|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Meta & Data Manager

Installation manual

Sommario

[1 Introduction 3](#_Toc33629260)

[1.1. Aim of this document 3](#_Toc33629261)

[1.2. Acronyms and glossary 3](#_Toc33629262)

[2 Software installation 3](#_Toc33629263)

[2.1 Pre-requisites 3](#_Toc33629264)

[2.1.1 Operating system 3](#_Toc33629265)

[2.1.2 Sql Server 4](#_Toc33629266)

[2.1.3 IIS 4](#_Toc33629267)

[2.1.4 .NET Core 4](#_Toc33629268)

[2.1.5 Eurostat Web Service 6](#_Toc33629269)

[2.1.6 SSL certificate 6](#_Toc33629270)

[2.2 IIS configuration 7](#_Toc33629271)

[2.2.1 Parameters for the management of big files 7](#_Toc33629272)

[2.2.2 Other parameters 9](#_Toc33629273)

[2.2.3 https Binding 11](#_Toc33629274)

[2.3 Software package 11](#_Toc33629275)

[2.4 Building databases 12](#_Toc33629276)

[2.5 Web Services deployment 13](#_Toc33629277)

[2.5.1 NSI WS 15](#_Toc33629278)

[2.5.2 MA API WS 18](#_Toc33629279)

[2.5.3 DM API WS 18](#_Toc33629280)

[2.5.4 METADATA API 20](#_Toc33629281)

[2.5.5 NodeApi 21](#_Toc33629282)

[2.6 Database initialization 21](#_Toc33629283)

[2.6.1 Login 21](#_Toc33629284)

[2.6.2 AuthDB + Extensions MSDB 21](#_Toc33629285)

[2.6.3 Check of MSDB status 22](#_Toc33629286)

[2.6.4 Check the Status of MSDB, DDB and RMDB 22](#_Toc33629287)

[2.7 Client deployment 22](#_Toc33629288)

[2.7.1 Reference Metadata 23](#_Toc33629289)

[2.8 The first time that the application is used 23](#_Toc33629290)

[2.9 Appendix 24](#_Toc33629291)

[2.9.1 Languages management 24](#_Toc33629292)

[2.9.2 Download formats support 25](#_Toc33629293)

[2.9.3 Superadmin password change 26](#_Toc33629294)

[2.9.4 Scenario: access to a remote node in reading and writing mode 27](#_Toc33629295)

[2.9.5 Permission rules management 27](#_Toc33629296)

[2.9.6 Welcome Page configuration 29](#_Toc33629297)

[2.10 Quick steps 30](#_Toc33629298)

# Introduction

## Aim of this document

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

## Acronyms and glossary

|  |  |
| --- | --- |
| Definizione / Acronimo | Descrizione |
| .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 | Opertating System |

# Software installation

## Pre-requisites

#### Operating system

The supported operation systems are the same of those supported by .NET Core.

As follows the Microsoft Operating System supported.

