
MetaDataManager - Installation manual

Release 2.1.10

ISTAT

Nov 28, 2023

CONTENTS

1	Introduction	3
1.1	Aim of this document	3
1.2	Acronyms and glossary	3
2	Software installation	5
2.1	Pre-requisites	5
2.1.1	Operating system	5
2.1.2	Sql Server	5
2.1.3	IIS	5
2.1.4	.NET	6
2.1.5	Eurostat Web Service	6
2.1.6	SSL certificate	7
2.2	IIS configuration	7
2.2.1	Parameters for the management of big files	8
2.2.2	Other parameters	10
2.2.3	Https Binding	12
2.3	Software package	12
2.4	Building databases	14
2.5	Web Services deployment	15
2.5.1	NSI WS	16
2.5.2	MA API WS	19
2.5.3	DM API WS	19
2.5.4	Metadata API	21
2.5.5	Node API	22
2.6	Database initialization	23
2.6.1	Login	23
2.6.2	AuthDB + Extensions MSDB	23
2.6.3	Check of MSDB status	24
2.6.4	Check the status of MSDB, DDB and RMDB	24
2.6.5	Disable the Wizard	25
2.7	Client deployment	25
2.7.1	Reference metadata	26
2.7.1.1	Deploy Report HTML	26
2.7.1.2	Add language report HTML CKAN	27
2.7.1.3	Example of using HTML widgets	29
2.7.1.3.1	Widget CKAN catalogs	29
2.7.1.3.2	Widget metadataset	29
2.7.1.3.3	Widget report	30
2.8	The first time that the application is used and upgrade to newer versions	31
2.9	DCAT-AP_IT report creation	33

2.10	Appendix	34
2.10.1	Languages management	34
2.10.2	Download formats support	35
2.10.3	Superadmin password change	36
2.10.4	Scenario: access to a remote node in reading and writing mode	36
2.10.5	Permission rules management	37
2.10.6	Welcome Page, Footer and Logo configuration	38
2.10.7	Configuration server Smtip	38
2.10.8	Configurations and contexts Metadata API	39
2.10.8.1	Integrated context	39
2.10.8.2	Decoupled context	40
2.11	Quick steps	41
2.11.1	IIS Configuration	41
2.11.2	Creation of the DBs	41
2.11.3	Web services deploy	41
2.11.3.1	NSI WS	42
2.11.3.2	MA API WS	42
2.11.3.3	DM API WS	42
2.11.3.4	METADATA API	42
2.11.3.5	NodeApi	43
2.11.4	DBs initialization	43
2.11.4.1	Initialization Wizard	43
2.11.4.2	Disable the Wizard	43
2.11.5	Client deploy	43
2.11.6	First access to the application	44

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

SOFTWARE INSTALLATION

2.1 Pre-requisites

2.1.1 Operating system

The supported operating systems are the same ones supported by .NET. 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

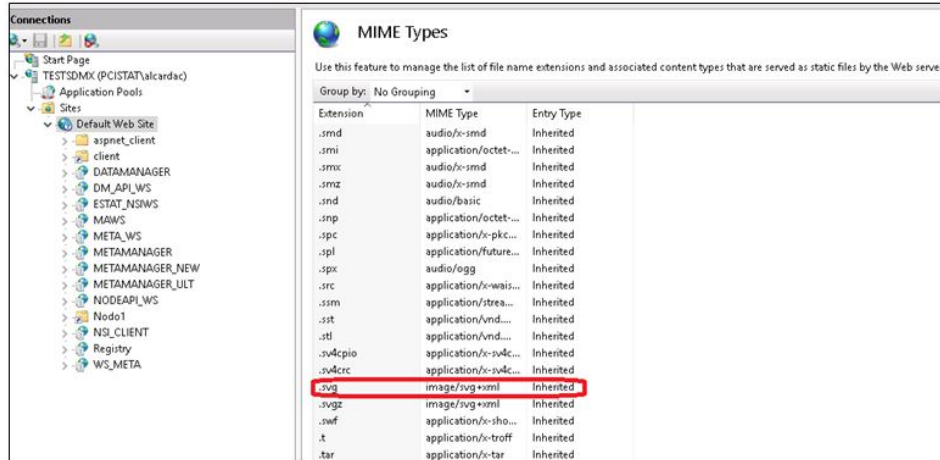
2.1.2 Sql Server

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

2.1.4 .NET

The **.NET Framework ver. 6.0** has to be installed, verifying before the compliance with all the related prerequisites (as specified in <https://learn.microsoft.com/it-it/dotnet/core/install/windows?pivot=os-windows&tabs=net60>).

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

Microsoft ASP.NET Core 6.0.10 - Shared Framework (x86)	Microsoft Corporation	19/10/2022
Microsoft .NET 6.0.10 - Windows Server Hosting	Microsoft Corporation	19/10/2022
Microsoft .NET Runtime - 6.0.10 (x64)	Microsoft Corporation	19/10/2022
Microsoft .NET Runtime - 6.0.10 (x86)	Microsoft Corporation	19/10/2022
Microsoft ASP.NET Core 6.0.10 - Shared Framework (x64)	Microsoft Corporation	19/10/2022

.Net

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

1. Download the package from the following URL: <https://dotnet.microsoft.com/en-us/download/dotnet/6.0>. Install from the package the .NET runtime, the libraries and the ASP.NET 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.

2.1.5 Eurostat Web Service

- **MA API WS** version 8.16.0 with MSDB in version 6.20
- **NSI WS** version 8.16.0 or higher supporting write mode, with the following characteristics:
 - MSDB version 6.20
 - 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 8.16.0.

