB	2.1
Microsoft .NET Core Runtime - 2.1.5 (x86)	Microsoft Corporation	16/02/2019	78,9 MB	2.1
Microsoft .NET Core Runtime - 2.2.2 (x64)	Microsoft Corporation	17/03/2019	90,4 MB	2.2
Microsoft .NET Core Runtime - 2.2.2 (x86)	Microsoft Corporation	17/03/2019	79,3 MB	2.2
Microsoft .NET Framework 4.5 Multi-Targeting Pack	Microsoft Corporation	16/02/2019	82,7 MB	4.5
Microsoft .NET Framework 4.5.1 Multi-Targeting Pack	Microsoft Corporation	16/02/2019	90,3 MB	4.5

.Net Core

If .NET Core has not been already installed, is possible to proceed as follows:

1. Download the package from the following URL: https://dotnet.microsoft.com/download/thank-you/dotnet-runtime-3.1.0-windows-hosting-bundle-installer. Install from the package the .NET Core runtime, the libraries and the ASP.NET Core module for IIS
2. Execute the **"net stop was /y"** e **"net start w3svc"** commands from the Windows Prompt started in administration mode in order to apply all the updates to IIS.

Warning! The installation of the Microsoft Visual C++ 2015 Redistributable Update 3 or higher is needed. It is possible to verify if it has been already installed from: *Control panel/Programs/Programs and functions* or, alternatively, the install package can be downloaded from the following URL: https://www.microsoft.com/en-us/download/details.aspx?id=52685.


Control Panel Home
View installed updates
Turn Windows features on or off
Install a program from the network
Organize

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Name	Publisher	Installed On
Microsoft SQL Server Data-Tier Application Framework (x86)	Microsoft Corporation	16/02/2019
Microsoft SQL Server Management Studio - 17.9.1	Microsoft Corporation	20/05/2019
Microsoft SQL Server System CLR Types	Microsoft Corporation	16/02/2019
Microsoft System CLR Types for SQL Server 2014	Microsoft Corporation	16/02/2019
Microsoft System CLR Types for SQL Server 2017	Microsoft Corporation	16/02/2019
Microsoft Visual C++ 2008 Redistributable - x64 9.0.30729.6161	Microsoft Corporation	16/02/2019
Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.6161	Microsoft Corporation	16/02/2019
Microsoft Visual C++ 2010 x64 Redistributable - 10.0.40219	Microsoft Corporation	16/02/2019
Microsoft Visual C++ 2010 x86 Redistributable - 10.0.40219	Microsoft Corporation	16/02/2019
Microsoft Visual C++ 2013 Redistributable (x64) - 12.0.21005	Microsoft Corporation	16/02/2019
Microsoft Visual C++ 2013 Redistributable (x64) - 12.0.40660	Microsoft Corporation	11/04/2019
Microsoft Visual C++ 2013 Redistributable (x86) - 12.0.21005	Microsoft Corporation	16/02/2019
Microsoft Visual C++ 2013 Redistributable (x86) - 12.0.40660	Microsoft Corporation	11/04/2019
Microsoft Visual C++ 2015 Redistributable (x64) - 14.0.24123	Microsoft Corporation	05/05/2019
Microsoft Visual C++ 2015 Redistributable (x86) - 14.0.23506	Microsoft Corporation	16/02/2019
Microsoft Visual Studio 2015 Shell (Isolated)	Microsoft Corporation	19/02/2019
Microsoft Visual Studio Tools for Applications 2015	Microsoft Corporation	19/02/2019

Microsoft Visual C++ installed modules

2.1.5 Eurostat Web Service

- **MA API WS** version 8.4.1 with MSDB in version 6.15
- **NSI WS** version 8.4.1 or higher supporting write mode, with the following characteristics:
  - MSDB version 6.15
  - AUTHDB version 1.0
  - Authentication through AUTHDB
  - Possibility to modify not final attributes and annotations of final artefacts
- **NSI WS** to support read mode also version lower than 8.4.1.

The current release package contains the following versions of the web services:

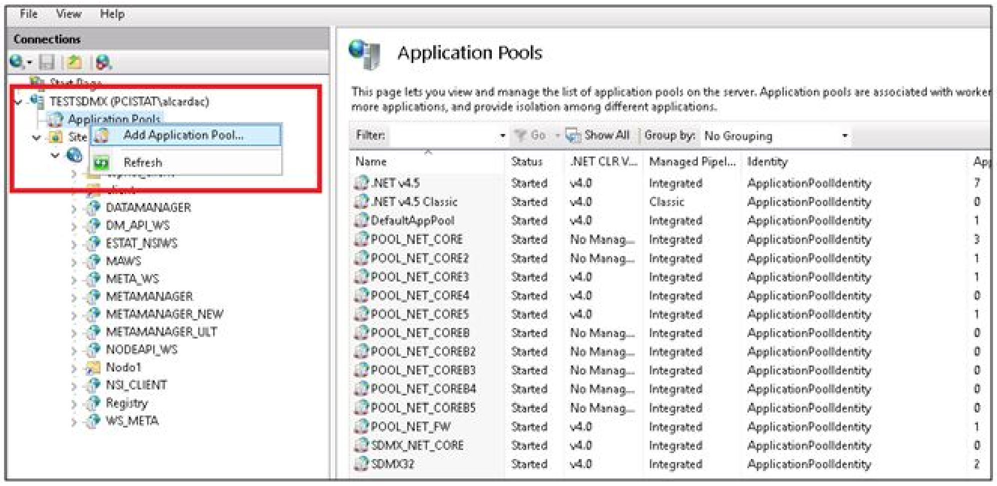
- **NSI WS**: 8.4.1
- **MA WS**: 8.4.1

2.1.6 SSL certificate

In order to publish the web services in https, an SSL certificate is needed. The instructions for creating that certificate depend on the certificate type and on the IIS version. For IIS ver.10 the instructions available at this URL can be followed: https://www.digicert.com/csr-creation-ssl-installation-iis-10.htm.

2.2 IIS configuration

First of all, the creation of an application pool for each .NET Core application is needed. It is possible to create a new application pool by right-clicking on "Application Pool" (see Fig. *Application pool IIS for .Net Core*) and by clicking on *'Add Application Pool'* Item).



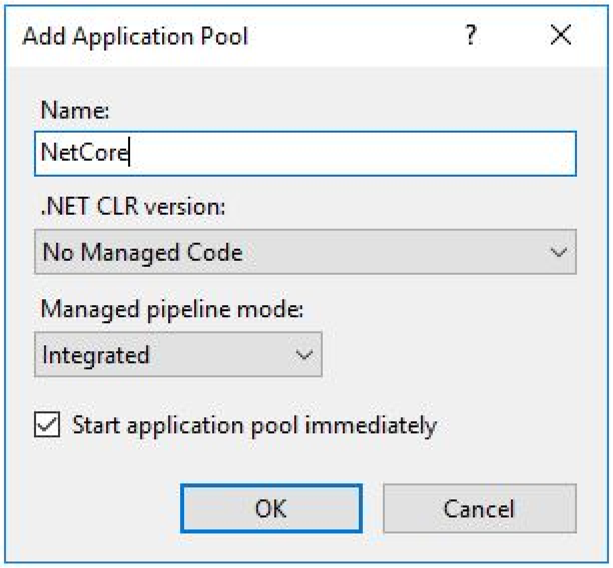
This page lets you view and manage the list of application pools on the server. Application pools are associated with worker
more applications, and provide isolation among different applications.

Name	Status	.NET CLR V...	Managed Pipel...	Identity	Ap
.NET v4.5	Started	v4.0	Integrated	ApplicationPoolIdentity	7
.NET v4.5 Classic	Started	v4.0	Classic	ApplicationPoolIdentity	0
DefaultAppPool	Started	v4.0	Integrated	ApplicationPoolIdentity	1
POOL_NET_CORE	Started	No Manag...	Integrated	ApplicationPoolIdentity	3
POOL_NET_CORE2	Started	No Manag...	Integrated	ApplicationPoolIdentity	1
POOL_NET_CORE3	Started	v4.0	Integrated	ApplicationPoolIdentity	1
POOL_NET_CORE4	Started	v4.0	Integrated	ApplicationPoolIdentity	0
POOL_NET_CORE5	Started	v4.0	Integrated	ApplicationPoolIdentity	1
POOL_NET_COREB	Started	No Manag...	Integrated	ApplicationPoolIdentity	0
POOL_NET_COREB2	Started	No Manag...	Integrated	ApplicationPoolIdentity	0
POOL_NET_COREB3	Started	No Manag...	Integrated	ApplicationPoolIdentity	0
POOL_NET_COREB4	Started	No Manag...	Integrated	ApplicationPoolIdentity	0
POOL_NET_COREB5	Started	No Manag...	Integrated	ApplicationPoolIdentity	0
POOL_NET_FW	Started	v4.0	Integrated	ApplicationPoolIdentity	1
SDMX_NET_CORE	Started	v4.0	Integrated	ApplicationPoolIdentity	0
SDMX32	Started	v4.0	Integrated	ApplicationPoolIdentity	2

Add a new application pool in IIS

The pools that must be created are the following:

- 5 application pools with .NET CLR *"No managed code"* Version for the web services implemented in .NET Core (e.g. named **POOL_NSIWS, POOL_MAWS, POOL_DMWS, POOL_NODEWS, POOL_METAWS**).



Name: NetCore

.NET CLR version: No Managed Code

Managed pipeline mode: Integrated

☑ Start application pool immediately

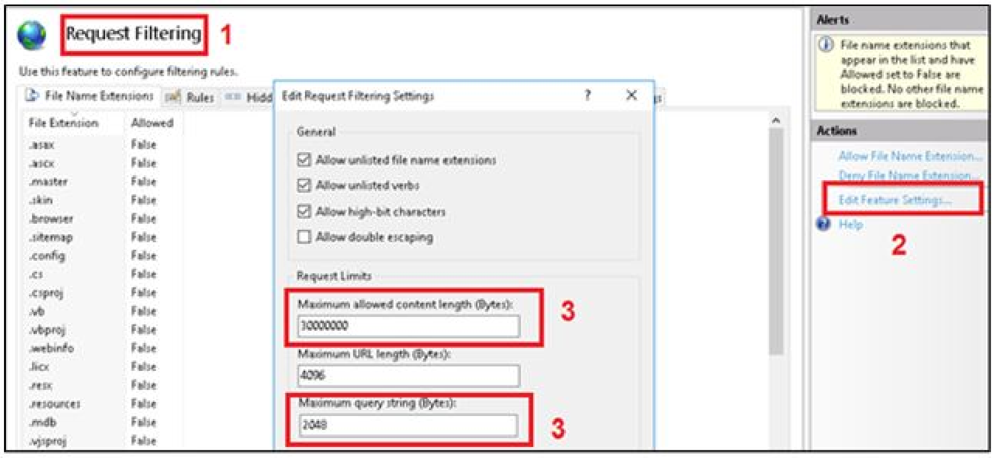
Application pool IIS for .Net Core

2.2.1 Parameters for the management of big files

To perform the uploading of big size files you have to set some parameters in order to avoid that IIS can generate time-out.

Maximum allowed length and Max Query String for the content

1. Click on the IIS website under which the application has to be installed;
2. Double click on the Requests *filtering* menu item;
3. Click on *"Edit feature settings"*;
4. Modify the *Maximum allowed content length (byte)* and *MaxQueryString* to the desired values.


Request Filtering
1
Use this feature to configure filtering rules.
File Name Extensions
Rules
Hidd
Edit Request Filtering Settings
Alerts
Actions
Allow File Name Extension...
Deny File Name Extension...
Edit Feature Settings...
Help
2
3
3

File name extensions that appear in the list and have Allowed set to False are blocked. No other file name extensions are blocked.

File Extension	Allowed
.asax	False
.ascx	False
.master	False
.skin	False
.browser	False
.sitemap	False
.config	False
.cs	False
.csproj	False
.vb	False
.vbproj	False
.webinfo	False
.licx	False
.resx	False
.resources	False
.mdb	False
.vjsproj	False

General

- [✓] Allow unlisted file name extensions
- [✓] Allow unlisted verbs
- [✓] Allow high-bit characters
- [ ] Allow double escaping

Request Limits

Maximum allowed content length (Bytes): 30000000

Maximum URL length (Bytes): 4096

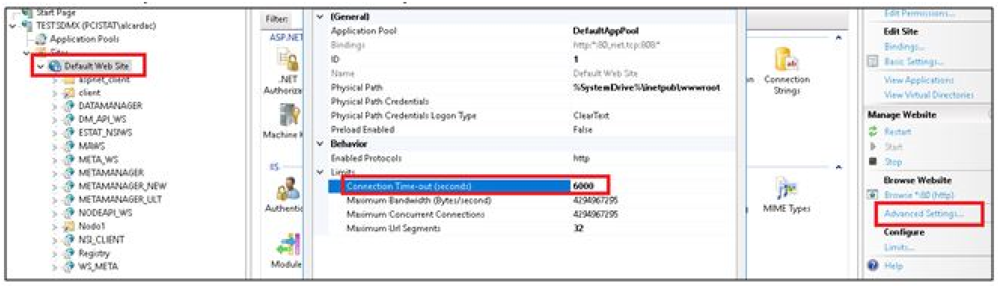
Maximum query string (Bytes): 2048

Editing of the "Maximum Allowed Content Length" parameter

Connection Time-out

The Connection Timeout parameter has to be set in order to allow delayed responses by the web services. The suggested value for this parameter is 6000 seconds (100 minutes).

1. Click on the IIS website under which the application has to be installed;
2. Select the *Advanced Settings menu*;
3. Click on *Limits*;
4. Modify the *Connection Time-out* parameter to the desired value.



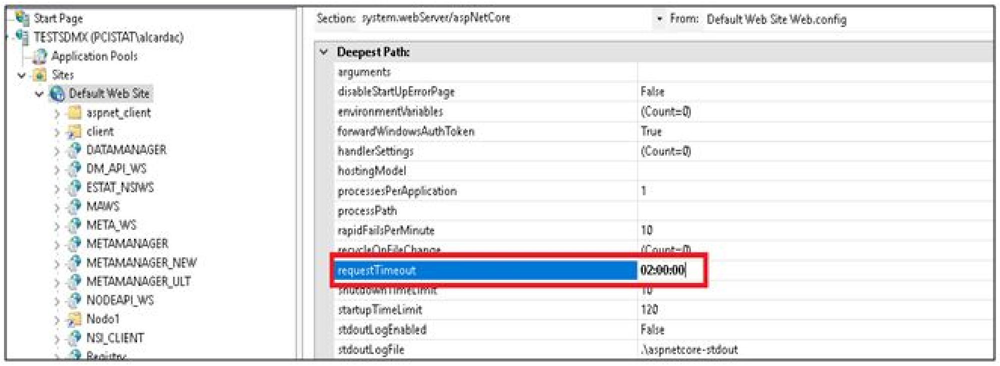
(General)	
Application Pool	DefaultAppPool
Bindings	http:*:80:,net.tcp:808:*
ID	1
Name	Default Web Site
Physical Path	%SystemDrive%\inetpub\wwwroot
Physical Path Credentials	
Physical Path Credentials Logon Type	ClearText
Preload Enabled	False
Behavior	
Enabled Protocols	http
Limits	
Connection Time-out (seconds)	6000
Maximum Bandwidth (Bytes/second)	4294967295
Maximum Concurrent Connections	4294967295
Maximum Url Segments	32

Editing of the Connection Timeout parameter

Request Time-out

This parameter allows to increase the time interval after which a time-out error is launched (blocking the execution) during the waiting of a response by a web service. It is suggested to increase this parameter to 120 minutes.