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

#### Sql Server

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

#### 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

1. If they are missing, add them by right clicking and selecting ‘Add’

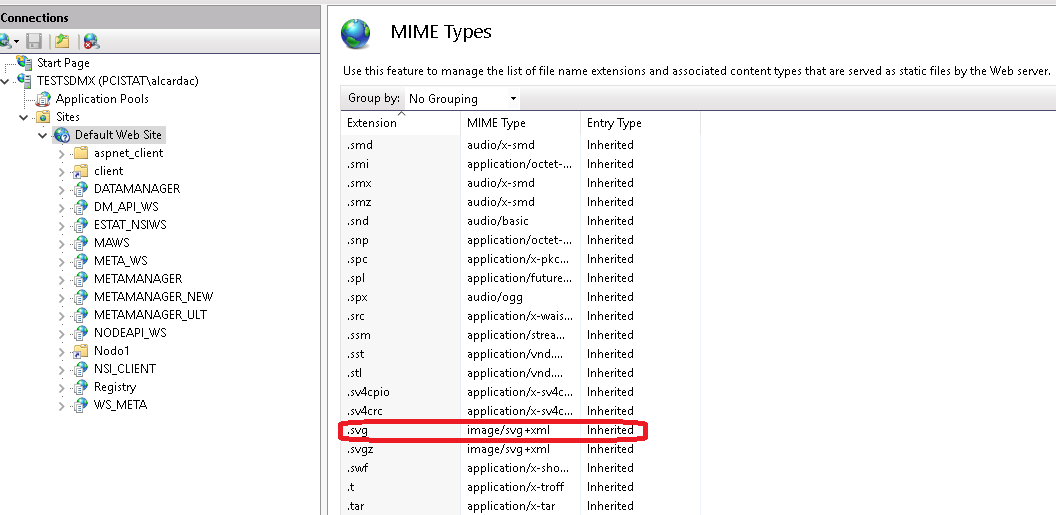


Fig. 1 – MIME types

#### .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, is possible to access to: *Control Panel/Programs/Programs and functions*:

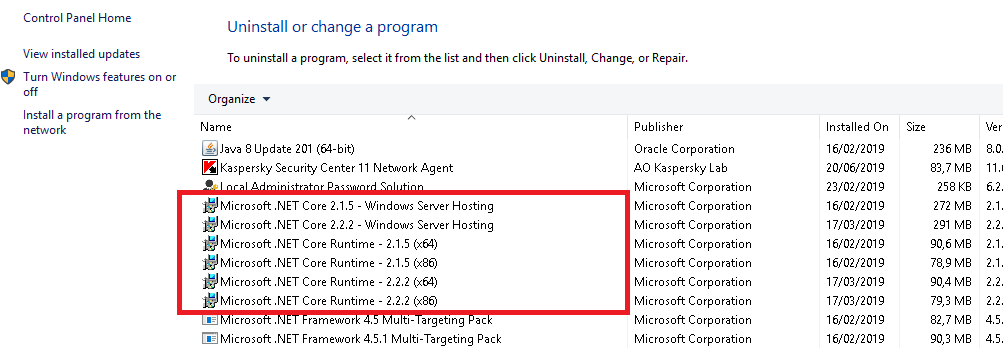


Fig. 2 – .Net Core

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!**  Is needed the installation of the *Microsoft Visual C++ 2015 Redistributable Update 3* or higher. 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>.

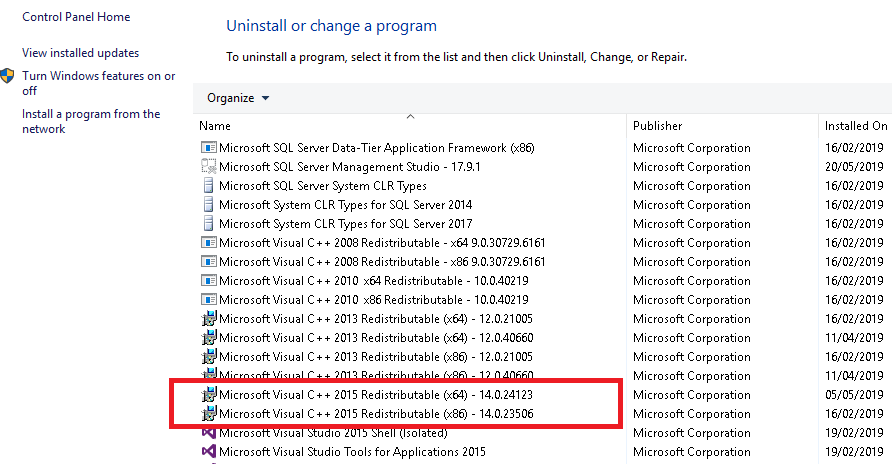


Fig. 3 – Microsoft Visual C++ installed modules

#### Eurostat Web Service

* **MA API WS** version 1.26.1 with MSDB in version 6.9
* **NSI WS** version 7.12.0 or higher supporting write mode, with the following characteristics:
  + MSDB version 6.9
  + 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 7.9.0.