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

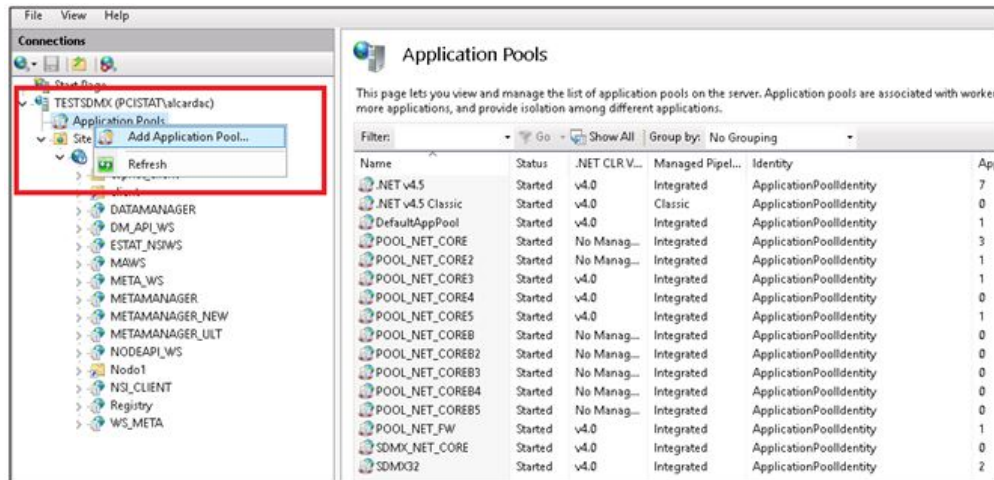
- NSI WS: 8.16.0
- MA WS: 8.16.0

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

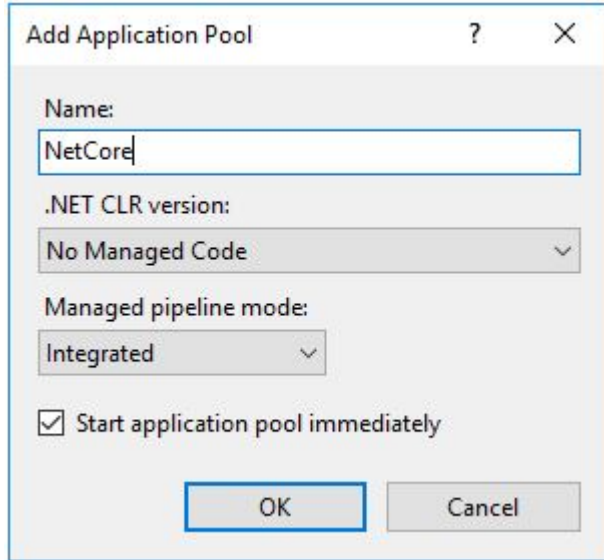
First of all, the creation of an application pool for each .NET application is needed. It is possible to create a new application pool by right-clicking on “Application Pool” (see Fig. *Application pool IIS for .NET*) and by clicking on ‘Add Application Pool’ Item.



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 (e.g. named **POOL_NSIWS**, **POOL_MAWS**, **POOL_DMWS**, **POOL_NODEWS**, **POOL_METAWS**).



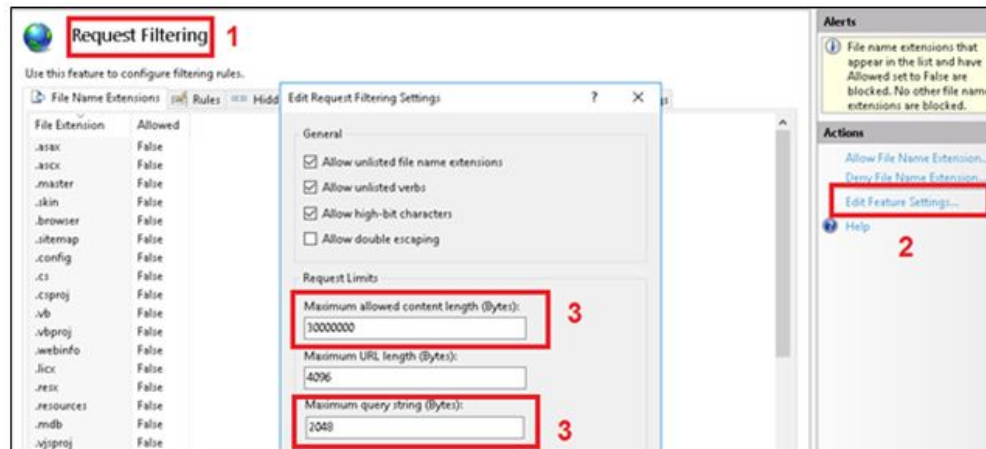
Application pool IIS for .NET

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.

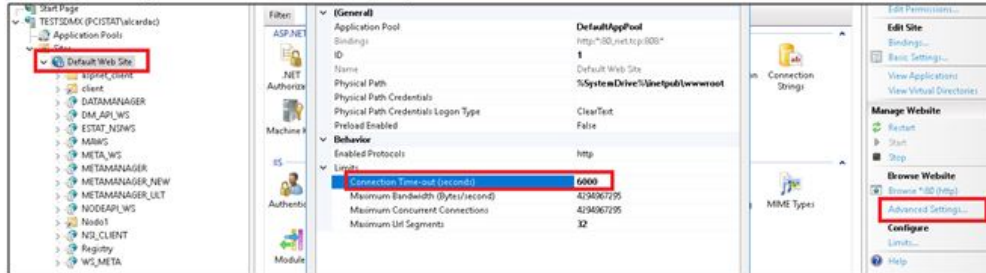


Editing of the “Maximum Allowed Content Length” parameter

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.