1. Click on the web site
2. Select Configuration Editor
3. Access to the *system.webServer/aspNetCore section*
4. Modify the *requestTimeout* parameter



Section: system.webServer/aspNetCore From: Default Web Site Web.config

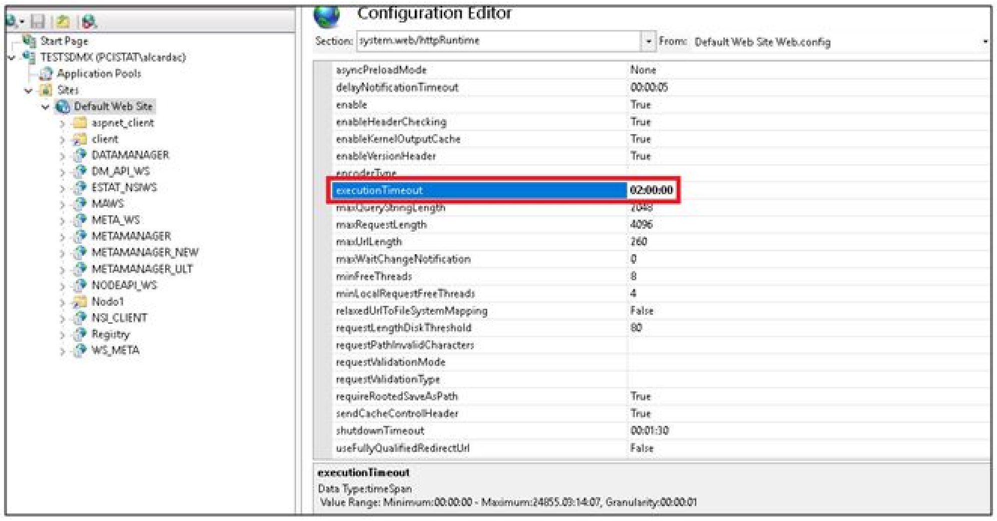
Deepest Path:	
arguments	
disableStartUpErrorPage	False
environmentVariables	(Count=0)
forwardWindowsAuthToken	True
handlerSettings	(Count=0)
hostingModel	
processesPerApplication	1
processPath	
rapidFailsPerMinute	10
recycleOnFileChange	(Count=0)
requestTimeout	02:00:00
shutdownTimeLimit	10
startupTimeLimit	120
stdoutLogEnabled	False
stdoutLogFile	.\aspnetcore-stdout

Editing of the requestTimeout parameter

Execution Time-out

This parameter, similar to the previous, allows to increase the time after which a timeout is launched (blocking the execution) after the execution of a web service that doesn't modify its execution status. It is suggested to increase this parameter to 120 minutes.

1. Click on the web site
2. Select *Configuration Editor*
3. Access to the *system.web/httpRuntime* section
4. Modify the *executionTimeout* parameter



Section: system.web/httpRuntime From: Default Web Site Web.config

asyncPreloadMode	None
delayNotificationTimeout	00:00:05
enable	True
enableHeaderChecking	True
enableKernelOutputCache	True
enableVersionHeader	True
encoderType	
executionTimeout	02:00:00
maxQueryStringLength	2048
maxRequestLength	4096
maxUrlLength	260
maxWaitChangeNotification	0
minFreeThreads	8
minLocalRequestFreeThreads	4
relaxedUrlToFileSystemMapping	False
requestLengthDiskThreshold	80
requestPathInvalidCharacters	
requestValidationMode	
requestValidationType	
requireRootedSaveAsPath	True
sendCacheControlHeader	True
shutdownTimeout	00:01:30
useFullyQualifiedRedirectUrl	False

executionTimeout
Data Type:timeSpan
Value Range: Minimum:00:00:00 - Maximum:24855.03:14:07, Granularity:00:00:01

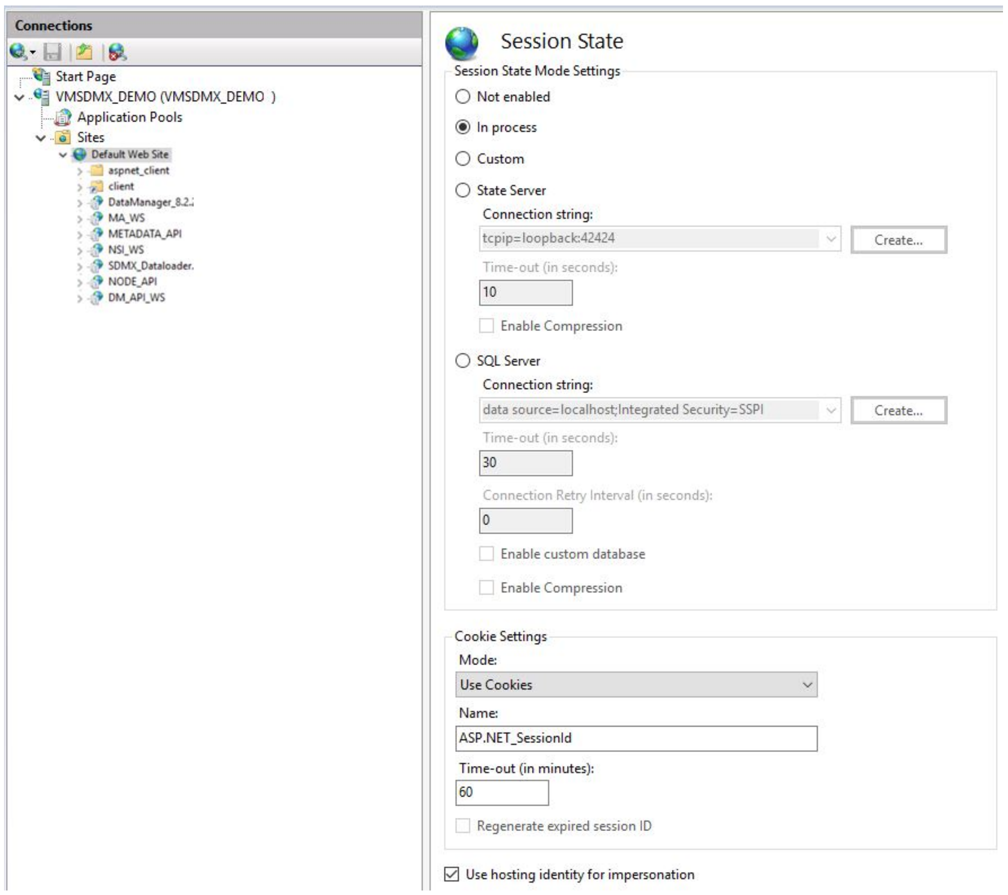
Editing the executionTimeout parameter

2.2.2 Other parameters

Session state

In order to increase the application session duration, the Session State parameter has to be set. It allows the maintenance of the session cookies without forcing users to re-login to the application.

1. In IIS manager, click on the "Default Web Site"
2. Click on the menu *"Session State"*
3. Set the option *TimeOut* (in minutes) to a suitable value (e.g. 60 minutes)



Session State Mode Settings

- ○ Not enabled
- ● In process
- ○ Custom
- ○ State Server
  - Connection string: tcpip=loopback:42424
  - Time-out (in seconds): 10
  - ☐ Enable Compression
- ○ SQL Server
  - Connection string: data source=localhost;Integrated Security=SSPI
  - Time-out (in seconds): 30
  - Connection Retry Interval (in seconds): 0
  - ☐ Enable custom database
  - ☐ Enable Compression

Cookie Settings

- Mode: Use Cookies
- Name: ASP.NET_SessionId
- Time-out (in minutes): 60
- ☐ Regenerate expired session ID

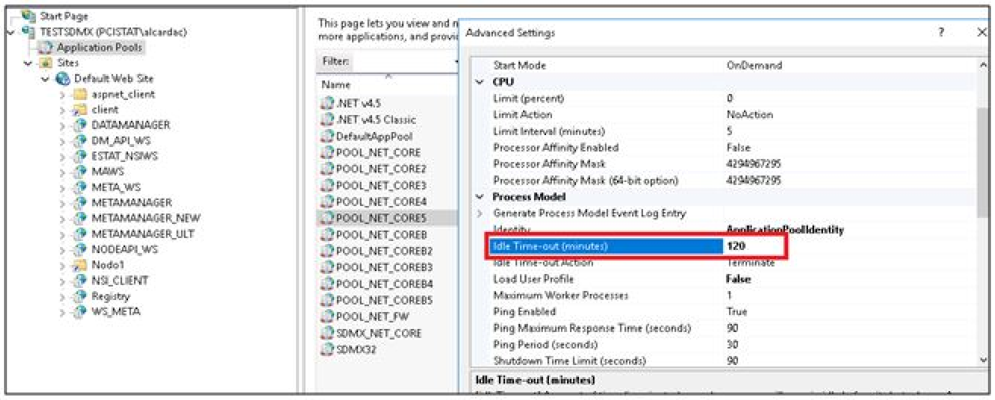
☑ Use hosting identity for impersonation

Session state Time-out parameter

Idle Time-out

This parameter determines the time after which an idle web service is stopped. It allows to eliminate the waiting time for restarting the web service in case of a very long session. It has to be set for each pool involved in long duration tasks.

1. Click on the pool
2. Select *Advanced Settings*
3. Modify the *Idle TimeOut* parameter i.e. by setting it to 120 minutes



Advanced Settings

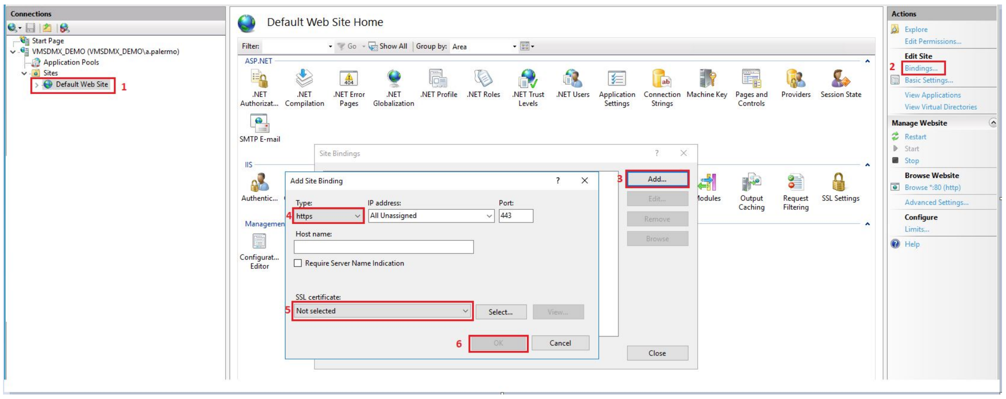
```
Start Mode                                   OnDemand
CPU
  Limit (percent)                            0
  Limit Action                               NoAction
  Limit Interval (minutes)                   5
  Processor Affinity Enabled                 False
  Processor Affinity Mask                    4294967295
  Processor Affinity Mask (64-bit option)    4294967295
Process Model
  Generate Process Model Event Log Entry
  Identity                                   ApplicationPoolIdentity
  Idle Time-out (minutes)                    120
  Idle Time-out Action                       Terminate
  Load User Profile                          False
  Maximum Worker Processes                   1
  Ping Enabled                               True
  Ping Maximum Response Time (seconds)       90
  Ping Period (seconds)                      30
  Shutdown Time Limit (seconds)              90
Idle Time-out (minutes)
```

Idle Time-Out parameter

2.2.3 Https Binding

It is needed to create a binding for https. The task can be performed as follows:

1. Click on "Default Web Site"
2. Click on Binding in the "Actions" menu on the top-right
3. Click on "Add"
4. Select https as type
5. Select an available SSL certificate
6. Click on "OK"



Add Site Binding

Type: 4 https
IP address: All Unassigned
Port: 443
Host name:
☐ Require Server Name Indication
SSL certificate: 5 Not selected
Select... View...
6 OK Cancel

Https binding configuration

2.3 Software package

The software is released as a zip file (e.g. MDM_V0.5.3_18-11-2019.zip), so it must be unzipped.
Files are organized in a hierarchical schema with the root named *"main"*:

- **app**: contains all the executable files that must be installed under IIS
  - **client**: contains the files for the client module
    - *static*

        - css: contains .css files
        - js: contanins .js files
        - locales: contains files for the localization of the GUI
        - media: contains graphical objects
        - png: contains files in .png format
        - config.json: client's configuration file
        - favicon.ico: application's favicon
    - *index.html*: the home page for the client
  - **ws**: contains all the WSs
    - *DM_API_WS*: contains the files for the Data Manager API
      - config: contains the log configuration file and the scripts to create the databases (you don't need to manually run the scripts)
      - runtime: contains all the .dll files
      - appsetting.json: it is the configuration file of the Web Service
    - *MA_WS*: contains the files for the Mapping Assistant API
      - AppData: contains the configuration files
      - bin: contains all the .dll files
      - doc: contains the documentation of the Web Service
      - script: contains the .js files
      - style: contains the .css files
      - log4net.config: log file configuration
      - web.config: file to configure the web service
    - *NODE_API*: contains the files for the NodeApi WS (the backend of the client)
      - config: contains the file to configure the Web Service
      - runtime: contains all the .dll files
      - appsetting.json: web service configuration file
    - *NSI_WS*: contains all the files of the NSI WS
    - *METADATA_API*: contains the files for the METADATA API (to manage Reference metadata and Digital catalogs)
      - config: configuration files for the Data Providers
      - resources: contains the resource files
      - runtime: contains all the .ddl files
      - appsetting.json: web service configuration file
- **doc**: contains the documentation (only the installation manual for now)
- **files**: contains some supporting files
  - **ReferenceMetadata**: folder containing files with default artefacts that describe a catalog base on the DCAT application profile for Italy

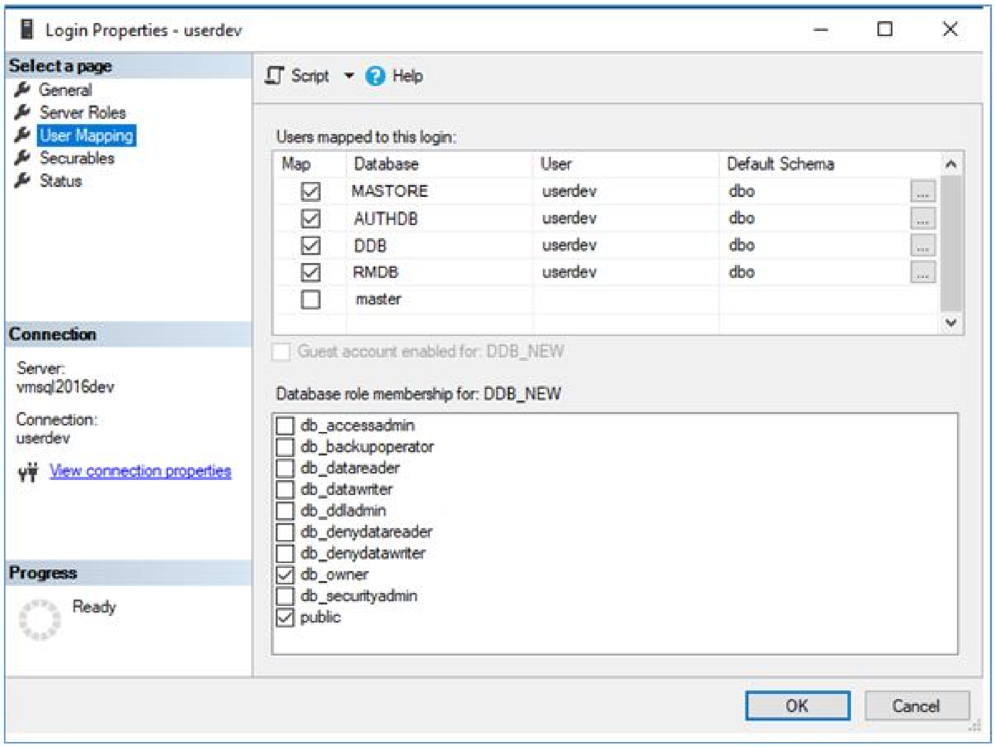
  - **Estat.SdmxSource.Extension.RDFPlugin.dll**: contains the plugin to download in RDF format
- **source**: contains the source code of the application
  - **client**: contains the source code of the client
  - **server**: contains the source code of the server APIs

2.4 Building databases

The following databases must be created:

- **AUTHDB**: create a new database named **AUTHDB**;
- **MSDB**: create a new database named **MSDB**;
- **DDB**: create a new database named **DDB**;
- **RMDB**: create a new database named **RMDB**.