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

* **NSI WS:** 7.12.1
* **MA WS:** 1.26.1

#### SSL certificate

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

## IIS configuration

First of all, it is needed the creation of an application pool for each .NET Core application. Is possible to create a new application pool by right-clicking on “Application Pool” (see Fig. 5) and by clicking on ‘*Add Application Pool’* Item).

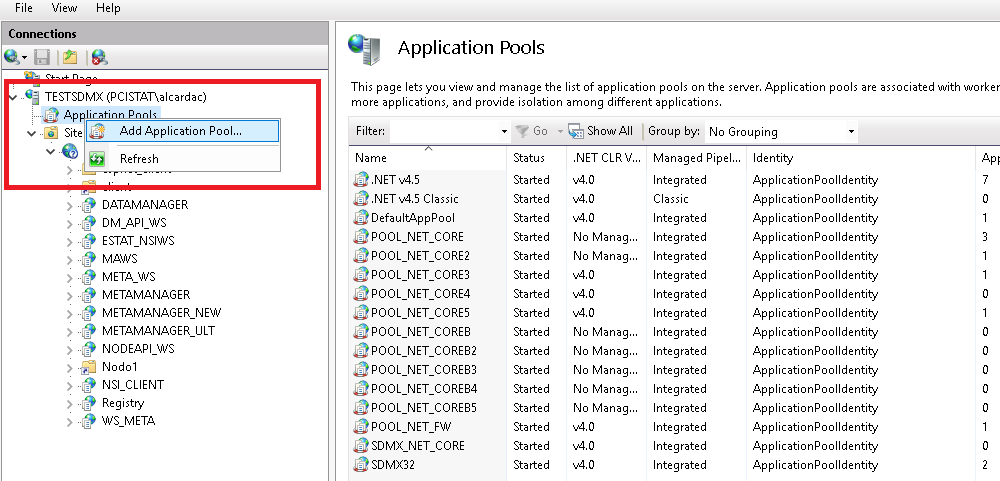


Fig. 4 – Add a new application pool in IIS

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**).