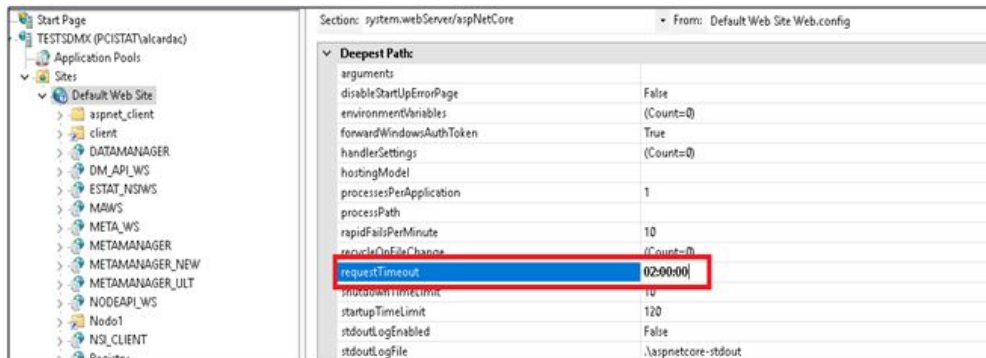


Editing of the Connection Timeout parameter

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

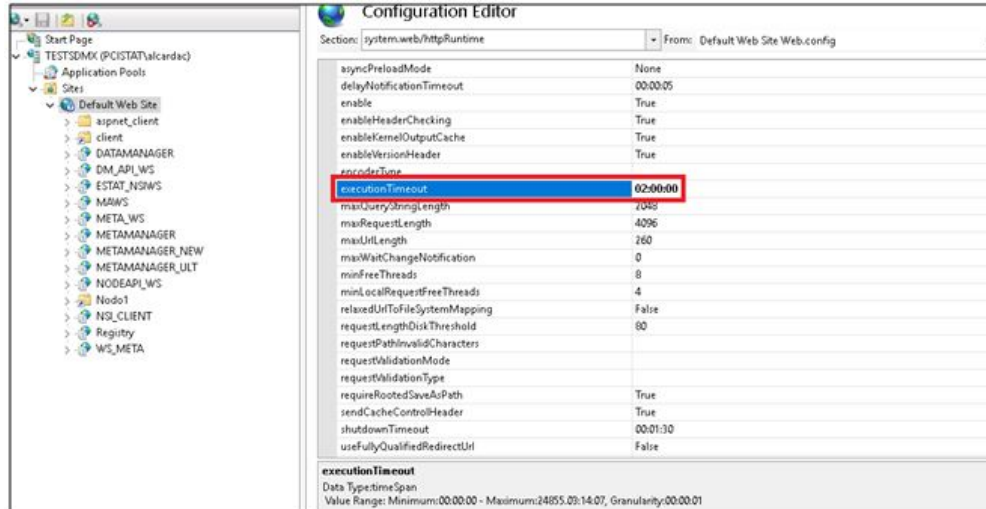


Editing of the requestTimeout parameter

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



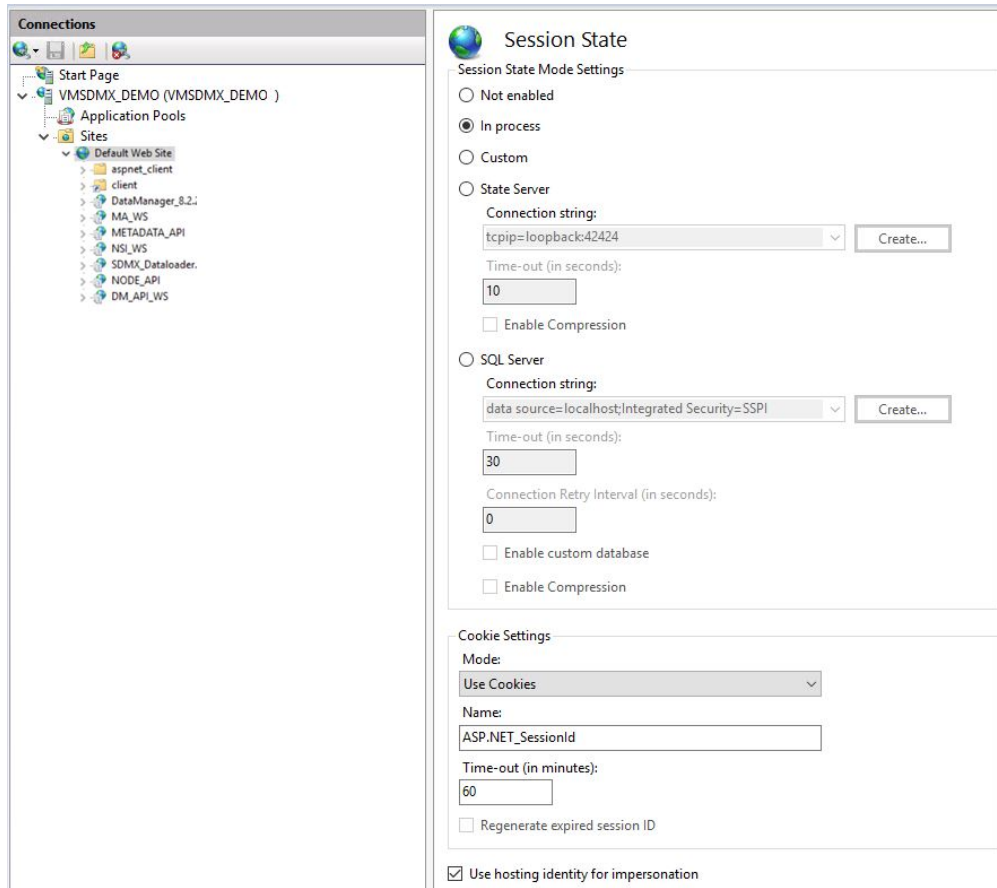
Editing the executionTimeout parameter

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)