The initializations of the Databases will be explained in the chapter *Initialization of the database*.
The users that will be used in the connection strings must have grants of read/write/execution on the above databases:



Server:
vmsql2016dev

Connection:
userdev

Users mapped to this login:

Map	Database	User	Default Schema
✓	MASTORE	userdev	dbo
✓	AUTHDB	userdev	dbo
✓	DDB	userdev	dbo
✓	RMDB	userdev	dbo
☐	master		

☐ Guest account enabled for: DDB_NEW

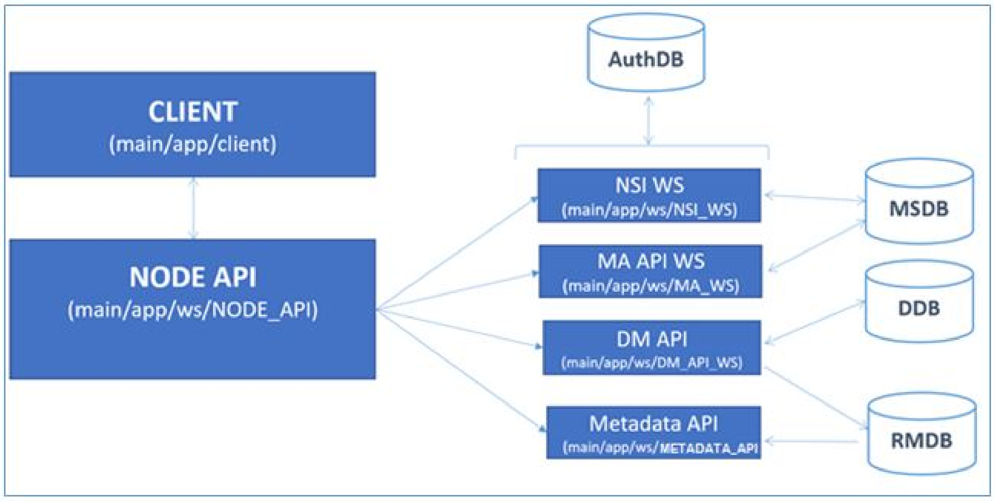
Database role membership for: DDB_NEW

- ☐ db_accessadmin
- ☐ db_backupoperator
- ☐ db_datareader
- ☐ db_datawriter
- ☐ db_ddladmin
- ☐ db_denydatareader
- ☐ db_denydatawriter
- ✓ db_owner
- ☐ db_securityadmin
- ✓ public

DB user mapping

2.5 Web Services deployment

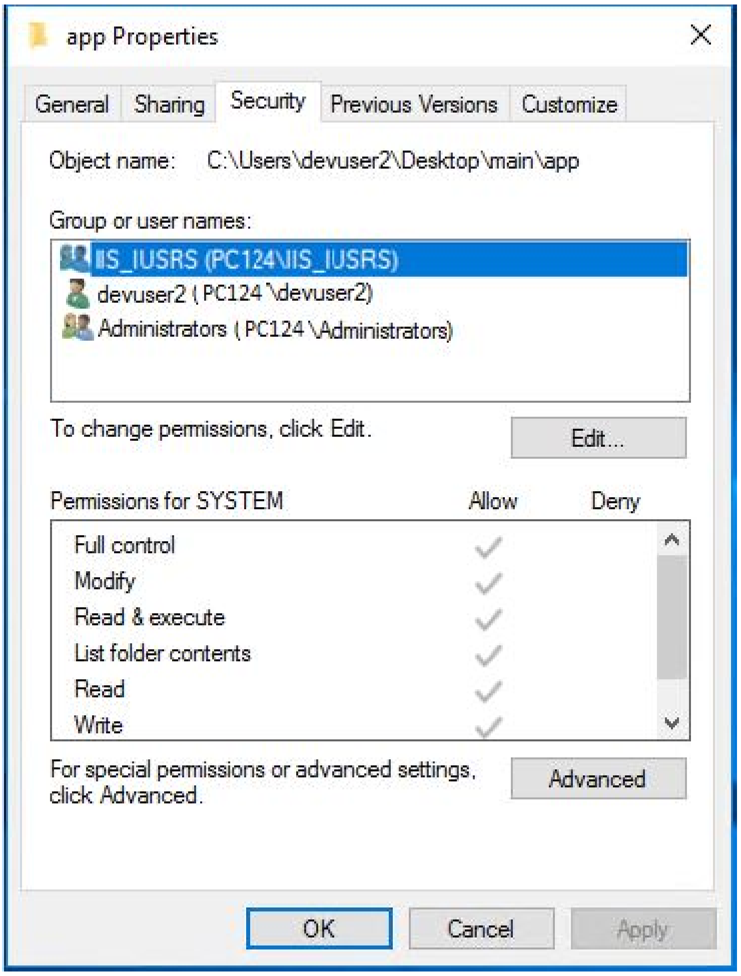
The software supports the following architecture:



Software architecture

The web services can be installed on the same Web Site of IIS.
The users IIS_IUSRS e IUSR must have the suitable permissions on the web applications, therefore on the folder *main/app*:

- right click on the folder;
- select *Property/Security*;
- click on *Edit/Add*;
- in the section *"Locations"*, select the local computer;
- in the section +*"Enter the object name to select"* write IIS_IUSRS;
- click on *"check names"* and then OK;
- in the section *"Permission for IIS_IUSRS"* include "full control";
- repeat steps from 3 to 6 for user IUSR;
- in the section *"Permission for IIS_IUSRS"* include *"write/read"* permissions.


app Properties

General | Sharing | Security | Previous Versions | Customize

Object name: C:\Users\devuser2\Desktop\main\app

Group or user names:

- IIS_IUSRS (PC124\IIS_IUSRS)
- devuser2 (PC124 \devuser2)
- Administrators (PC124 \Administrators)

To change permissions, click Edit. Edit...

Permissions for SYSTEM	Allow	Deny
Full control	✓	
Modify	✓	
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write	✓	

For special permissions or advanced settings, click Advanced. Advanced

OK Cancel Apply

Modify the permissions for the app folder

The following Web services must be installed:

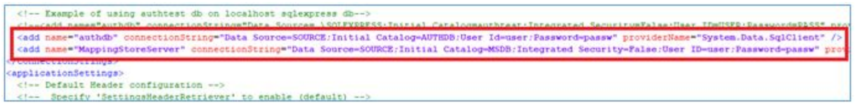
- *NSI WS*
- *MA API WS*
- *DM API WS*
- *Metadata API*
- *Node API*

In the next paragraphs the installation and configuration of the web service will be explained.

2.5.1 NSI WS

The installation and configuration of the NSI WS consists of the following steps:

1. In IIS manager, right click on *Default WebSite* and then *“Add application”*
2. Insert the alias **NSI_WS**, and associate the Application Pool **POOL_NSIWS**. Select for the physical path the folder *main/app/ws/NSI_WS*.
3. Modify the file *main/app/ws/NSI_WS/config/app.config* in order to configure the connection strings for MSDB and AuthDB databases.

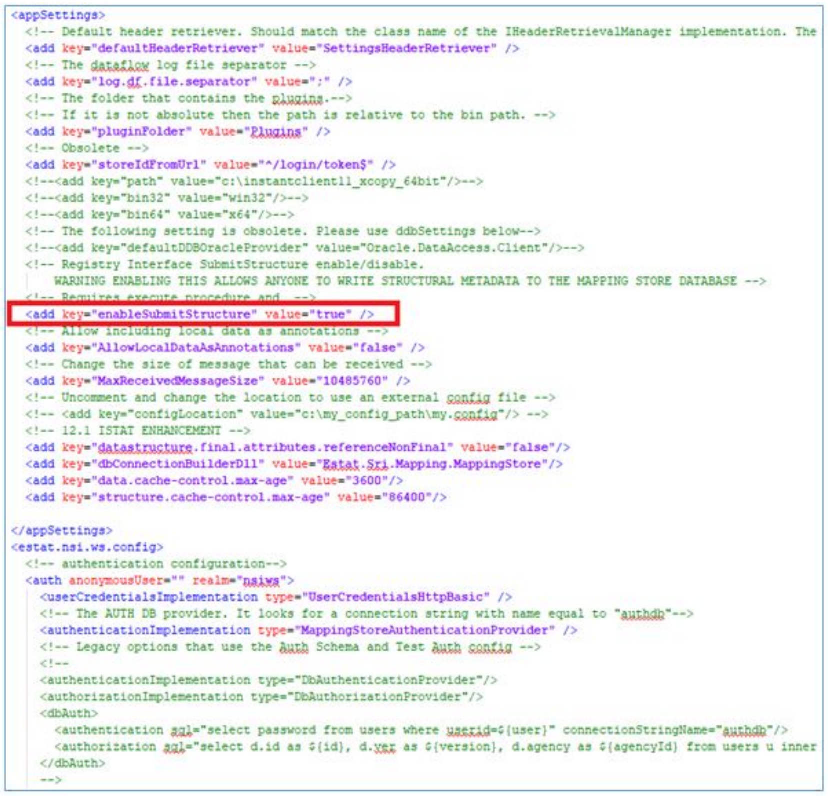


```
  <!-- Example of using authtest db on localhost sqlexpress db-->
  <!--<add name="authdb" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=authtest;Integrated Security=False;User ID=USER;Password=PASS" pro
  <add name="authdb" connectionString="Data Source=SOURCE;Initial Catalog=AUTHDB;User Id=user;Password=passw" providerName="System.Data.SqlClient" />
  <add name="MappingStoreServer" connectionString="Data Source=SOURCE;Initial Catalog=MSDB;Integrated Security=False;User ID=user;Password=passw" pro
</connectionStrings>
<applicationSettings>
  <!-- Default Header configuration -->
  <!--  Specify 'SettingsHeaderRetriever' to enable (default) -->
```

Set the connection strings for MSDB and AUTHDB databases

4. Set the following parameters as follows:

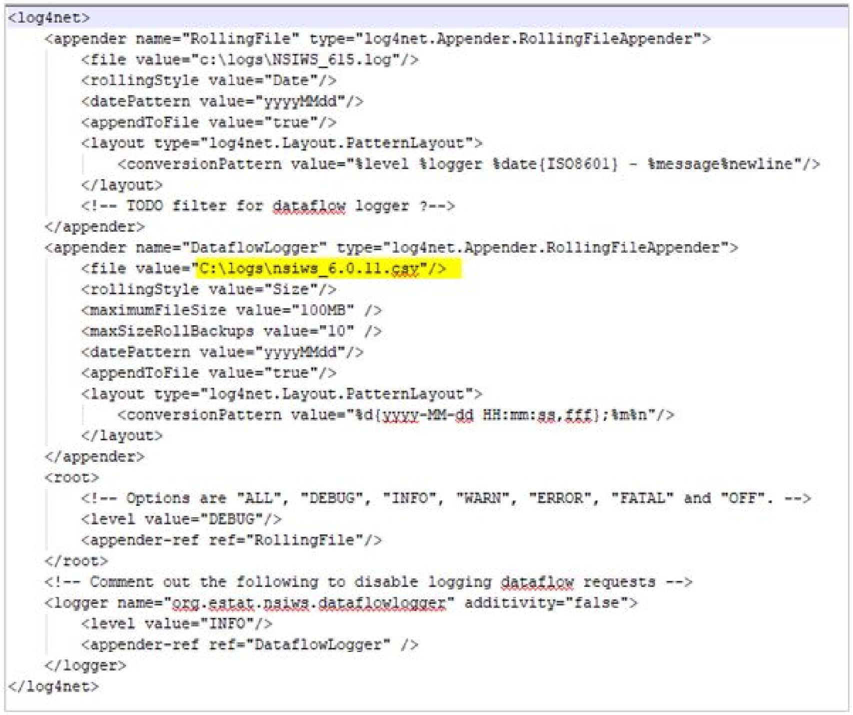
   1. *"enableSubmitStructure"* = "true"
   2. *"ignoreProductionFlagForStructure"* = "true"
   3. *"InsertNewItems"* = "true"



```
<appSettings>
  <!-- Default header retriever. Should match the class name of the IHeaderRetrievalManager implementation. The
  <add key="defaultHeaderRetriever" value="SettingsHeaderRetriever" />
  <!-- The dataflow log file separator -->
  <add key="log.df.file.separator" value=";" />
  <!-- The folder that contains the plugins.-->
  <!-- If it is not absolute then the path is relative to the bin path. -->
  <add key="pluginFolder" value="Plugins" />
  <!-- Obsolete -->
  <add key="storeIdFromUrl" value="^/login/token$" />
  <!--<add key="path" value="c:\instantclient11_xcopy_64bit"/>-->
  <!--<add key="bin32" value="win32"/>-->
  <!--<add key="bin64" value="x64"/>-->
  <!-- The following setting is obsolete. Please use ddbSettings below-->
  <!--<add key="defaultDDBOracleProvider" value="Oracle.DataAccess.Client"/>-->
  <!-- Registry Interface SubmitStructure enable/disable.
      WARNING ENABLING THIS ALLOWS ANYONE TO WRITE STRUCTURAL METADATA TO THE MAPPING STORE DATABASE -->
  <!-- Requires execute procedure and  -->
  <add key="enableSubmitStructure" value="true" />
  <!-- Allow including local data as annotations -->
  <add key="AllowLocalDataAsAnnotations" value="false" />
  <!-- Change the size of message that can be received -->
  <add key="MaxReceivedMessageSize" value="10485760" />
  <!-- Uncomment and change the location to use an external config file -->
  <!-- <add key="configLocation" value="c:\my_config_path\my.config"/> -->
  <!-- 12.1 ISTAT ENHANCEMENT -->
  <add key="datastructure.final.attributes.referenceNonFinal" value="false"/>
  <add key="dbConnectionBuilderDll" value="Estat.Sri.Mapping.MappingStore"/>
  <add key="data.cache-control.max-age" value="3600"/>
  <add key="structure.cache-control.max-age" value="86400"/>

</appSettings>
<estat.nsi.ws.config>
  <!-- authentication configuration-->
  <auth anonymousUser="" realm="nsiws">
    <userCredentialsImplementation type="UserCredentialsHttpBasic" />
    <!-- The AUTH DB provider. It looks for a connection string with name equal to "authdb"-->
    <authenticationImplementation type="MappingStoreAuthenticationProvider" />
    <!-- Legacy options that use the Auth Schema and Test Auth config -->
    <!--
    <authenticationImplementation type="DbAuthenticationProvider"/>
    <authorizationImplementation type="DbAuthorizationProvider"/>
    <dbAuth>
      <authentication sql="select password from users where userid=${user}" connectionStringName="authdb"/>
      <authorization sql="select d.id as ${id}, d.ver as ${version}, d.agency as ${agencyId} from users u inner
    </dbAuth>
    -->
```

A section of the file app.config

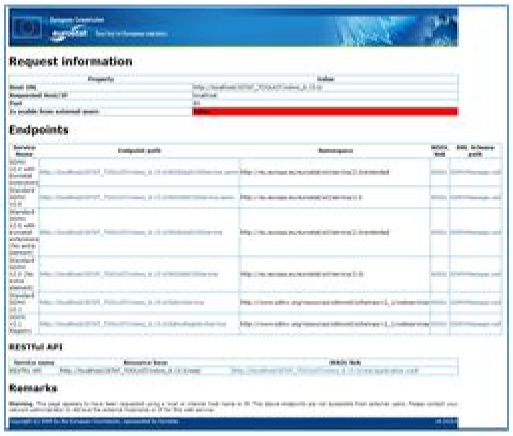
5. Modify the file *main/app/ws/NSI_WS/log4net.config* to configure del log.



```
<log4net>
    <appender name="RollingFile" type="log4net.Appender.RollingFileAppender">
        <file value="c:\logs\NSIWS_615.log"/>
        <rollingStyle value="Date"/>
        <datePattern value="yyyyMMdd"/>
        <appendToFile value="true"/>
        <layout type="log4net.Layout.PatternLayout">
            <conversionPattern value="%level %logger %date{ISO8601} - %message%newline"/>
        </layout>
        <!-- TODO filter for dataflow logger ?-->
    </appender>
    <appender name="DataflowLogger" type="log4net.Appender.RollingFileAppender">
        <file value="C:\logs\nsiws_6.0.11.csv"/>
        <rollingStyle value="Size"/>
        <maximumFileSize value="100MB" />
        <maxSizeRollBackups value="10" />
        <datePattern value="yyyyMMdd"/>
        <appendToFile value="true"/>
        <layout type="log4net.Layout.PatternLayout">
            <conversionPattern value="%d{yyyy-MM-dd HH:mm:ss,fff};%m%n"/>
        </layout>
    </appender>
    <root>
        <!-- Options are "ALL", "DEBUG", "INFO", "WARN", "ERROR", "FATAL" and "OFF". -->
        <level value="DEBUG"/>
        <appender-ref ref="RollingFile"/>
    </root>
    <!-- Comment out the following to disable logging dataflow requests -->
    <logger name="org.estat.nsiws.dataflowlogger" additivity="false">
        <level value="INFO"/>
        <appender-ref ref="DataflowLogger" />
    </logger>
</log4net>
```

A section of the file log4net.config

6. Check the configuration using the following request: *https://localhost/NSI_WS*. The web service helper of the web service is shown.



Request information

Endpoints

RESTful API

Remarks

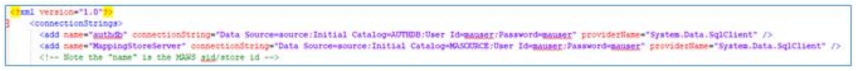
The NSI WS helper

7. The request *https://localhost/NSI_WS/rest/codelist* will answer with a set of code lists that are installed by default.