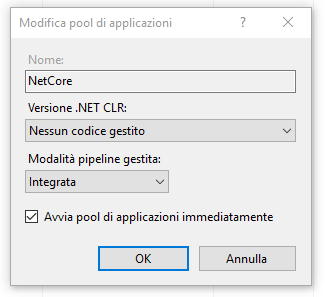


Fig. 5 - Application pool IIS for .Net Core

#### 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.

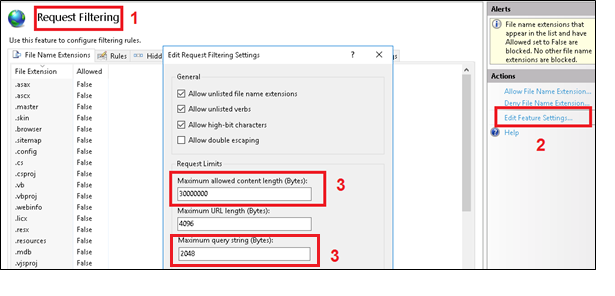


Fig. 6 - Editing of the “Maximum Allowed Content Length” parameter

**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 Timeout* parameter to the desired value.

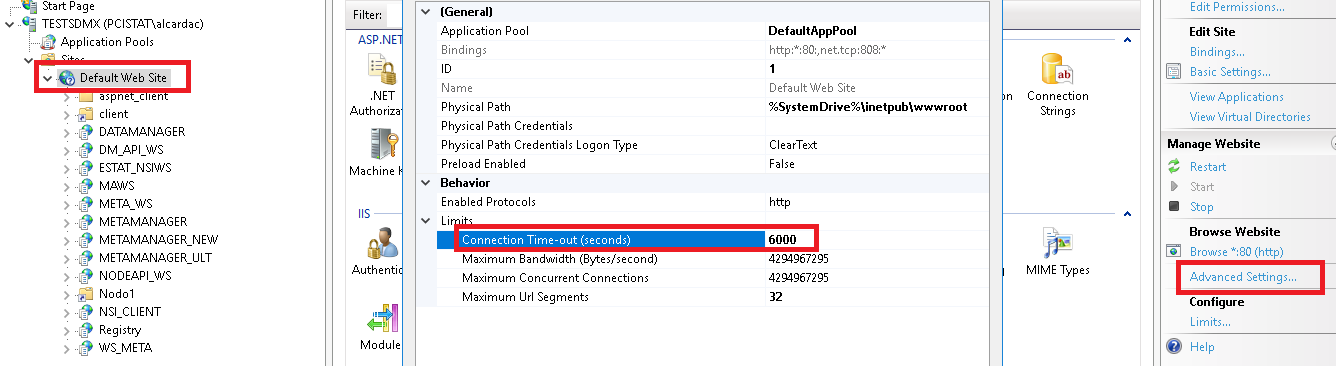
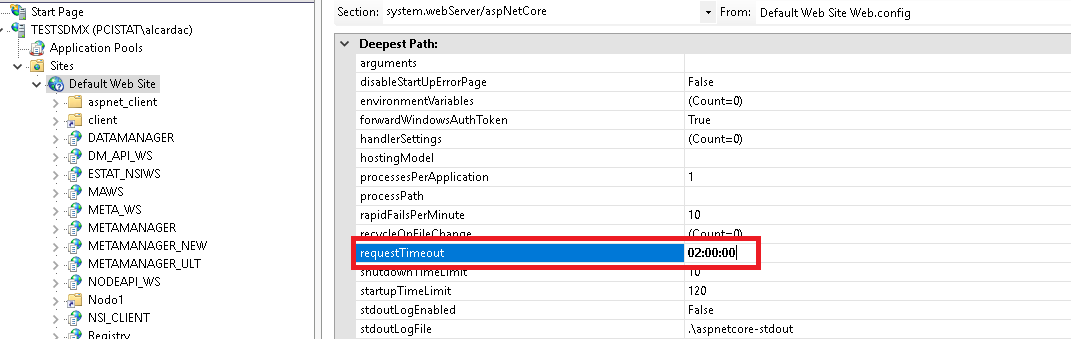


Fig. 7 - – Editing of the Connection Timeout parameter

**Request TimeOut**

This parameter allows to increase the time interval after which a timeout 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