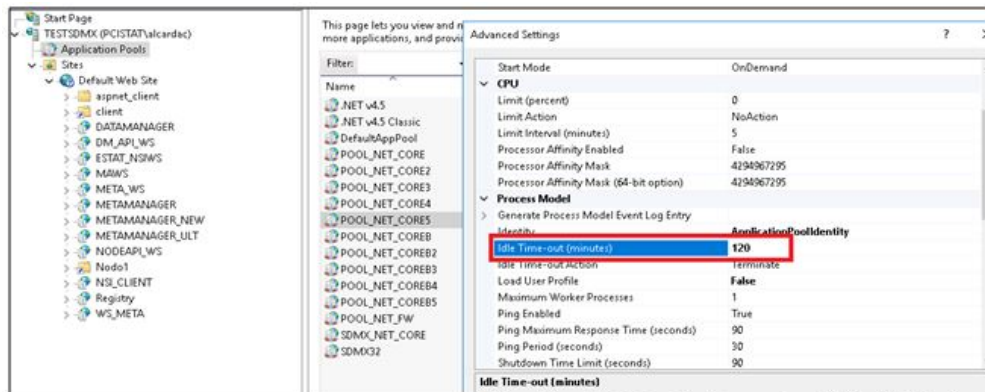


Session state Time-out parameter

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

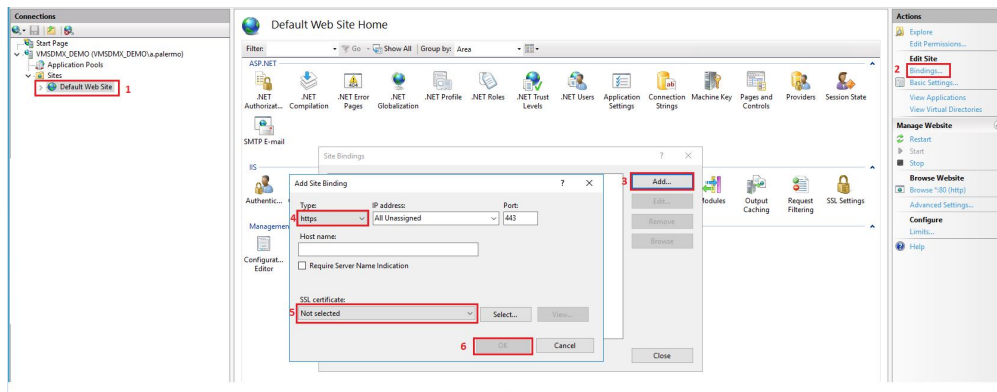


Idle Time-Out parameter

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”



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*: contains .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
- **Plugins**: contains all the plugins
- * **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

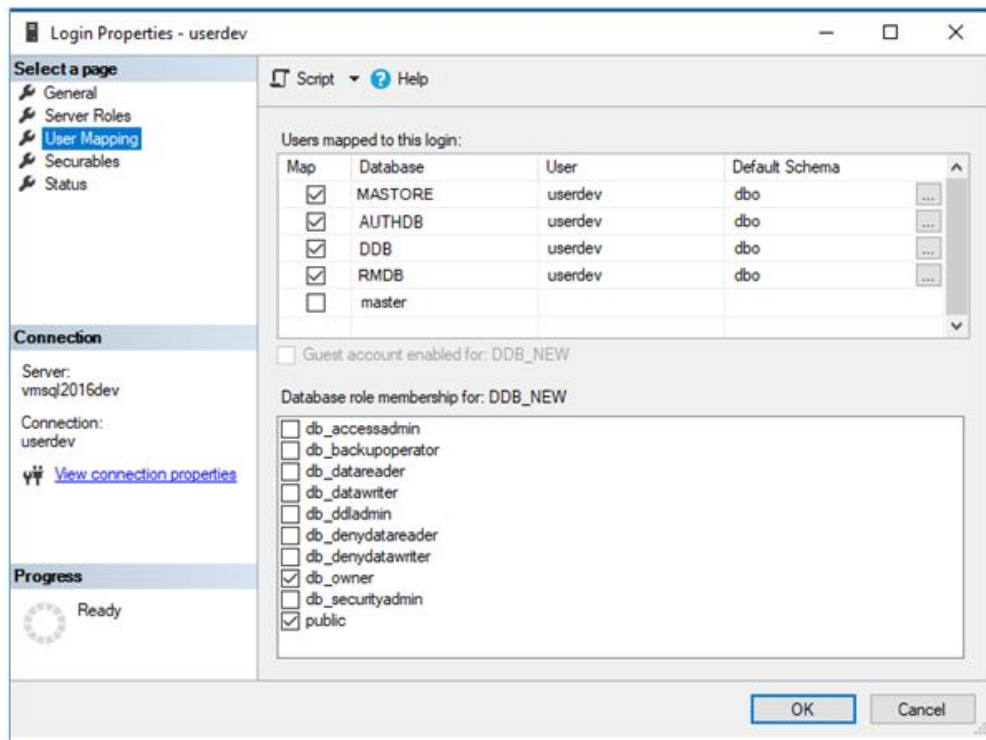
2.4 Building databases

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*.

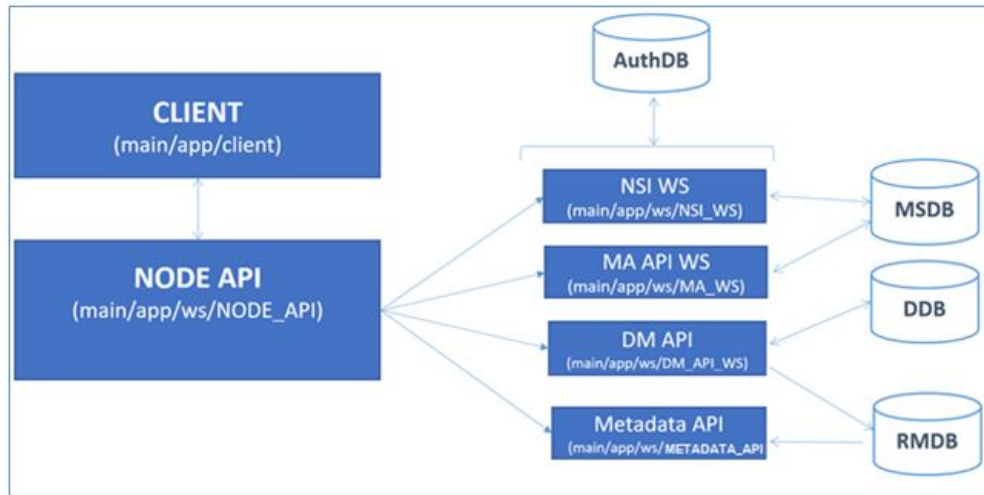
The users that will be used in the connection strings must have grants of read/write/execution on the above databases:



DB user mapping

2.5 Web Services deployment

The software supports the following architecture:

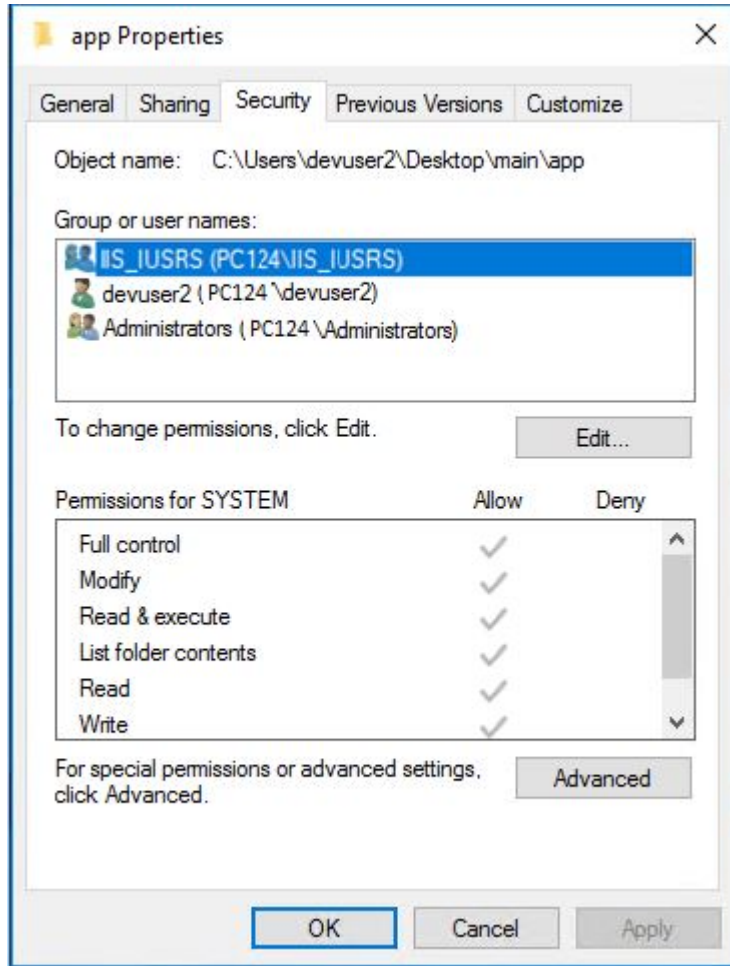


Software architecture

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.



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 sqlserver db-->
<!-- Add name="authdb" connectionString="Data Source=LOCALHOST;Initial Catalog=auth;Integrated Security=False;User ID=sa;Password=test" />
<add name="authdb" connectionString="Data Source=SOURCE;Initial Catalog=AUTHDB;User ID=user;Password=pass" providerName="System.Data.SqlClient" />
<add name="MappingStoreServer" connectionString="Data Source=SOURCE;Initial Catalog=MSDB;Integrated Security=False;User ID=user;Password=pass" />
</connectionStrings>
<applicationSettings>
  <!-- Default header retriever -->
  <!-- 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 default log file separator -->
  <add key="log.dk.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:\instantclient_11_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 -->
  <!-- Remove 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="Esat.Ski.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"/>
  </auth>
  <authAuth>
    <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>
  </appender>
  <!-- TODO filter for dataflow logger ?-->
  <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.



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

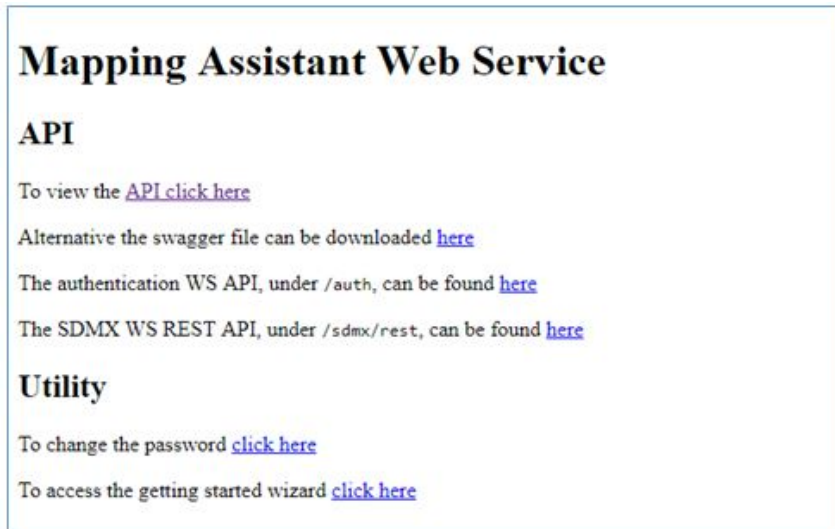
To install and configure the MA WS the following actions must be performed:

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=MASTORE;User ID=mauser;Password=mauser; providerName="System.Data.SqlClient" />
<!-- Note the "name" is the RMDB id/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.



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 and after elaboration, using the paths
 - DMApiSettings/IMPORT_FILE_BASE_DIR
 - DMApiSettings/TRANSFORMATION_BASE_DIR
 - DMApiSettings/PROCESSING_BASE_DIR
 - DMApiSettings/OUTPUT_FILE_BASE_DIR
 - DMApiSettings/AUTO_OUTPUT_FILE_BASE_DIR.