2.5.2 MA API WS

To install and configure the MA WS the following actions must be performed:

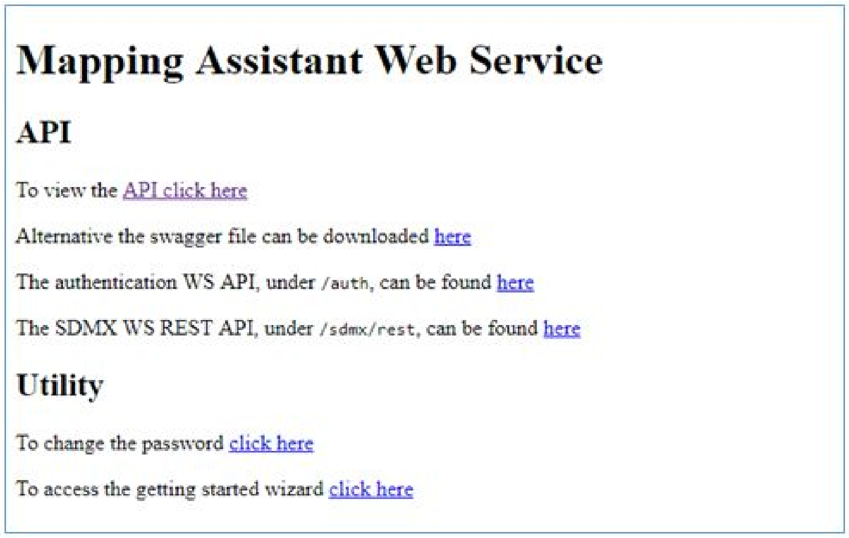
1. In IIS manager, right click on *Default WebSite* and then *"Add application"*
2. Insert the alias **MA_WS**, and associate the Application Pool **POOL_MAWS**. As physical path include the folder *main/app/ws/MA_WS*.
3. Modify the file *main/app/ws/MA_WS/AppData/ConnectionString* to configure the connection strings for the AuthDB e al MSDB databases.

   ```
   <?xml version="1.0"?>
     <connectionStrings>
       <add name="authdb" connectionString="Data Source=source;Initial Catalog=AUTHDB;User Id=mauser;Password=mauser" providerName="System.Data.SqlClient" />
       <add name="MappingStoreServer" connectionString="Data Source=source;Initial Catalog=MASOURCE;User Id=mauser;Password=mauser" providerName="System.Data.SqlClient" />
       <!-- Note the "name" is the MAWS sid/store id -->
   ```

   A section of the file ConnectionString.config

4. Modify the file *main/app/ws/MA_WS/log4net.config* to configure the log.
5. Check the configuration using the following request: *https://localhost/MA_WS*. The helper of the web service is shown.

   # Mapping Assistant Web Service

   ## API

   To view the API click here

   Alternative the swagger file can be downloaded here

   The authentication WS API, under `/auth`, can be found here

   The SDMX WS REST API, under `/sdmx/rest`, can be found here

   ## Utility

   To change the password click here

   To access the getting started wizard click here

   MA API WS helper

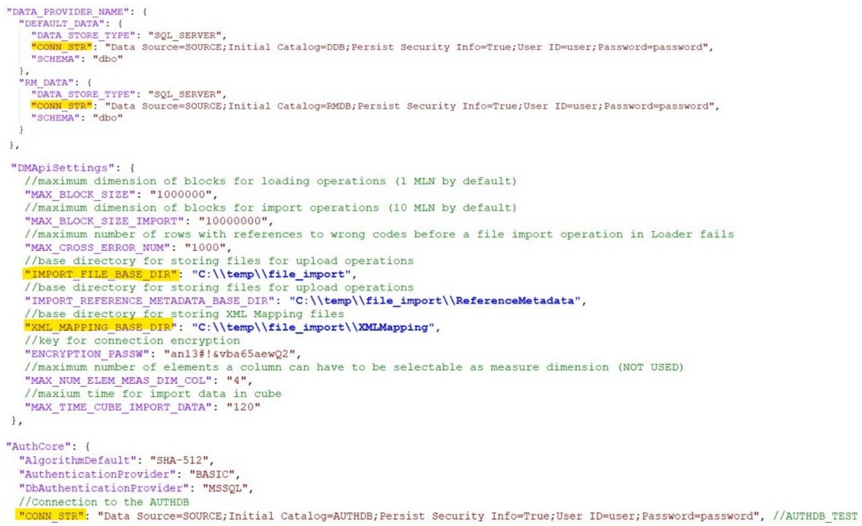
2.5.3 DM API WS

To install and configure the DM API WS the following actions must be performed:

1. In IIS manager, right click on Default WebSite and then "Add application"
2. Insert as alias DM_API_WS, and associate the Application Pool POOL_DMWS. As physical path specify the folder main/app/ws/DM_API_WS.
3. Modify the file main/app/ws/DM_API_WS/appsettings.json, and set the connection strings for AUTHDB, DDB and RMDB databases, through the following parameters:
   - AuthCore/CONN_STR
   - DATA_PROVIDER_NAME/DEFAULT_DATA/CONN_STR
   - DATA_PROVIDER_NAME/RM_DATA/CONN_STR

Pay attention to the escape characters (e.g. \\ for \)

4. Modify the path where the files will be stored during the uploading, using the DMApiSettings/IMPORT_FILE_BASE_DIR and DMApiSettings/XML_MAPPING_BASE_DIR parameters. IIS_IUSRS and IUSR users must have the permissions read/write on this folder.
   Pay attention to the escape character (es. \\ for \)

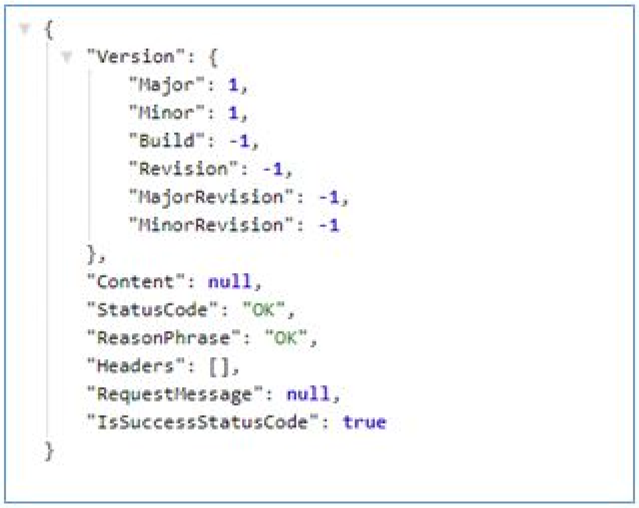


```
"DATA_PROVIDER_NAME": {
  "DEFAULT_DATA": {
    "DATA_STORE_TYPE": "SQL_SERVER",
    "CONN_STR": "Data Source=SOURCE;Initial Catalog=DDB;Persist Security Info=True;User ID=user;Password=password",
    "SCHEMA": "dbo"
  },
  "RM_DATA": {
    "DATA_STORE_TYPE": "SQL_SERVER",
    "CONN_STR": "Data Source=SOURCE;Initial Catalog=RMDB;Persist Security Info=True;User ID=user;Password=password",
    "SCHEMA": "dbo"
  }
},

 "DMApiSettings": {
   //maximum dimension of blocks for loading operations (1 MLN by default)
   "MAX_BLOCK_SIZE": "1000000",
   //maximum dimension of blocks for import operations (10 MLN by default)
   "MAX_BLOCK_SIZE_IMPORT": "10000000",
   //maximum number of rows with references to wrong codes before a file import operation in Loader fails
   "MAX_CROSS_ERROR_NUM": "1000",
   //base directory for storing files for upload operations
   "IMPORT_FILE_BASE_DIR": "C:\\temp\\file_import",
   //base directory for storing files for upload operations
   "IMPORT_REFERENCE_METADATA_BASE_DIR": "C:\\temp\\file_import\\ReferenceMetadata",
   //base directory for storing XML Mapping files
   "XML_MAPPING_BASE_DIR": "C:\\temp\\file_import\\XMLMapping",
   //key for connection encryption
   "ENCRYPTION_PASSW": "an13#!&vba65aewQ2",
   //maximum number of elements a column can have to be selectable as measure dimension (NOT USED)
   "MAX_NUM_ELEM_MEAS_DIM_COL": "4",
   //maxium time for import data in cube
   "MAX_TIME_CUBE_IMPORT_DATA": "120"
 },

"AuthCore": {
  "AlgorithmDefault": "SHA-512",
  "AuthenticationProvider": "BASIC",
  "DbAuthenticationProvider": "MSSQL",
  //Connection to the AUTHDB
  "CONN_STR": "Data Source=SOURCE;Initial Catalog=AUTHDB;Persist Security Info=True;User ID=user;Password=password", //AUTHDB_TEST
```

File appsettings.json

5. Modify the file *main/app/ws/DM_API_WS/config/base/logconfig.xml* to configure the log.

6. Check the web service through the request *https://localhost/DM_API_WS/api/DMApi/Ping*. If the web service is configured correctly, the version of the web service will be shown.



```
{
  "Version": {
    "Major": 1,
    "Minor": 1,
    "Build": -1,
    "Revision": -1,
    "MajorRevision": -1,
    "MinorRevision": -1
  },
  "Content": null,
  "StatusCode": "OK",
  "ReasonPhrase": "OK",
  "Headers": [],
  "RequestMessage": null,
  "IsSuccessStatusCode": true
}
```

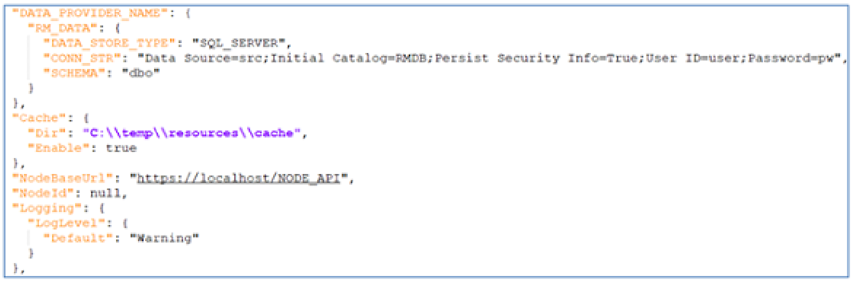
DM API WS is configured correctly

7. The request *https://localhost/DM_API_WS/swagger* shows the list of the methods exposed by the web service.

2.5.4 Metadata API

To install and configure the METADATA API WS the following actions must be performed:

1. In IIS manager, right click on *Default WebSite* and then *"Add application"*.
2. Insert as alias **METADATA_API**, and associate the Application Pool **POOL_METAWS**. As physical path select the folder *main/app/ws/METADATA_API*.
3. Modify the file *main/app/ws/METADATA_API/appsetting.json* and set the connection string for RMDB database.

   ```
   "DATA_PROVIDER_NAME": {
     "RM_DATA": {
       "DATA_STORE_TYPE": "SQL_SERVER",
       "CONN_STR": "Data Source=src;Initial Catalog=RMDB;Persist Security Info=True;User ID=user;Password=pw",
       "SCHEMA": "dbo"
     }
   },
   "Cache": {
     "Dir": "C:\\temp\\resources\\cache",
     "Enable": true
   },
   "NodeBaseUrl": "https://localhost/NODE_API",
   "NodeId": null,
   "Logging": {
     "LogLevel": {
       "Default": "Warning"
     }
   },
   ```

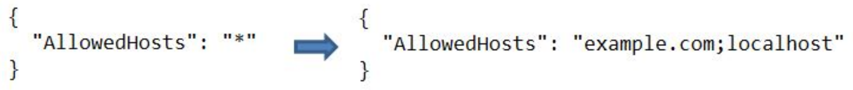
   Configuration of the METADATA API

4. In the same file, modify the value of the **NodeBaseUrl** key, so that it points to the NodeAPI endpoint, described in the next paragraph.

   Note also that if **NodeId** is valorized, its value is always used to contact the NodeAPI; this behavior is adopted for security reasons, that is to avoid that the client can request information of other nodes simply by passing a different NodeId as a parameter in the URLs. **The configuration of the module is detailed in the paragraph *2.10.8 Configurations and contexts Metadata API*.**

   In the same file, modify the value of the **Cache.Dir** key, so that it points to the path where the user wants cache files to be stored. The user must have read and write permissions on this path. Pay attention to the escape character (es. \\ for \) It is possible to enable or disable the use of the cache by modifying the **Cache.Enable** key with the value true or false, as shown in the previous paragraph. If cache is enabled, read data will be saved in the folder pointed by the Cache.Dir key, to speed up the application.

   If the application needs to be accessible by using a particular address, it is possible to modify the **AllowedHosts** key with a list of semicolon-delimited valid addresses without port numbers. For example:

   ```
   {
     "AllowedHosts": "*"
   }
   ```

   ```
   {
     "AllowedHosts": "example.com;localhost"
   }
   ```

   Configuration AllowedHosts of the METADATA API

   This way, the host filtering middleware does not allow the application to bind to any other hostname except for example.com or localhost.

5. Modify the file *main/app/ws/METADATA_API/config/base/logconfig.xml* to configure log.
6. Check the web service through the request *https://localhost/METADATA_API/Ping*. If the web service is configured

correctly it returns 'true'.

7. The request *https://localhost/METADATA_API/swagger* shows the list of the methods exposed by the web service.

2.5.5 Node API

To install and configure the NodeApi service, the following actions must be performed.

1. In IIS manager, right click on *Default WebSite* and then *"Add application"*
2. Insert as alias **NODE_API**, and associate the Application Pool **POOL_NODEWS**. As physical path specify the folder *main/app/ws/NODE_API*.
3. If the web service is installed correctly the request *https://localhost/NODE_API/swagger* shows the list of the methods exposed by the web service.
4. In the **appsettings.json** file it is also possible to customize the name of the SENDER present in the json and xml messages created by the data exports. Just add or configure the **EXPORT_RM_SDMX_SENDER_ID** entry as shown in the following example:

   ```
   {
     "EXPORT_RM_SDMX_SENDER_ID": "MDM_SENDER"
   }
   ```

5. In the **appsettings.json** file it is also possible to customize the name of the dataset agency and date format present in the xml messages created by the data exports. Just add or configure the **EXPORT_RM_SDMX_DATASET_AGENCY** and **GREGORIAN_DAY_FORMAT_SDMX_ML** entries as shown in the following example:

   ```
   {
     "EXPORT_RM_SDMX_DATASET_AGENCY": null,
     "GREGORIAN_DAY_FORMAT_SDMX_ML": "dd/MM/yyyy",
   }
   ```

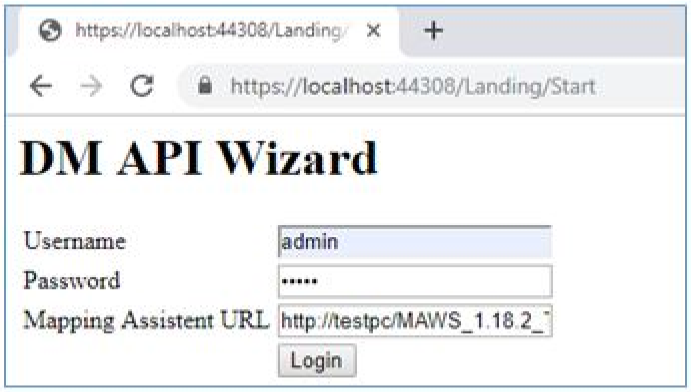
   In this case dataset agency (EXPORT_RM_SDMX_DATASET_AGENCY) is not specified, so the metadataflow agency will be used as the default value.

2.6 Database initialization

The initialization wizard at the URL: *<WebSiteBasePath of the DM API>/Wizard/Home*, for example *https://localhost/DM_API_WS/Wizard/Home*.
The initialization actions must be performed by a user with the permissions of administrator.

2.6.1 Login

In order to run the initialization wizard insert the l'url of the MA WS (e.g. *https://localhost/MA_WS*) and Username/Password of the Administrator (by default admin/[*empty string*]).


https://localhost:44308/Landing/
https://localhost:44308/Landing/Start

DM API Wizard

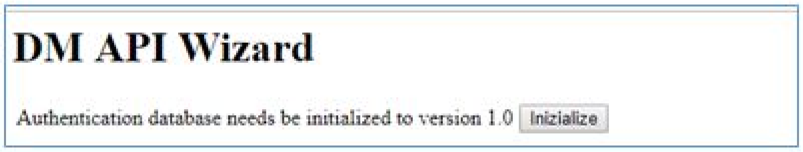
Username	admin
Password	•••••
Mapping Assistent URL	http://testpc/MAWS_1.18.2_

Installation Wizard - Login

2.6.2 AuthDB + Extensions MSDB

The steps that must be performed in case the AUTHDB is not still initialized are listed below.