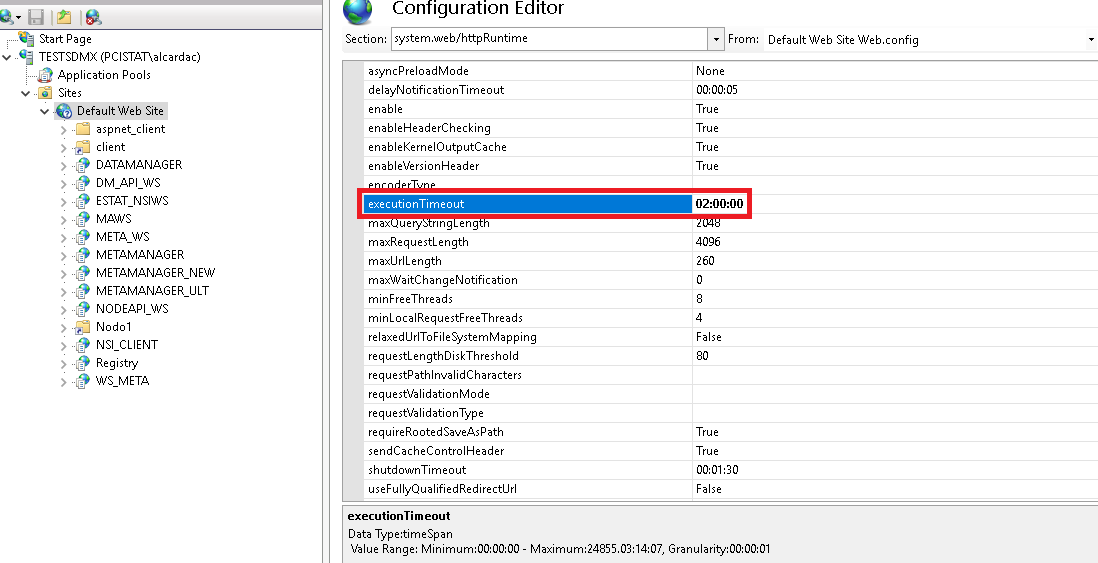


**Fig. 9 – Editing of the *requestTimeout* parameter**

**Execution TimeOut**

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. 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



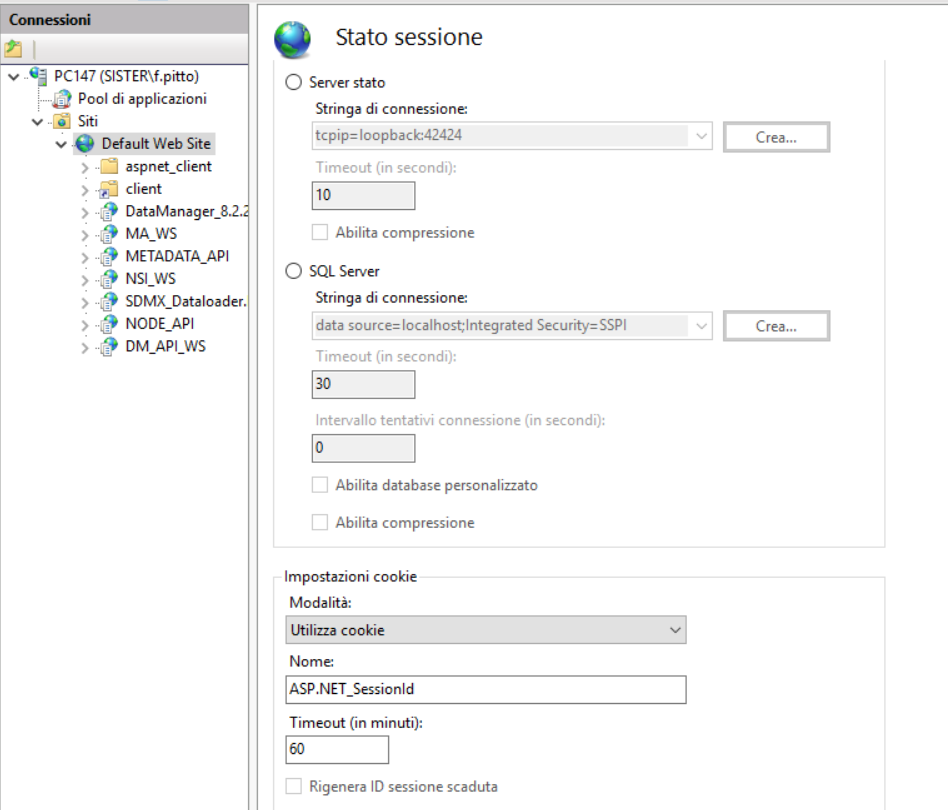
**Fig. 8 – Editing the *executionTimeout* parameter**

#### 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 constraining 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)