IIS_IUSRS and IUSR users must have the permissions read/write on this folders.
Pay attention to the escape character (es. \\ for \)

```

"DATA_PROVIDER_NAME": {
  "DEFAULT_DATA": {
    "DATA_STORE_TYPE": "SQL_SERVER",
    "CONNECTION_STRING": "Data Source=src;Initial Catalog=DM;Persist Security Info=True;User ID=user;Password=pw",
    "SCHEMA": "dbo"
  },
  "MR_DATA": {
    "DATA_STORE_TYPE": "SQL_SERVER",
    "CONNECTION_STRING": "Data Source=src;Initial Catalog=DM;Persist Security Info=True;User ID=user;Password=pw",
    "SCHEMA": "dbo"
  }
},
"AuthConn": {
  "AlwaysUseDefault": "DM-512",
  "AuthenticationProvider": "BASIC",
  "MutualAuthentication": "NONE",
  "Connection to the AUTHDB":
  "CONNECTION_STRING": "Data Source=src;Initial Catalog=DM;Persist Security Info=True;User ID=user;Password=pw",
  "Setup": {
    "Host": "mail.domain.it",
    "Port": 25,
    "Username": "user",
    "Password": "pw",
    "UseSSL": false,
    "FromAddress": "user@domain.it",
    "FromAlias": "Name Surname",
    "TempMailPath": "\\config\\TempMail"
  }
}
}
DMApiSettings: {
  //maximum dimension of blocks for loading operations (1 MB by default)
  "MAX_BLOCK_SIZE": "1000000",
  //maximum dimension of blocks for import operations (10 MB 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_REFROW_NUM": "1000",
  //base directory for storage files for upload operations
  "IMPORT_FILE_BASE_DIR": "C:\\Temp\\File_Import",
  //base directory for storing Transformation files
  "IMPORT_REFERENCE_METADATA_BASE_DIR": "C:\\Temp\\File_Import\\ReferenceMetadata",
  //base directory for storing Transformation files
  "TRANSFORMATION_BASE_DIR": "C:\\Temp\\File_Import\\VRLMapping",
  //base directory for storing Processing files
  "PROCESSING_BASE_DIR": "C:\\Temp\\File_Import\\VRLMapping",
  //base directory for storing processed files
  "OUTPUT_FILE_BASE_DIR": "C:\\Temp\\File_output",
  //base directory for storing processed for standalone mode
  "AUTO_OUTPUT_FILE_BASE_DIR": "C:\\Temp\\File_output_auto",
  //key for connection encryption
  "ENCRYPTION_PASSWD": "a1b1c1d1e1f1g1",
  //maximum number of elements a column can have to be selectable as measure dimension (NOT USED)
  "MAX_NUM_DIM_MEAS_DIM_COL": "4",
  //maximum time for import data in cube
  "MAX_TIME_CUBE_IMPORT_DATA": "120"
}

```

File appsettings.json

5. Copy into the folder Plugins your plugins. Create one folder for each plugin and copy into the folder the dll.
6. Modify the file *main/app/ws/DM_API_WS/config/base/logconfig.xml* to configure the log.
7. 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

8. 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 url of the MA WS (e.g. *http://localhost/MDM/ws/MA_WS*) and Username/Password of the Administrator (by default admin/[empty string]).

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

Installation Wizard – AuthDB initialization

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



Installation Wizard – Extension of the AuthDB

2.6.3 Check of MSDB status

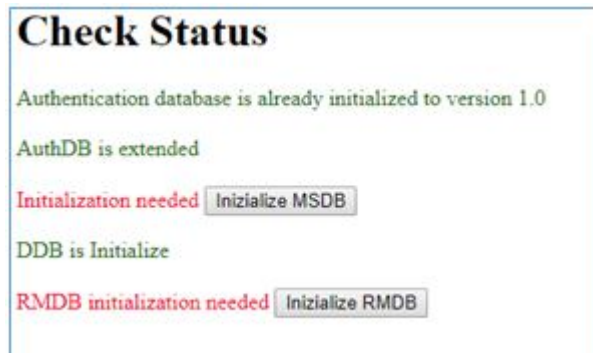
Select the MappingStoreServer and click on the button [Check]



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.