1. Click on the button [Initialize] to initialize the AUTHDB database

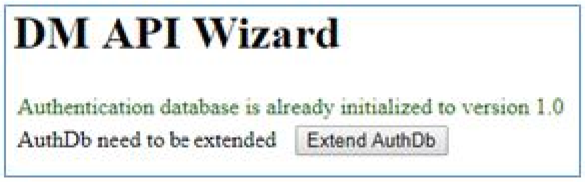


DM API Wizard

Authentication database needs be initialized to version 1.0

Installation Wizard – AuthDB initialization

2. The AUTHDB must be extended. Click on the the button [Extend AuthDB]



DM API Wizard

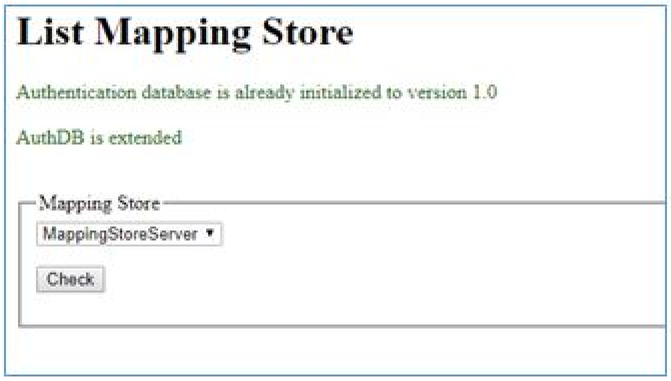
Authentication database is already initialized to version 1.0

AuthDb need to be extended

Installation Wizard – Extension of the AuthDB

2.6.3 Check of MSDB status

Select the MappingStoreServer and click on the button [Check]


List Mapping Store

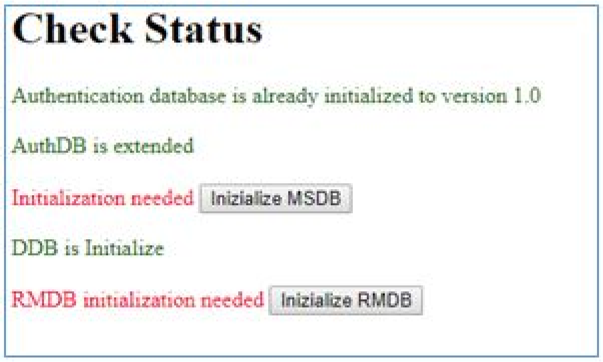
Authentication database is already initialized to version 1.0

AuthDB is extended

Status Wizard – Mapping Store

2.6.4 Check the status of MSDB, DDB and RMDB

In this screenshot the MSDB, DDB and RMDB are shown. The red color means that the database must be initialized. Click on the related button and initialize the databases one by one.


Check Status

Authentication database is already initialized to version 1.0

AuthDB is extended

Initialization needed

DDB is Initialize

RMDB initialization needed

Installation Wizard– initialization of the DBs

2.6.5 Disable the Wizard

After these steps have been done, for security reason we suggest to disable the access to the following wizard's methods. To do that you have to add the following emphasized lines in the file *main/app/ws/DM_API_WS/web.config*:

```
<configuration>
 <system.webServer>
  <security>
   <requestFiltering>
     <denyUrlSequences>
       <add sequence="DM_API_WS/Wizard/CheckMetaData"/>
       <add sequence="DM_API_WS/Wizard/Step01Login"/>
       <add sequence="DM_API_WS/Wizard/CheckEndPoint"/>
       <add sequence="DM_API_WS/Wizard/Start"/>
       <add sequence="DM_API_WS/Wizard/ChangePassword"/>
       </denyUrlSequences>
   </requestFiltering>
  </security>
```

(continues on next page)

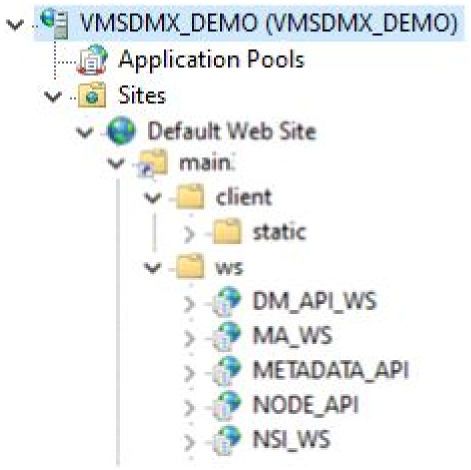
(continued from previous page)

```
  </system.webServer>
</configuration>
```

2.7 Client deployment

Below the steps to install and configure the client module.

1. In IIS manager, right click on Default WebSite and then "Add application"
2. Insert as alias **client**, and as physical path the folder main/app/client.



How IIS applications should appear

3. Modify the parameter **fetchBaseUrl** in the file *main/app/client/static/config.json*, providing the url of the NodeApi, for example *https://localhost/NODE_API*.

2.7.1 Reference metadata

For the correct operation of the client it will be necessary to verify/modify the reference Metadata/metadataapi.json file in the following way:

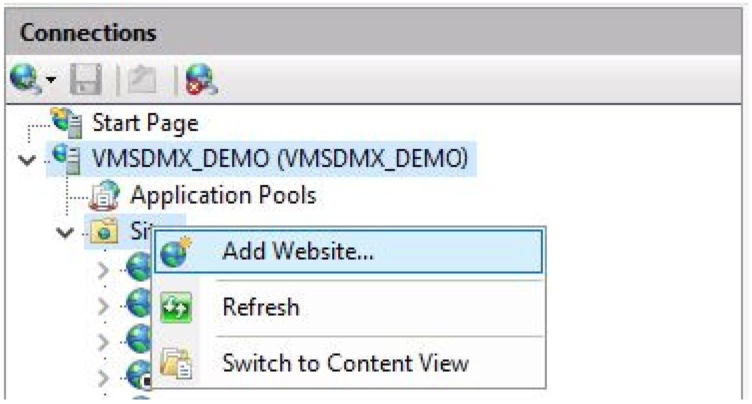
```
{
   "baseUrl": "<BASE URL METADATAAPI>"
}
```

- **baseUrl** key must contain the base url of the MetadataApi url.

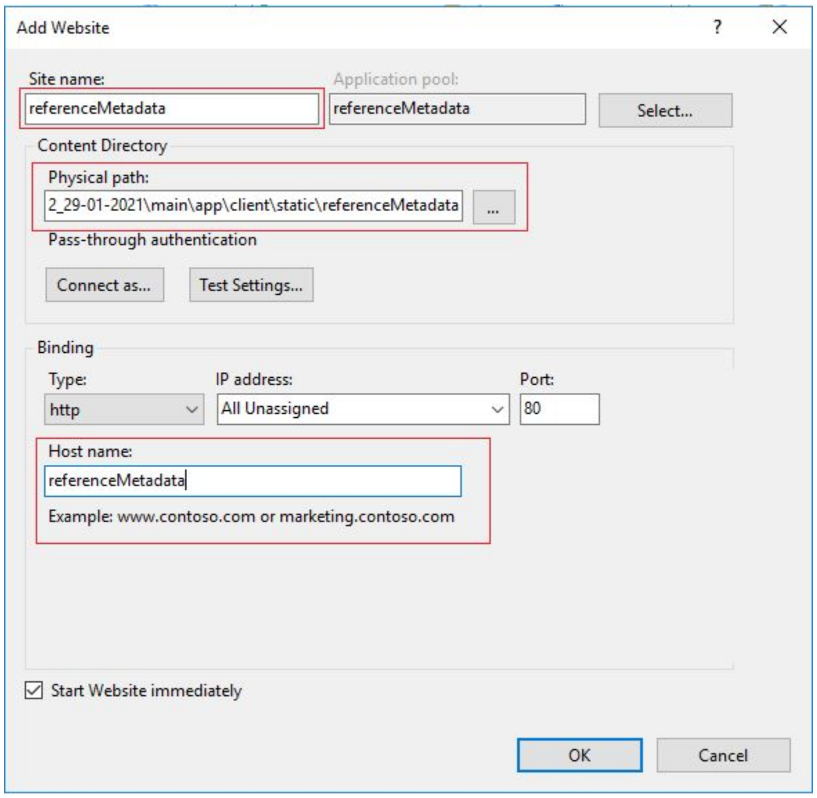
The configuration of the module is detailed in the paragraph *2.10.8 Configurations and contexts Metadata API*

2.7.1.1 Deploy Report HTML

The HTML components exposed and integrated into the client (discussed in the previous paragraphs), can also be used in standalone mode (without using the MDM client).
To do this, just deploy the folder **referenceMetadata**, into **../main/app/client/static/** (in the MDM client installation package).
To deploy in IIS, just open the tool manager and right click on the folder **Sites** (in the tree on the left) and click on **Add Website**:



A site configuration form will open like the following:



Site name: referenceMetadata
Application pool: referenceMetadata
Physical path: 2_29-01-2021\main\app\client\static\referenceMetadata
Type: http
IP address: All Unassigned
Port: 80
Host name: referenceMetadata
☑ Start Website immediately

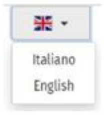
Where you need to fill in the fields highlighted in red in the figure.
Once the deployment is complete, to view the component that shows the CKAN catalogs, open the browser at the following url (replacing the placeholders indicated with <>):

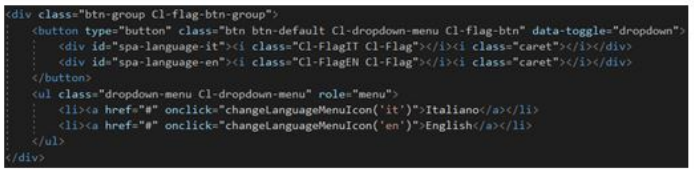
*http://referenceMetadata/CategoryTemplate.html?nodeId=<NODE_ID>&
metadataSetId=<ID_METADATASET_DCAT_AP_IT>&lang=&
BaseUrlMDA=<BASE_URL_METADATA_API>&BaseUrlMDM=<BASE_URL_CLIENT>*

2.7.1.2 Add language report HTML CKAN

To add a language to the template that displays the CKAN catalog data you need to edit the HTML page **CategoryTemplate.html** into **../main/app/client/static/referenceMetadata** (in the MDM client installation package). Here are the portions to be changed.
NB: HTML elements that contain language-based values have type values as id attribute **<HTML_ID>-<ABBREVIATION-LANGUAGE>** for example **spa-language-it**. This convention must be maintained when adding a new language.

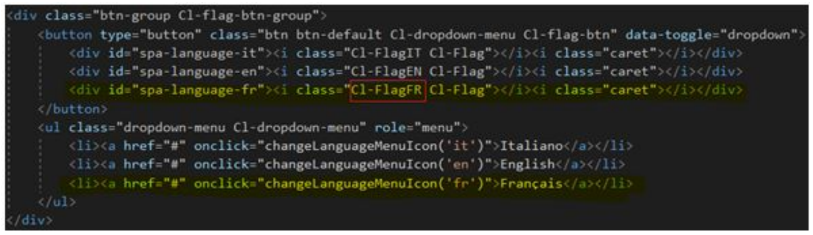
Language selector





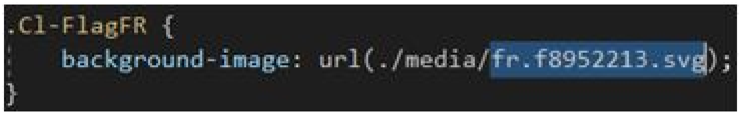
```
<div class="btn-group Cl-flag-btn-group">
    <button type="button" class="btn btn-default Cl-dropdown-menu Cl-flag-btn" data-toggle="dropdown">
        <div id="spa-language-it"><i class="Cl-FlagIT Cl-Flag"></i><i class="caret"></i></div>
        <div id="spa-language-en"><i class="Cl-FlagEN Cl-Flag"></i><i class="caret"></i></div>
    </button>
    <ul class="dropdown-menu Cl-dropdown-menu" role="menu">
        <li><a href="#" onclick="changeLanguageMenuIcon('it')">Italiano</a></li>
        <li><a href="#" onclick="changeLanguageMenuIcon('en')">English</a></li>
    </ul>
</div>
```

Add a **div** and a **li** respectively under the elements **button** and **ul** on the basis of those already present. For example to add the French language (fr) the code above will become:



```
<div class="btn-group Cl-flag-btn-group">
    <button type="button" class="btn btn-default Cl-dropdown-menu Cl-flag-btn" data-toggle="dropdown">
        <div id="spa-language-it"><i class="Cl-FlagIT Cl-Flag"></i><i class="caret"></i></div>
        <div id="spa-language-en"><i class="Cl-FlagEN Cl-Flag"></i><i class="caret"></i></div>
        <div id="spa-language-fr"><i class="Cl-FlagFR Cl-Flag"></i><i class="caret"></i></div>
    </button>
    <ul class="dropdown-menu Cl-dropdown-menu" role="menu">
        <li><a href="#" onclick="changeLanguageMenuIcon('it')">Italiano</a></li>
        <li><a href="#" onclick="changeLanguageMenuIcon('en')">English</a></li>
        <li><a href="#" onclick="changeLanguageMenuIcon('fr')">Français</a></li>
    </ul>
</div>
```

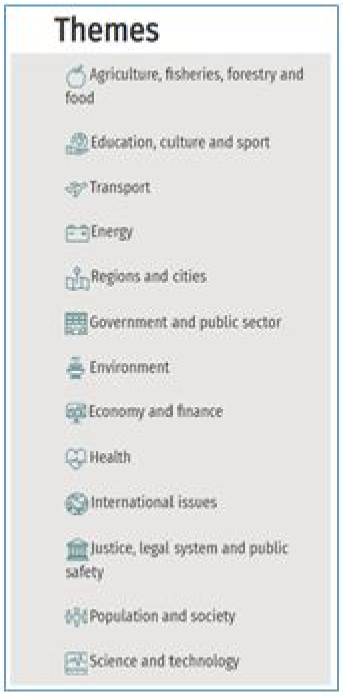
The red rectangle highlights the css class that allows you to view the flag, this must be defined in the file **../static/referenceMetadata/mystyle.css**. In the following way:



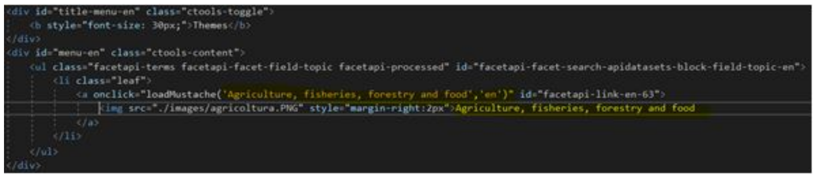
```
.Cl-FlagFR {
    background-image: url(./media/fr.f8952213.svg);
}
```

The file **fr.f8952213.svg** is the flag image in svg format, this file must be present in the folder **..static/referenceMetadata/media**. It can be copied directly from the folder **main/app/client/static/media** which contains examples for various languages.

Theme menu



In this case you have to change the value of the title (div with id **title-menu-***) and the names of the links below that forward requests to extract the report data for the related topic. Below is the code for the English language to be changed.
Within the function **loadMustache** you must indicate the name of the theme and the prefix of the language.
The value of the theme will have to be recovered from codelist **CL_DCAT_AP_IT_THEME** of the **MSD DCAT-AP_IT** used.



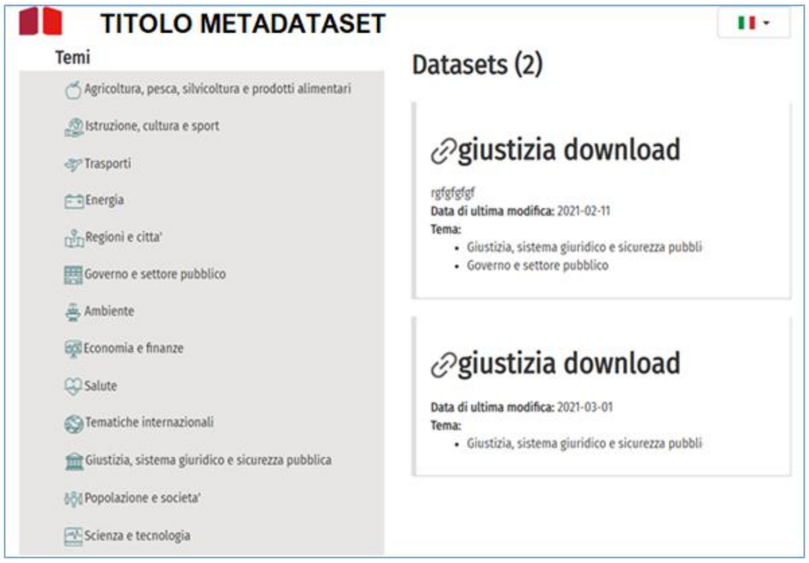
```
<div id="title-menu-en" class="ctools-toggle">
    <b style="font-size: 30px;">Themes</b>
</div>
<div id="menu-en" class="ctools-content">
    <ul class="facetapi-terms facetapi-facet-field-topic facetapi-processed" id="facetapi-facet-search-apidatasets-block-field-topic-en">
        <li class="leaf">
            <a onclick="loadMustache('Agriculture, fisheries, forestry and food','en')" id="facetapi-link-en-63">
                <img src="./images/agricoltura.PNG" style="margin-right:2px">Agriculture, fisheries, forestry and food
            </a>
        </li>
    </ul>
</div>
```

2.7.1.3 Example of using HTML widgets

Here are some examples of invocation of HTML widgets in the case of decoupled scenario (see appendix *Configurations and contexts Metadata API*).