Installation Wizard– initialization of the DBs

2.6.5 Disable the Wizard

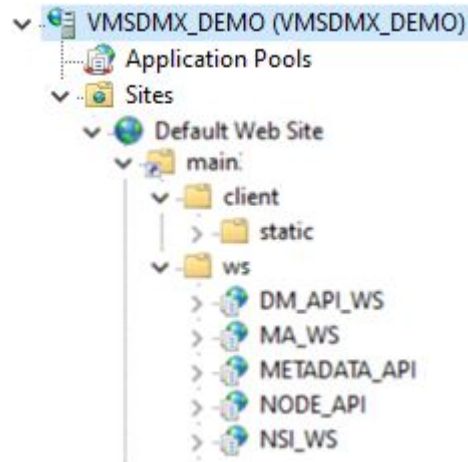
After these steps have been done, for security reason we suggest to disable the access to the following wizard's methods. To do that you have to add the following emphasized lines in the file `main/app/ws/DM_API_WS/web.config`:

```
<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>
  </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.

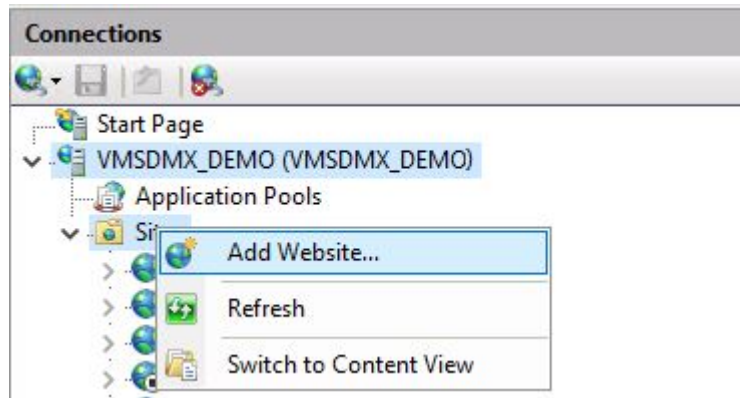
The configuration of the module is detailed in the paragraph [2.10.8 Configurations and contexts Metadata API](#)

2.7.1.1 Deploy Report HTML

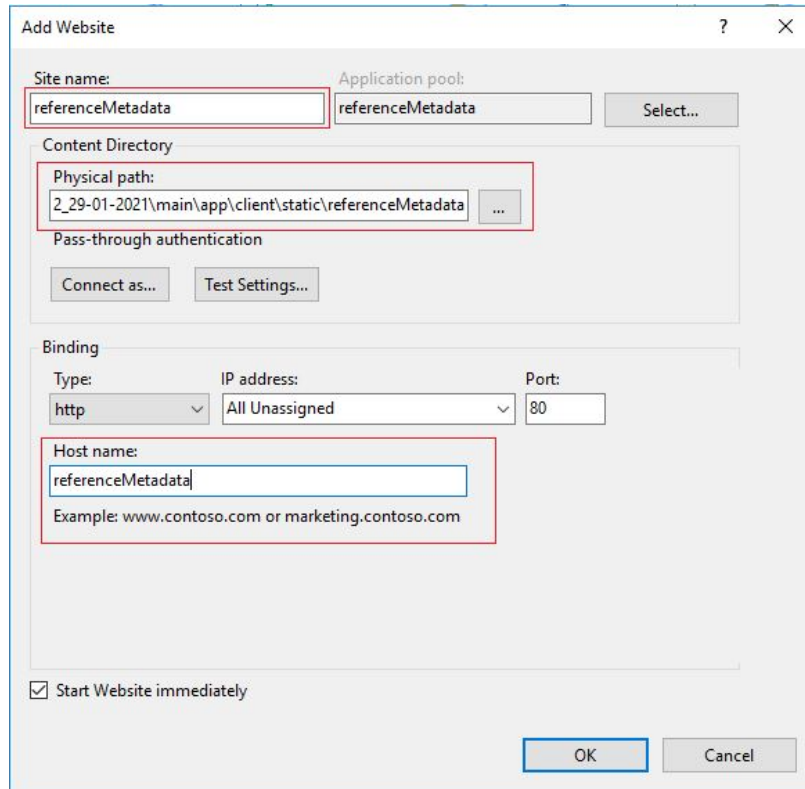
The HTML components exposed and integrated into the client (discussed in the previous paragraphs), can also be used in standalone mode (without using the MDM client).

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:



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 <>):

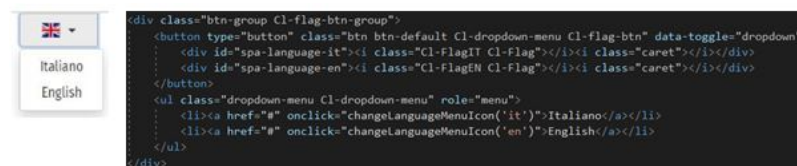
http://referenceMetadata/CategoryTemplate.html?nodeId=<NODE_ID>&metadataSetId=<ID_METADATASET_DCAT_AP_IT>&lang=&BaseUrIMDA=<BASE_URL_METADATA_API>&BaseUrIMDM=<BASE_URL_CLIENT>

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



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">
  <div style="font-size: 30px;">Themes</div>
</div>
<div id="mmu-en" class="ctools-content">
  <div class="facetapi-terms facetapi-facet-field-topic facetapi-processed" id="facetapi-facet-search-apidatasets-block-field-topic-en">
    <ul class="leaf">
      <li class="leaf">
        <a onclick="loadMustache('Agriculture, fisheries, forestry and food','en')" id="facetapi-link-en-63">
          Agriculture, fisheries, forestry and food
        </a>
      </li>
    </ul>
  </div>
</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)>#`



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	
METADATASET	IT1:DCAT-AP_IT_METADATASET111(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_METADATASET111(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)	
Reference metadata	
1. Dataset	
Download Report	
1. Dataset	
1.1. Dataset identifier	DF_JUS2_DOWNL=1+1.0
1.2. Dataset title	guidance download
1.3. Dataset description	
1.4. Dataset modified	2021-02-11
1.5. Dataset theme	Justice, legal system and public safety
1.6. Dataset theme	Government and public sector
1.7. Dataset rights holder	
1.7.1. Agent identifier	ca
1.7.2. Agent name	ca
1.8. Dataset accrual periodicity	ann
1.9. Dataset distribution	
1.9.1. Distribution format	zip
1.9.2. Distribution access url	100_10000_00000_0
1.9.3. Distribution license	
1.9.3.1. License type	
1.10. Dataset contact point	
1.10.1. Name	discontatto
1.10.2. Email	0000.aaaaa@iscali.it
1.10.3. Website	100_10000_00000_0

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:



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

```
<script id="template-it" type="text/html; charset=UTF-8">
  {{<!-->}}
  <div class="page-title">
    Datasets {{<!-->}}
  </div>
  <div class="page-content">
    <div class="view">
      {{<!-->}}
      <div>
        <div class="reportTab">
          <div class="vista-generale">
            <article>
              <header>
                <a target="blank" href="{{<!-->}}&lang={{<!-->}}&metadatasetId={{<!-->}}&reportId={{<!-->}}">
                  {{<!-->}}
                </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">
                    {{<!-->}}
                  </div>
                </div>
              </div>
            </div>
            <div>
              <div id="pubDate-{{<!-->}}"><span class="Field-label">Pubblicato da: </span>{{<!-->}}</div>
              <div id="updateDate-{{<!-->}}"><span class="Field-label">Data di ultima modifica: </span>{{<!-->}}</div>
              <div><span class="Field-label">Tema: </span><ul>{{<!-->}}<li id="list-group-{{<!-->}}">{{<!-->}}</li>{{<!-->}}</ul></div>
            </div>
          </article>
        </div>
      </div>
    </div>
  </div>
  {{<!-->}}
</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

General	* SDMX WS Endpoint type:
Agencies	SOAP
Endpoint	* SDMX WS Endpoint: https://localhost/NSI_WS/SdmxRegistryService Ping
Custom annotations	Remote SDMX WS username: <input type="text"/>
Annotations	Remote SDMX WS password: <input type="text"/>
Proxy	<input type="text"/>
Search	Initial WSDL: <input type="text"/>
DCAT-AP_IT	
Data Browser	* Ping Artefact: Concept Scheme
Plugin	MA Endpoint: https://localhost/MA_WS Ping
	Post filters support : <input type="checkbox"/>
	DM Endpoint: https://localhost/DM_API_WS/api/DMApi Ping
	LDAP Endpoint: <input type="text"/>
	Active Directory Endpoint: <input type="text"/>
	Metadata API base url: https://localhost/METADATA_API Ping

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).

The screenshot shows the 'Application' configuration page. The 'User Interface' section is selected in the sidebar. The 'Languages' section is expanded, showing a table with two entries:

ID	Country code
en	gb
it	it

Below the table is a '+ Add language' button.