2.7.1.3.1 Widget CKAN catalogs

<BASE_URL>/static/referenceMetadata/template/CategoryTemplate.html?metadataSetId=<ID_METADATASET>&lang=<LINGUA(it,en)>#


TITOLO METADATASET
Temi
Agricoltura, pesca, silvicoltura e prodotti alimentari
Istruzione, cultura e sport
Trasporti
Energia
Regioni e citta'
Governo e settore pubblico
Ambiente
Economia e finanze
Salute
Tematiche internazionali
Giustizia, sistema giuridico e sicurezza pubblica
Popolazione e societa'
Scienza e tecnologia

Datasets (2)

giustizia download

rgfgfgfgf
Data di ultima modifica: 2021-02-11
Tema:
- Giustizia, sistema giuridico e sicurezza pubbli
- Governo e settore pubblico

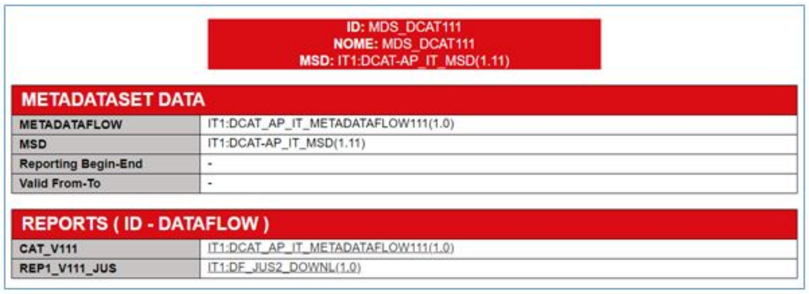
giustizia download

Data di ultima modifica: 2021-03-01
Tema:
- Giustizia, sistema giuridico e sicurezza pubbli

By selecting one of the themes from the left menu, all reports with that theme will be searched. Clicking on the title of one of them will open the report widget with the related data.

2.7.1.3.2 Widget metadataset

<BASE_URL>/static/referenceMetadata/template/MetadatasetTemplate.html?metadataSetId=<ID_METADATASET>&reportId =<ID_REPORT>& lang=<LINGUA(it,en)>#



ID: MDS_DCAT111
NOME: MDS_DCAT111
MSD: IT1:DCAT-AP_IT_MSD(1.11)

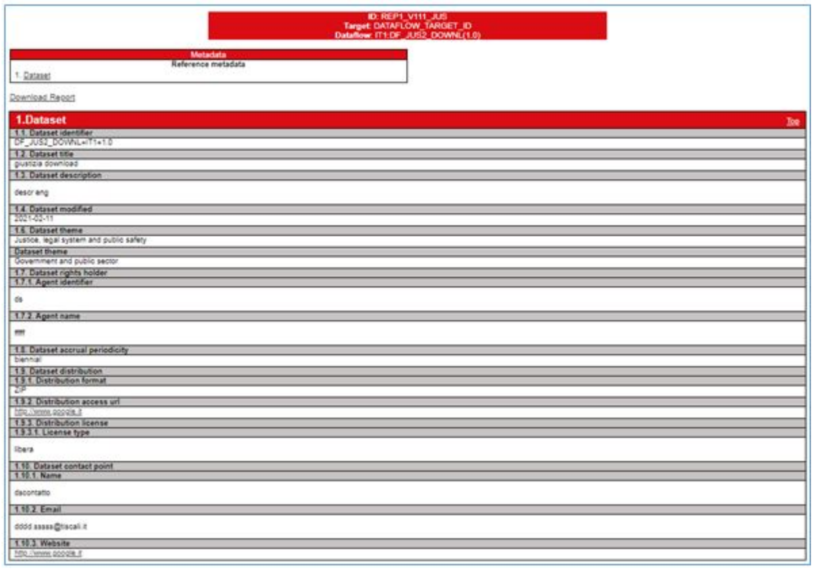
METADATASET DATA	
METADATAFLOW	IT1:DCAT_AP_IT_METADATAFLOW111(1.0)
MSD	IT1:DCAT-AP_IT_MSD(1.11)
Reporting Begin-End	-
Valid From-To	-

REPORTS (ID - DATAFLOW)	
CAT_V111	IT1:DCAT_AP_IT_METADATAFLOW111(1.0)
REP1_V111_JUS	IT1:DF_JUS2_DOWNL(1.0)

Shows the metadataset data passed as a parameter and the list of reports associated with it.

2.7.1.3.3 Widget report

<BASE_URL>/static/referenceMetadata/template/GenericMetadataTemplate.html?metadataSetId=<ID_METADATASET>&reportId =<ID_REPORT>& lang=<LINGUA(it,en)>#



ID: REP1_V111_JUS
Target: DATAFLOW_TARGET_ID
Dataflow: IT1:DF_JUS2_DOWNL(1.0)

Metadata
Reference metadata
1. Dataset

Download Report

1.Dataset Top

1.1. Dataset identifier
DF_JUS2_DOWNL+IT1+1.0
1.2. Dataset title
giustizia download
1.3. Dataset description
descr eng
1.4. Dataset modified
2021-02-11
1.6. Dataset theme
Justice, legal system and public safety
Dataset theme
Government and public sector
1.7. Dataset rights holder
1.7.1. Agent identifier
ds
1.7.2. Agent name
fffff
1.8. Dataset accrual periodicity
biennial
1.9. Dataset distribution
1.9.1. Distribution format
ZIP
1.9.2. Distribution access url
http://www.google.it
1.9.3. Distribution license
1.9.3.1. License type
libera
1.10. Dataset contact point
1.10.1. Name
dscontatto
1.10.2. Email
dddd.sssss@tiscali.it
1.10.3. Website
http://www.google.it

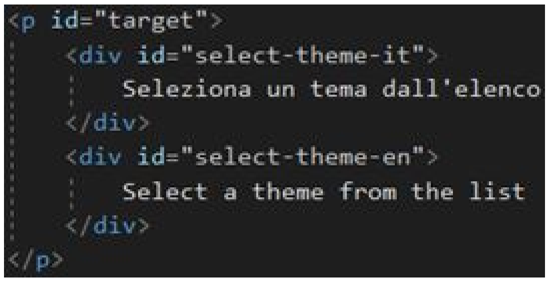
It shows all the report data divided by section, the field name in gray and its value in white.

Data display

Change the message for the theme selection:

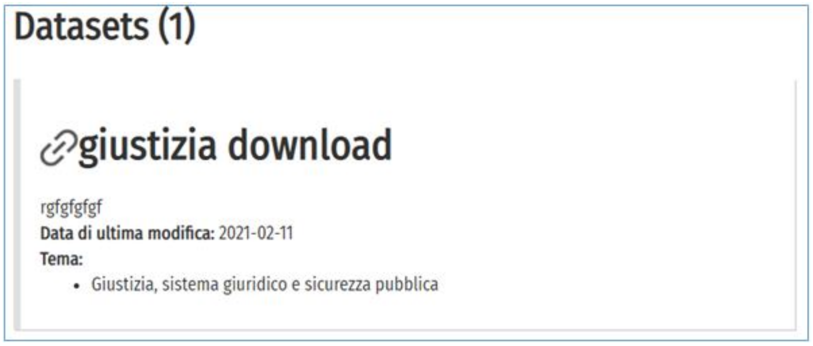
Seleziona un tema dall'elenco

Add a new div into:



```
<p id="target">
    <div id="select-theme-it">
        Seleziona un tema dall'elenco
    </div>
    <div id="select-theme-en">
        Select a theme from the list
    </div>
</p>
```

And finally add the part with the report data found in the selected theme:

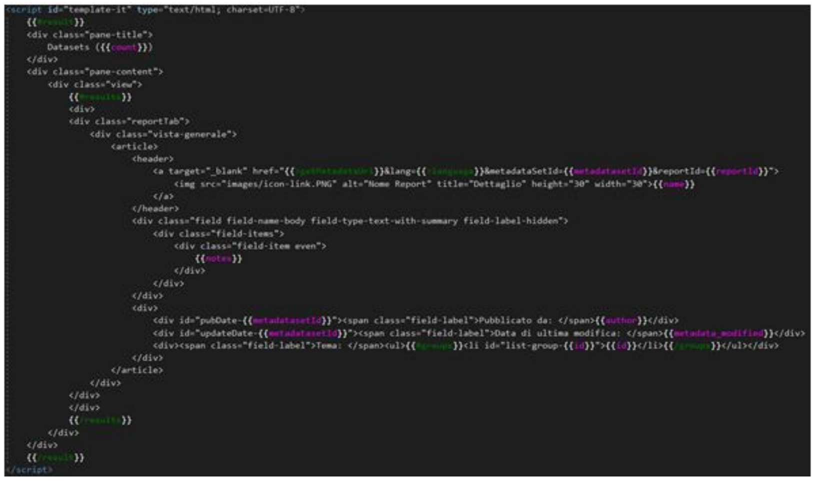


Datasets (1)

giustizia download

rgfgfgfgf
Data di ultima modifica: 2021-02-11
Tema:
- Giustizia, sistema giuridico e sicurezza pubblica

Adding a new script tag like the following for the Italian language:



```
<script id="template-it" type="text/html; charset=UTF-8">
    {{#result}}
    <div class="pane-title">
        Datasets ({{count}})
    </div>
    <div class="pane-content">
        <div class="view">
            {{#results}}
            <div>
            <div class="reportTab">
                <div class="vista-generale">
                    <article>
                        <header>
                            <a target="_blank" href="{{~getMetadataUrl}}&lang={{~language}}&metadataSetId={{metadatasetId}}&reportId={{reportId}}">
                                <img src="images/icon-link.PNG" alt="Nome Report" title="Dettaglio" height="30" width="30">{{name}}
                            </a>
                        </header>
                        <div class="field field-name-body field-type-text-with-summary field-label-hidden">
                            <div class="field-items">
                                <div class="field-item even">
                                    {{notes}}
                                </div>
                            </div>
                        </div>
                        <div>
                            <div id="pubDate-{{metadatasetId}}"><span class="field-label">Pubblicato da: </span>{{author}}</div>
                            <div id="updateDate-{{metadatasetId}}"><span class="field-label">Data di ultima modifica: </span>{{metadata_modified}}</div>
                            <div><span class="field-label">Tema: </span><ul>{{#groups}}<li id="list-group-{{id}}">{{id}}</li>{{/groups}}</ul></div>
                        </div>
                    </article>
                </div>
            </div>
            </div>
            {{/results}}
        </div>
    </div>
    {{/result}}
</script>
```

Modifying it with the labels and identifiers for the new language.

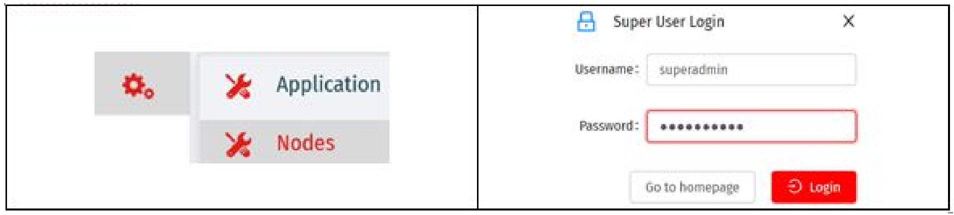
2.8 The first time that the application is used and upgrade to newer versions

The default credentials are:

- Administrator of the node (each node can have a different administrator): **admin**/[empty string]
- Superadmin (at the level of the application): **superadmin/superadmin**

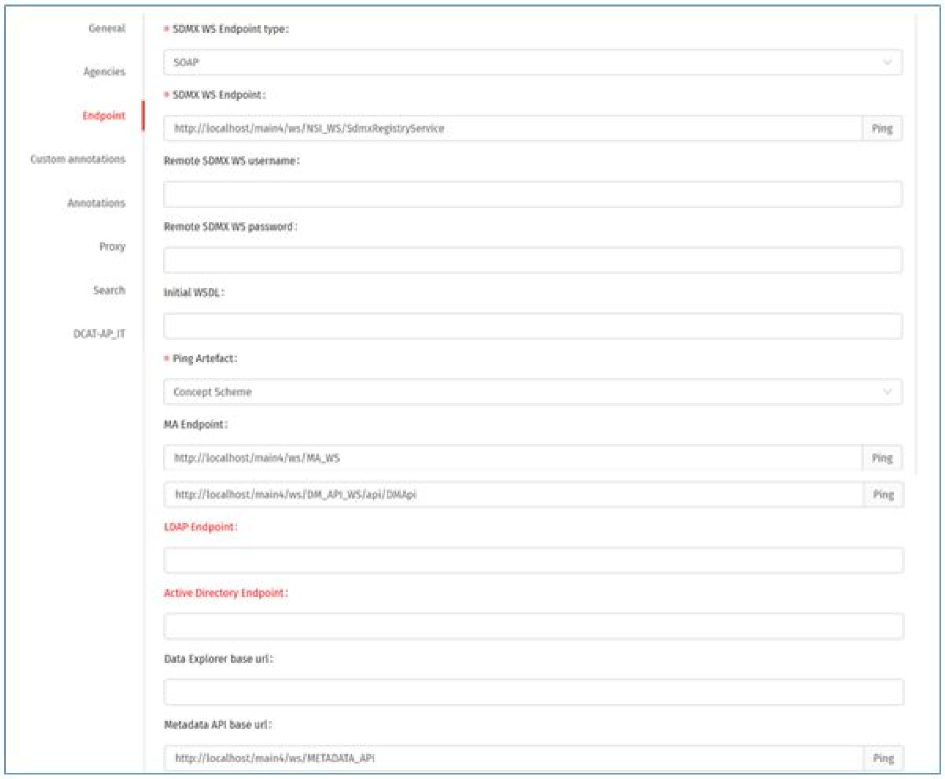
1. Run the URL of the client module (e.g. *https://localhost/client*). The 'welcome" page is shown. The first time the application is used, it is necessary to configure at least one Node. In this case a pop-up alerts that no Nodes are configured is shown, and the application suggests a link that allows to configure Nodes.

2. If you click on the link or if you select the suitable menu on the left side, the credentials of the superadmin is required



3. The configuration consists in specifying the endpoint of the web services:

    1. SDMX WS Endpoint: https://localhost/NSI_WS/SdmxRegistryService
    2. MA Endpoint: https://localhost/MA_WS
    3. DM Endpoint: https://localhost/DM_API_WS/api/DMApi
    4. Base Url Metadata Api: https://localhost/METADATA_API



* SDMX WS Endpoint type: SOAP

* SDMX WS Endpoint: http://localhost/main4/ws/NSI_WS/SdmxRegistryService

Remote SDMX WS username:

Remote SDMX WS password:

Initial WSDL:

* Ping Artefact: Concept Scheme

MA Endpoint: http://localhost/main4/ws/MA_WS

http://localhost/main4/ws/DM_API_WS/api/DMApi

LDAP Endpoint:

Active Directory Endpoint:

Data Explorer base url:

Metadata API base url: http://localhost/main4/ws/METADATA_API

Endpoints configuration for a specific Node

Each endpoint configuration can be checked clicking on the button [Ping].

When the configuration is saved the credential of the administrator of the Node are required.

4. Through the menu "Import structure", import the files contained in the folder *main/files/ReferenceMetadata*, that contains the artefacts used for DCAT_AP_IT
5. Through the menu "Import structure", import at least a Category Scheme (or create one interactively) that will be used internally by the application to categorize data cubes.
6. Through the menu "User management" add new users, and through the menu "Set permissions" assign the suitable authorizations to the new users.

For software version upgrades, in addition to updating the ws and possibly the db, you must refer to the file *main/app/readme.txt*.

2.9 DCAT-AP_IT report creation

Below you can find attributes (listed by id) that are mandatory for creating reports according to the last (10.0) version of the MSD for DCAT-AP_IT standard. Attributes are divided by typology:

- **Catalogue**
  - DCAT_AP_IT_CATALOGUE_TITLE
  - DCAT_AP_IT_CATALOGUE_DESCRIPTION
  - DCAT_AP_IT_CATALOGUE_AGENT_IDENTIFIER
  - DCAT_AP_IT_CATALOGUE_AGENT_NAME

  - DCAT_AP_IT_CATALOGUE_LAST_UPDATE
- **Report**
  - DCAT_AP_IT_DATASET_IDENTIFIER
  - DCAT_AP_IT_DATASET_TITLE
  - DCAT_AP_IT_DATASET_DESCRIPTION
  - DCAT_AP_IT_DATASET_MODIFIED
  - DCAT_AP_IT_DATASET_THEME
  - DCAT_AP_IT_DATASET_RIGHTS_HOLDER_AGENT_IDENTIFIER
  - DCAT_AP_IT_DATASET_RIGHTS_HOLDER_AGENT_NAME
  - DCAT_AP_IT_DATASET_ACCRUAL_PERIODICITY
  - DCAT_AP_IT_DATASET_DISTRIBUTION_FORMAT
  - DCAT_AP_IT_DATASET_DISTRIBUTION_ACCESS_URL

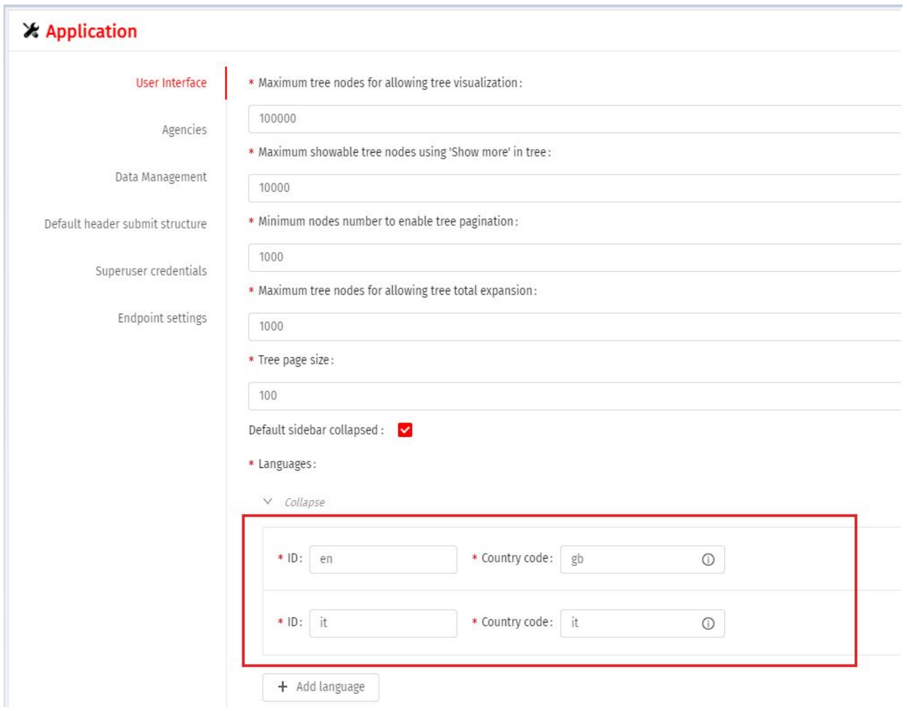
If you do not indicate the values for all the above attributes, you can only create reports in DRAFT state.

2.10 Appendix

2.10.1 Languages management

To localize the GUI in a specific language, perform the following steps:

1. Enter the folder *main/app/client/static/locales*
2. Duplicate the folder **en** and rename it with the ISO letters of the language (e.g. **es** for Spanish)
3. Inside the new folder edit the file **translation.json** and translate all the labels and messages. In the 'languages' section add the translation for the language you want to add (eg. "ar": "Arabic"). This last operation must be for the translation.json file of each folder in path: *main/app/client/static/locales*
4. Access to the page http://localhost/client/#/configurations/app and add in the *User Interface* section the code of the language for the inserted ID (eg. *ar* per Arabic) in the *Languages* section and the code for the flag to be shown for the language in the field 'Country Code' (eg. sa for South Arabia).



Application

* Maximum tree nodes for allowing tree visualization:	100000
* Maximum showable tree nodes using 'Show more' in tree:	10000
* Minimum nodes number to enable tree pagination:	1000
* Maximum tree nodes for allowing tree total expansion:	1000
* Tree page size:	100
Default sidebar collapsed:	✓

* Languages:

* ID	* Country code
en	gb
it	it

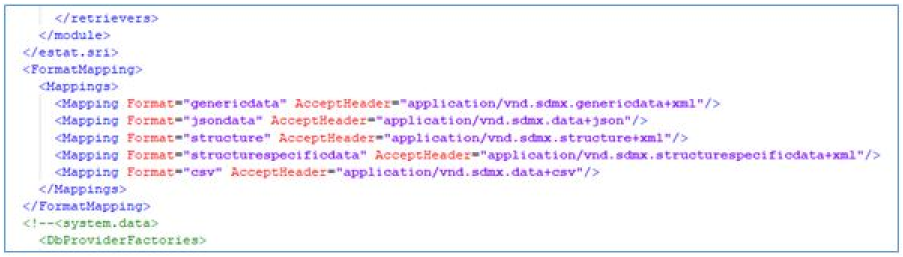
- Add a new entry in the field Languages in order to localize the GUI*

To add a new data language:

- Access the page: http://localhost/client/#/configurations/app and in the tab *Data Management* add the new language as described at step 4 above
- If you want to associate a label to the new added language, access to the *translation.json* file of each folder in the path: *main/app/client/static/locales* for whom you want to add a label and add in the *Languages* section the translation for the code of the language you want to translate (eg. "ar": "Arabic").

2.10.2 Download formats support

In order to support all the available formats for download first of all is necessary to define the corresponding FormatMapping in the following file: *main/app/ws/MA_WS/Estat.Sri.Mapping.Ws.ServiceCore.dll.config*.



```
    </retrievers>
  </module>
</estat.sri>
<FormatMapping>
  <Mappings>
    <Mapping Format="genericdata" AcceptHeader="application/vnd.sdmx.genericdata+xml"/>
    <Mapping Format="jsondata" AcceptHeader="application/vnd.sdmx.data+json"/>
    <Mapping Format="structure" AcceptHeader="application/vnd.sdmx.structure+xml"/>
    <Mapping Format="structurespecificdata" AcceptHeader="application/vnd.sdmx.structurespecificdata+xml"/>
    <Mapping Format="csv" AcceptHeader="application/vnd.sdmx.data+csv"/>
  </Mappings>
</FormatMapping>
<!--<system.data>
  <DbProviderFactories>
```

Supported formats for download

The list of the formats that can be supported by the Eurostat's web services includes the following formats:

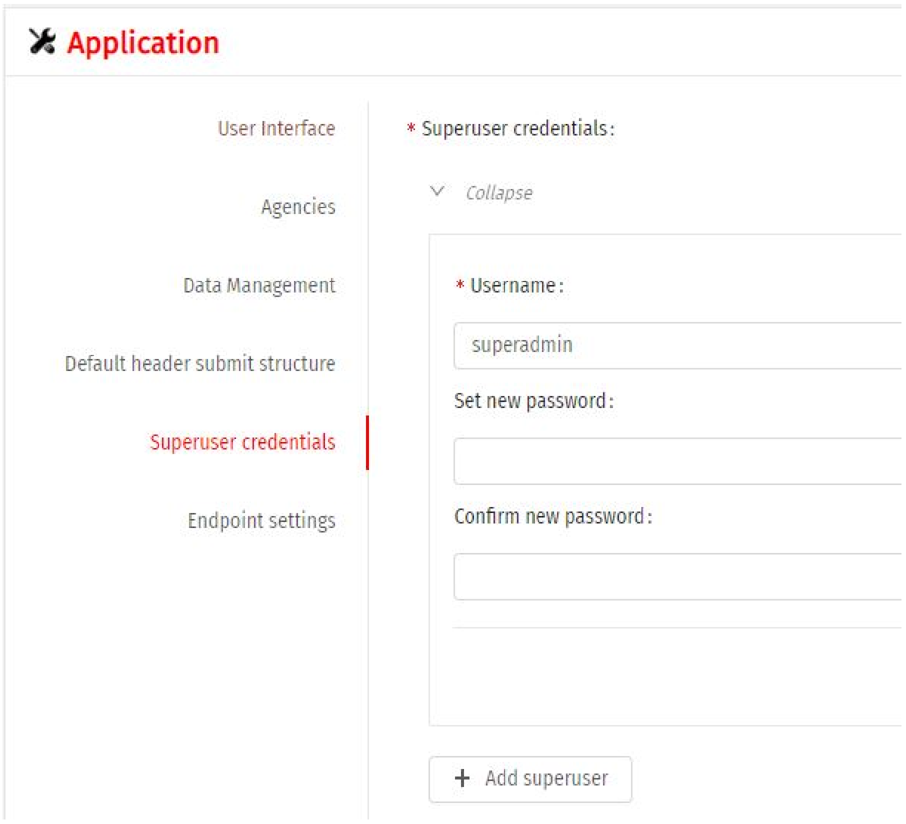
Format	AcceptHeader
genericdata	application/vnd.sdmx.genericdata+xml
genericdata20	application/vnd.sdmx.genericdata+xml; version=2.0; charset=utf-8
jsondata	application/vnd.sdmx.data+json
structurespecific-data	application/vnd.sdmx.structurespecificdata+xml
csv	application/vnd.sdmx.data+csv
rdf	application/rdf+xml
compactdata20	application/vnd.sdmx.compactdata+xml
edidata	application/vnd.sdmx.edidata
crosssectionaldata	application/vnd.sdmx.crosssectionaldata+xml
structure	application/vnd.sdmx.structure+xml
xml	application/xml

For each format not originally included that has to be supported for download, is necessary to add such type of row:
<Mapping Format="NomeFormato" AcceptHeader="application/…"/>.
The name of the format can be not standard: in the developed functions has been made a mapping among the download format requested by the user and the corresponding AcceptHeader, that is inserted in the header of the related query to the web service.
Finally, if not present, the file *main/files/Estat.SdmxSource.Extension.RDFPlugin.dll* has to be copied in the respective folders of the NSI WS (*main/app/ws/NSI_WS/Plugins*) and of the MA WS (*main/app/ws/MA_WS/Plugins*) in order to support the RDF formatted download.

2.10.3 Superadmin password change

In order to change the password of the "superadmin" user, the following tasks have to be performed:

1. Access the *Configuration/Application* page with the current superadmin user's credentials
2. Access the *Superuser Credentials* section
3. Fill in the new password in the *Set New Password* and *Confirm New Password* text boxes



* Superuser credentials:

* Username: superadmin

Set new password:

Confirm new password:

Change superadmin credentials

2.10.4 Scenario: access to a remote node in reading and writing mode

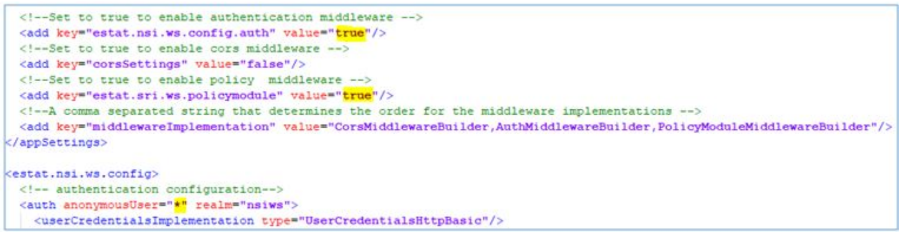
Let's consider a scenario in which there's a remote SDMX endpoint to which it is necessary to access in reading and writing mode. In order to access in writing mode to the node, it is necessary to install and configure the DM API service together with the AUTHDB database opportunely configured.
If the SDMX endpoint is "open" (Authentication Middlware disabled, see the "estat.nsi.ws.config.auth" property inside the app.config file of the SDMX WS endpoint), authentication by using the credentials of a registered user into that AUTHDB database is enough. If instead the SDMX endpoint has the Authentication Middlware enabled (and configured with its own AUTHDB database, known as AUTHDB_SDMX database), it is necessary to insert in the AUTHDB database a user (with the related permissions) with the same credentials stored in the AUTHDB_SDMX database that has to be used.

The node configuration then needs to foresee:

- **SdmxWS**: the endpoint to which to connect
- **MA WS**: empty
- **DM API WS**: url of the ws used for the authentication

2.10.5 Permission rules management

The AuthDb database foresees the use of a set of AccessRules that regulate the access of the users to the methods exposed by the web service.
In order to activate this middleware, the following configuration parameters have to be set in the *app.config* file of the NSI WS or in the *Estat.Sri.Mapping.Ws.ServiceCore.dll.config* file of the MA WS.



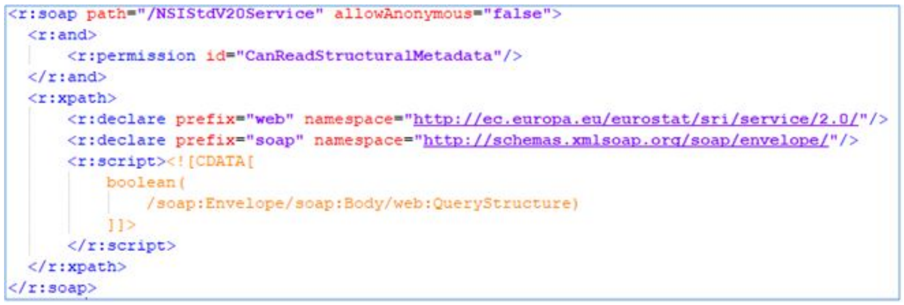
```
  <!--Set to true to enable authentication middleware -->
  <add key="estat.nsi.ws.config.auth" value="true"/>
  <!--Set to true to enable cors middleware -->
  <add key="corsSettings" value="false"/>
  <!--Set to true to enable policy  middleware -->
  <add key="estat.sri.ws.policymodule" value="true"/>
  <!--A comma separated string that determines the order for the middleware implementations -->
  <add key="middlewareImplementation" value="CorsMiddlewareBuilder,AuthMiddlewareBuilder,PolicyModuleMiddlewareBuilder"/>
</appSettings>

<estat.nsi.ws.config>
  <!-- authentication configuration-->
  <auth anonymousUser="*" realm="nsiws">
    <userCredentialsImplementation type="UserCredentialsHttpBasic"/>
```

Configuration parameters for activating the management of the permissions rules

- Estat.nsi.ws.config.auth = true
- Estat.sri.ws.policymodule = true
- anonymousUser = *

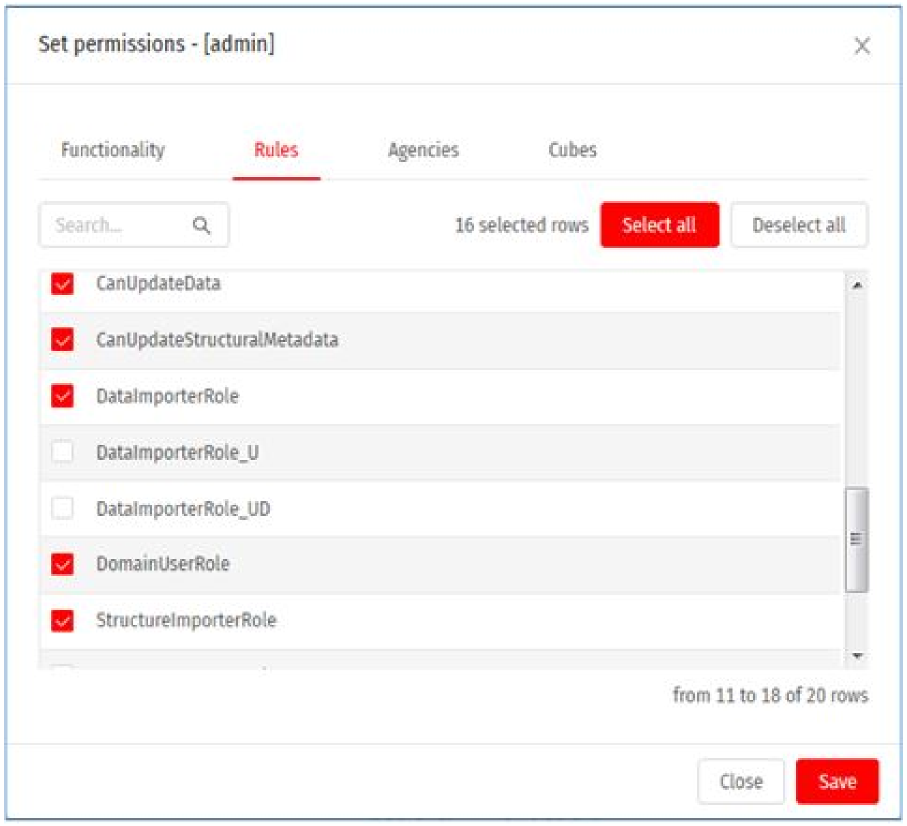
For each method of the NSI WS, in the *nsiws.xml* file inside the *App_Data* folder, are defined which rules the user has to have in order to access the various methods exposed by the web service.

For instance the rule:



```
<r:soap path="/NSIStdV20Service" allowAnonymous="false">
  <r:and>
      <r:permission id="CanReadStructuralMetadata"/>
  </r:and>
  <r:xpath>
      <r:declare prefix="web" namespace="http://ec.europa.eu/eurostat/sri/service/2.0/"/>
      <r:declare prefix="soap" namespace="http://schemas.xmlsoap.org/soap/envelope/"/>
      <r:script><![CDATA[
          boolean(
              /soap:Envelope/soap:Body/web:QueryStructure)
          ]]>
      </r:script>
  </r:xpath>
</r:soap>
```

says that in order to access paths as "Envelope/Body/QueryStructure" the "CarReadStrucuturalMetadata" permission is needed and that the access to that path is forbidden to the anonymous user (allowAnonymous = "false"). That file can be manually edited on the base of the specific needs.
The assignment of the rules to the users can be performed through the interface of the section Permissions/Set permissions/Rules.