- 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

The screenshot shows a web application interface with a sidebar on the left and a main content area on the right. The sidebar contains several menu items: 'User Interface', 'Agencies', 'Data Management', 'Default header submit structure', 'Superuser credentials' (highlighted in red), and 'Endpoint settings'. The main content area is titled 'Application' and contains a section for '* Superuser credentials:'. This section is collapsed, indicated by a downward arrow and the word 'Collapse'. Below the collapse indicator, there are three input fields: '* Username:' with the value 'superadmin', 'Set new password:', and 'Confirm new password:'. At the bottom of this section is a button labeled '+ Add superuser'.

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 Middleware 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 Middleware 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"/>
</auth>
</estat.nsi.ws.config>
```

Configuration parameters for activating the management of the permissions rules

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

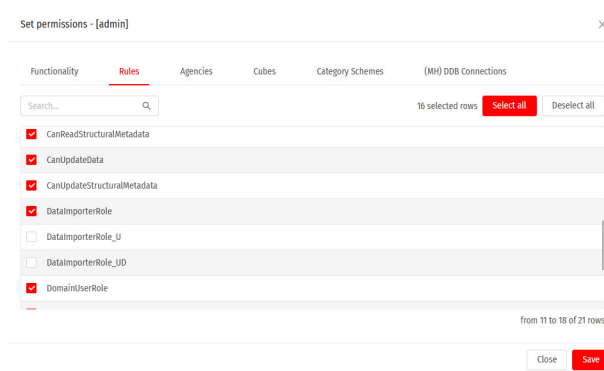
For each method of the NSI WS, in the *nsiws.xml* file inside the *App_Data* folder, are defined which rules the user has to have in order to access the various methods exposed by the web service.

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 “CanReadStructuralMetadata” 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.



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*:

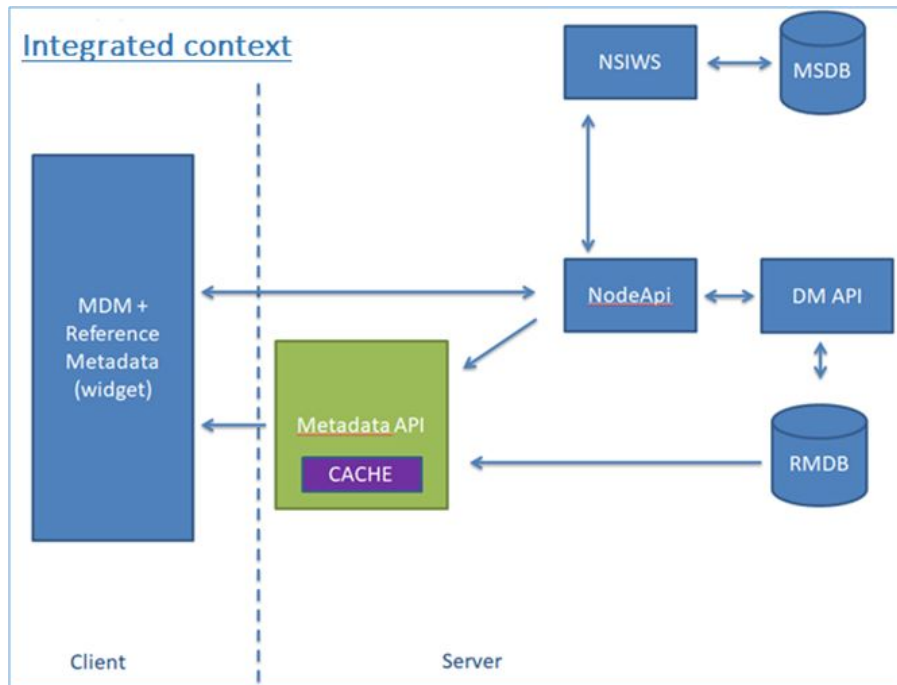
- insert in the *encryptKey.txt* file the secret encryption key (which must be the same one used in DMApi (see ENCRYPTION_PASSW key in the *appsettings.json* file mentioned above))
- enter the password you wish to encrypt in the *input.txt* file
- double click on *CriptUtilityApp.exe*
- an *output.txt* 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.



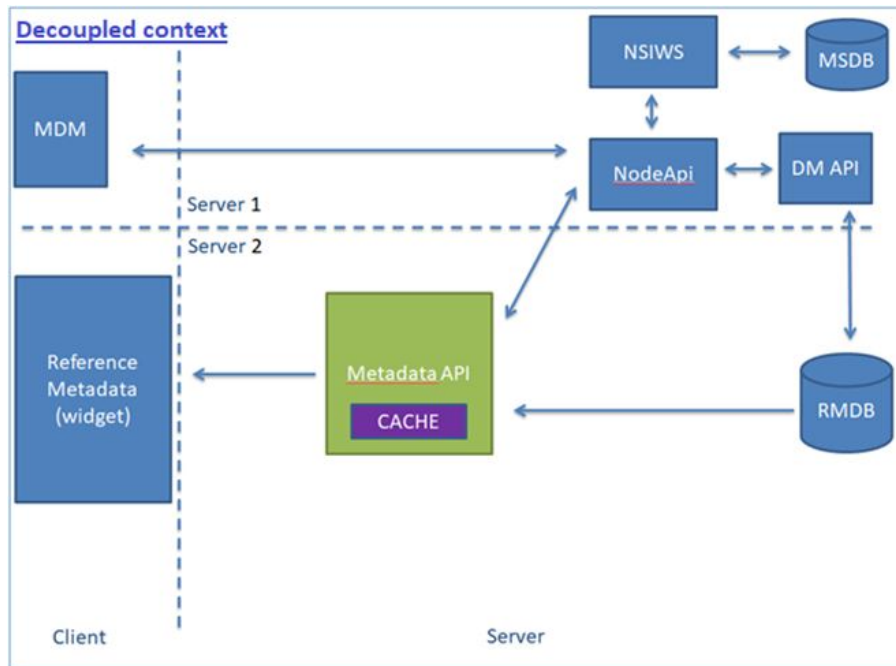
In this case, the system must be configured as follows:

- **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 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_NSIIWS** 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

- Initialize AuthDB
- Extend AuthDb
- Check Mapping Store
- Initialize Mapping Store
- Initialize DDB
- Initialize 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: ‘NI’, 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.