Set permissions - [admin]

Functionality | Rules | Agencies | Cubes

Search...   16 selected rows   Select all   Deselect all

✓	CanUpdateData
✓	CanUpdateStructuralMetadata
✓	DataImporterRole
	DataImporterRole_U
	DataImporterRole_UD
✓	DomainUserRole
✓	StructureImporterRole

from 11 to 18 of 20 rows

Close   Save

Management of the rules for the users

Some permissions are hierarchical and then if for instance the 'AdminRole' rule is assigned to a user, he implicitly receives all the other rules on cascade.
If, for instance, a user hasn't got the *"CanImportStructures"* right or a rule (role) that indirectly implies that permission (as for example "DataImporterRole") he will not be able to access the 'Import Structure' function and will receive an error response as 'Forbidden'.

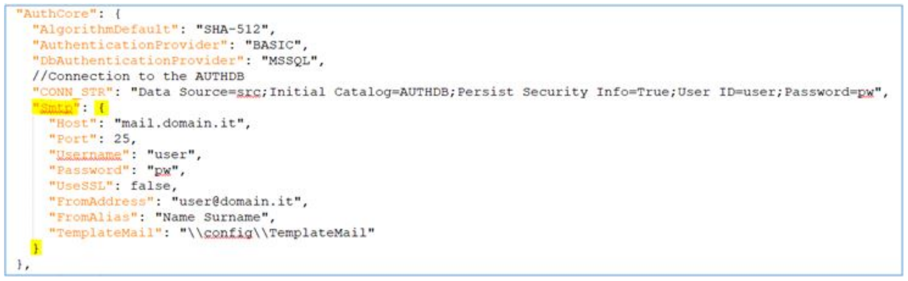
2.10.6 Welcome Page, Footer and Logo configuration

It is possible to make configurable the following elements of the application (via appropriate configuration files):

- the application footer information (text and logo) by adding a file *index.html* to the path static/footer;
- the image (logo) that is displayed when the application is loaded by adding a file named *loading-logo.png* to the path static/png;
- the introductory text of the home-page, also managing localization, by moving the *homePage.html* file to the static/homepage path and renaming it according to the desired language in the format {lang}.html (for example it.html, en.html, etc.);
- the image placed in the top left corner, above the main menu. For header images, the files header-logo.png and header-logo-mini.png must be added to the path static/png.

2.10.7 Configuration server Smtp

To send an e-mail for the Recovery Password service, it is necessary to configure the SMTP server in the *main/app/ws/DM_API_WS/appsetting.json* file.



```
"AuthCore": {
  "AlgorithmDefault": "SHA-512",
  "AuthenticationProvider": "BASIC",
  "DbAuthenticationProvider": "MSSQL",
  //Connection to the AUTHDB
  "CONN_STR": "Data Source=src;Initial Catalog=AUTHDB;Persist Security Info=True;User ID=user;Password=pw",
  "Smtp": {
    "Host": "mail.domain.it",
    "Port": 25,
    "Username": "user",
    "Password": "pw",
    "UseSSL": false,
    "FromAddress": "user@domain.it",
    "FromAlias": "Name Surname",
    "TemplateMail": "\\config\\TemplateMail"
  }
},
```

SMTP Server configuration

The fields to be modified are those between the brackets highlighted in the figure. For security reasons, the password must be entered already encrypted. To encrypt a password, a special tool is available in *main/files/crypt*:

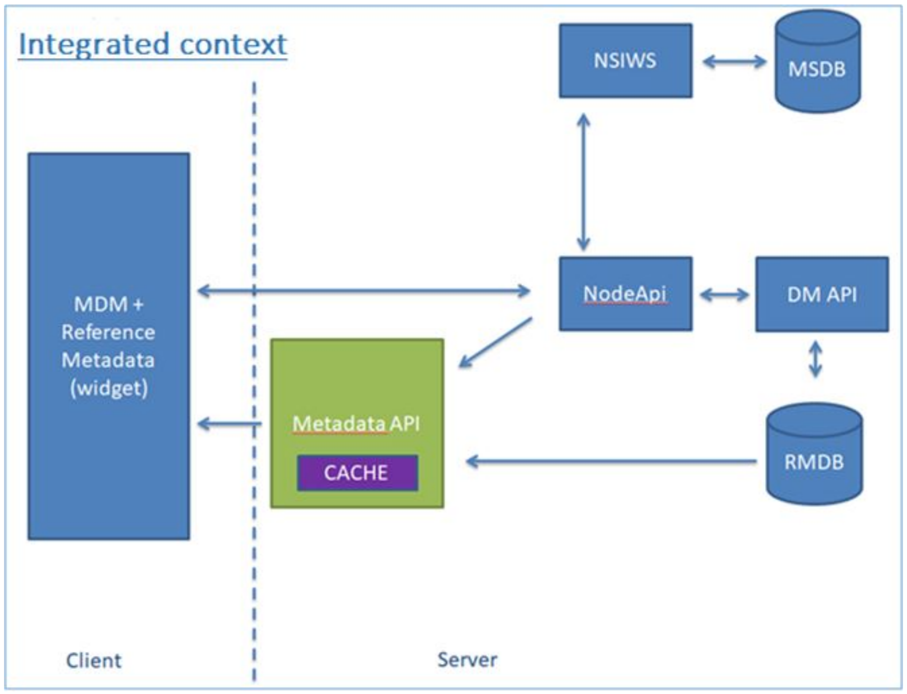
- insert in the encriptKey.txt file the secret encryption key (which must be the same one used in DMApi (see ENCRYPTION_PASSW key in the *appsettings.json* file mentioned above))
- enter the password you wish to encrypt in the *input.txt* file
- double click on *CriptUtilityApp.exe*
- an *output.txt* file with the encrypted password to be entered in the Password field in the figure will be generated in the folder

2.10.8 Configurations and contexts Metadata API

The operation and configuration of the module in question can be analyzed in two distinct contexts:

2.10.8.1 Integrated context

The following figure describes the system where the server modules are installed on a single host and the HTML Metadata API widgets are integrated into the MDM client.

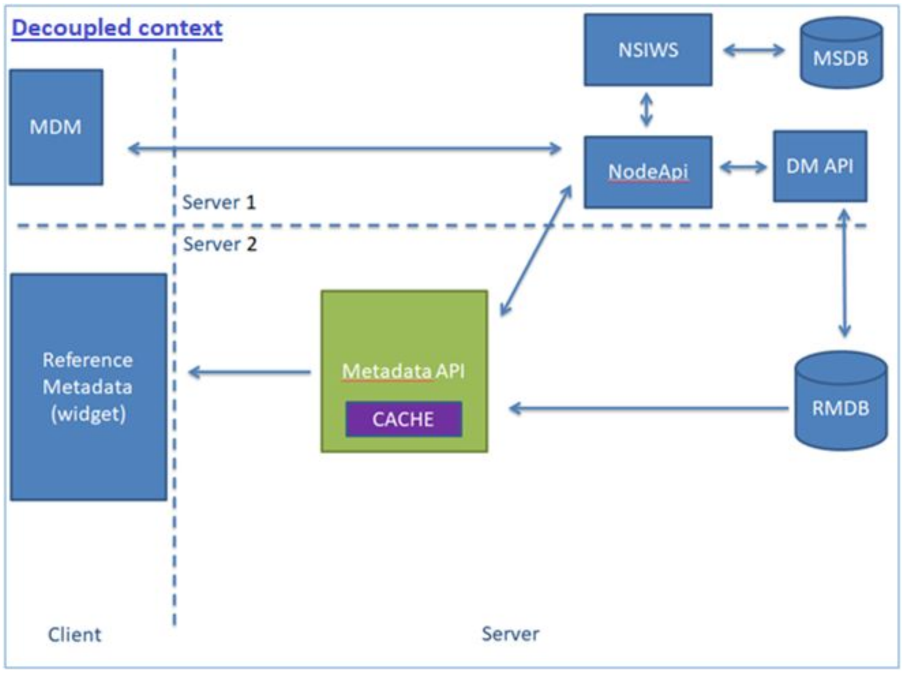


In this case, the system must be configured as follows:

- **METADATA_API**, in the appsettings.json file edit the fields:
  - "NodeBaseUrl": NodeAPI URL. **MANDATORY** because:
    - the internal URL may be different from the external URL;
    - The CKAN standard has a well-defined and not editable format that does not allow information to be retrieved except from configuration.
  - "NodeId": Node identifier; **MANDATORY** for safety reasons. If valorized this NodeId value is always used to contact the NodeAPI; this behavior is adopted for security reasons, that is to avoid that the client can request information of other nodes simply by passing a different NodeId as a parameter in the URLs.
- **DM_API**, in the appsettings.json file edit the fields:
  - "IMPORT_REFERENCE_METADATA_BASE_DIR": directory dedicated to uploading files dedicated to referential metadata
- **CLIENT** (MDM, from interface)
  - Fill in (**MANDATORY** if not done in the following point), in the node configuration, the "Metadata API base url" field with the URL of the Metadata API
  - **WIDGET HTML**, in the /static/referenceMetadata/metadataapi.json file edit
    - "baseUrl": Metadata API URL. **OPTIONAL** because specified by the MDM client

2.10.8.2 Decoupled context

The following figure describes the system with the Metadata API module on a different server from the rest of the components:



In this case, the system must be configured as follows:

- **METADATA_API (appsettings.json, as an integrated context)**
  - **"NodeBaseUrl"**: URL delle NodeAPI. **MANDATORY** because:
    - the internal URL may be different from the external URL
    - The CKAN standard has a well-defined and not editable format that does not allow information to be retrieved except from configuration.
  - **"NodeId"**: Node identifier; **MANDATORY** for safety reasons. If valorized this NodeId value is always used to contact the NodeAPI; this behavior is adopted for security reasons, that is to avoid that the client can request information of other nodes simply by passing a different NodeId as a parameter in the URLs.
- **DM_API (appsettings.json , as an integrated context)**
  - "IMPORT_REFERENCE_METADATA_BASE_DIR": directory dedicated to the upload of referential metadata files
  - **CLIENT (MDM, from interface)**
  - Fill in (OPTIONAL), in the node configuration, the "Metadata API base url" field with the Metadata API URL
  - **WIDGET HTML (/static/referenceMetadata/metadataapi.json)**
    - "baseUrl": Metadata API URL. MANDATORY because on a different server

2.11 Quick steps

This paragraph contains the synthetic summary of the steps needed for installing and configuring the application, starting from the point that the prerequisites have been already satisfied.

2.11.1 IIS Configuration

- Create 5 IIS pools for.NET Core and call them as follows: **POOL_NSIWS, POOL_MAWS, POOL_DMWS, POOL_METAWS** e **POOL_NODEWS**.
- Create an https binding using the SSL certificate present on the install server.

2.11.2 Creation of the DBs

- **AUTHDB**: create a new db and call it **AUTHDB**;
- **MSDB**: create a new db and call it **MASTORE**;
- **DDB**: create a new db and call it **DDB**;
- **RMDB**: create a new db and call it **RMDB**;
- Create a user with Administrator rights and with read/write permissions on the above dbs.

2.11.3 Web services deploy

- Assign to the IIS_IUSRS e IUSERS users read/write grants to the *main/app* folder.

2.11.3.1 NSI WS

- Create under the IIS Default Web Site a new application having **"NSI_WS"** alias, **POOL_NSIWS** application pool and path: *main/app/ws/NSI_WS*;
- Modify the *main/app/ws/NSI_WS/config/app.config* file in order to allow it to point to the **AUTHDB** and to the **MSDB** databases previously created.

2.11.3.2 MA API WS

- Create under the IIS Default Web Site a new application with **"MA_WS"** alias, **POOL_MAWS** application pool and path: *main/app/ws/MA_WS*;
- Modify the: *main/app/ws/MA_WS/AppData/ConnectionString* file in order to allow it to point to the **AUTHDB** and to the **MSDB** databases previously created.

2.11.3.3 DM API WS

- Create under the IIS Default Web Site a new application with **"DM_API_WS"** alias, **POOL_DMAPIWS** application pool and path: *main/app/ws/DM_API_WS*;
- Modify as follows the *main/app/ws/DM_API_WS/appsetting.json* file:
  - **AuthCore/CONN_STR** must point to the previously created **AUTHDB** database
  - **DATA_PROVIDER_NAME/DEFAULT_DATA/CONN_STR** must point to the previously created **DDB** database
  - **DATA_PROVIDER_NAME/RM_DATA/CONN_STR** must point to the previously created **RMDB** database
  - **DMApiSettings/IMPORT_FILE_BASE_DIR** must indicate the base filesystem path in which the loaded data files will be saved
  - **DMApiSettings/XML_MAPPING_BASE_DIR** must indicate the base filesystem path in which the loaded xml mapping for excel files will be saved

2.11.3.4 METADATA API

- Create under the IIS Default Web Site a new application with **"METADATA_API"** alias, **POOL_METAWS** application pool and path *main/app/ws/METADATA_API*;
- Modify the *main/app/ws/METADATA_API/appsetting.json* file in order to allow it to point to the **RMDB** previously created.
  - The value of the **NodeBaseUrl** key, to the url of the **NodeApi** module, described in the next paragraph.
  - The value of the **NodeId** key, its value is always used to contact the NodeAPI.
  - The value of the **Cache.Dir** key, so that it points to the path where the user wants cache files to be stored.
  - The value of the **Cache.Enable** key, with the value true or false for use or not the cache component.
  - The value of the **AllowedHosts** key, with a list of hostname addresses, for change the default host filtering middleware.

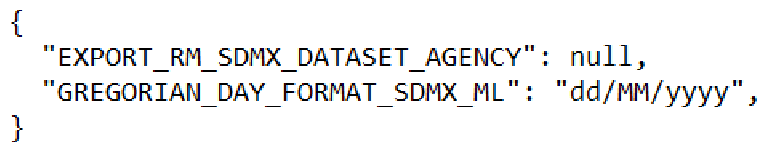
2.11.3.5 NodeApi

- Create under the IIS Default Web Site a new application with **"NODE_API"** alias, **POOL_NODEWS** application pool and path *main/app/ws/NODE_API.*
- In the **appsettings.json** file it is also possible to customize the name of the SENDER present in the json and xml messages created by the data exports. Just add or configure the entry **EXPORT_RM_SDMX_SENDER_ID**:

  ```
  {
    "EXPORT_RM_SDMX_SENDER_ID": "MDM_SENDER"
  }
  ```

  Configuration SENDER ID for export

- In the **appsettings.json** file it is also possible to customize the name of the dataset agency and date format present in the xml messages created by the data exports. Just add or configure the **EXPORT_RM_SDMX_DATASET_AGENCY** and **GREGORIAN_DAY_FORMAT_SDMX_ML** entries as shown in the following example:

  ```
  {
    "EXPORT_RM_SDMX_DATASET_AGENCY": null,
    "GREGORIAN_DAY_FORMAT_SDMX_ML": "dd/MM/yyyy",
  }
  ```

  Other Configurations for export

  In this case dataset agency (EXPORT_RM_SDMX_DATASET_AGENCY) is not specified, so the metadataflow agency will be used as the default value.

2.11.4 DBs initialization

2.11.4.1 Initialization Wizard

- Inizialize AuthDB
- Extend AuthDb
- Check Mapping Store
- Inizialize Mapping Store
- Inizialize DDB
- Inizialize RMDB

2.11.4.2 Disable the Wizard

2.11.5 Client deploy

- Create under the IIS Default Web Site a new virtual directory with **"client"** alias and path: *main/app/client*;
- Modify the *main/app/client/static/config.json* file by setting the fetchUrl field to *https://localhost/NODE_API*.
- Configure the **baseUrl** entry of the file **metadataapi.json** file in the folder **referenceMetadata** of the client with the MetadataApi url.

2.11.6 First access to the application

- Access to: *https://localhost/client/#/configurations/nodes* with the "superadmin" credentials, click on the *'Add item'* button and create a new node with Id: *'N1'*, Name: *'Test'* and the following Endpoints:
  - SDMX WS Endpoint: *https://localhost/NSI_WS/SdmxRegistryService*
  - MA Endpoint: *https://localhost/MA_WS*
  - DM Endpoint: *https://localhost/DM_API_WS/api/DMApi*
  - Base Url Metadata Api: *https://localhost/METADATA_API*

- Verify the correct definition of the endpoints by clicking on the related 'Ping' buttons.
- *Warning!* After the import of the CategoryScheme for categorizing cubes, access, from the main menu, the *Manage users/Set permissions* and assign to the "admin" user the permissions on all the categories cubes.
- Import through the *Import structures* function from the *Utilities* menu, the files contained in the *main/files/ReferenceMetadata* folder after having logged in.