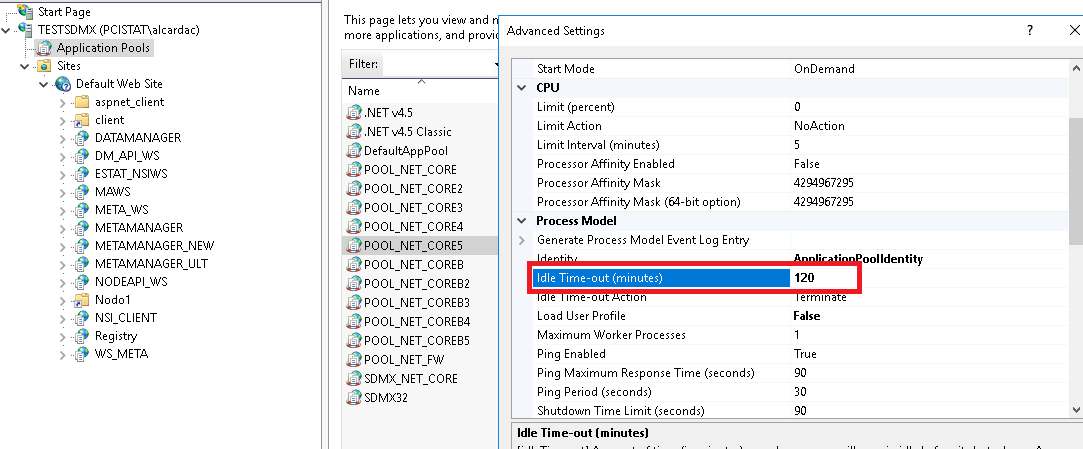


**Fig. 9 – *TimeOut Stato Sessione parameter***

**Idle Time Out**

This parameter determines the time after which an idle web service is stopped. 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

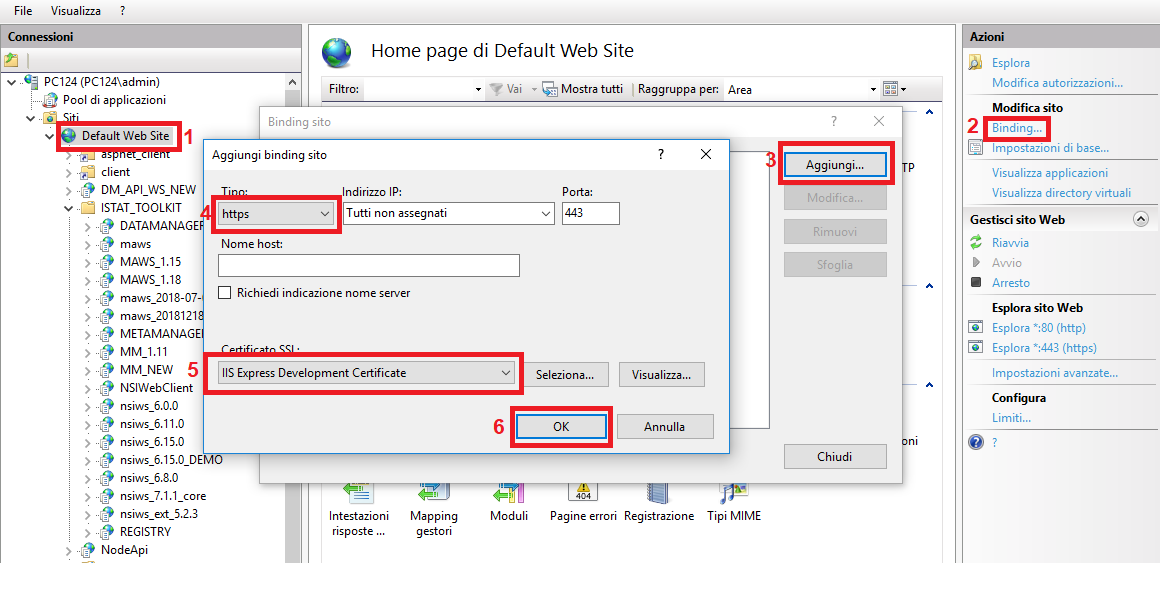


**Fig. 10 – Idle TimeOut parameter**

#### 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 http sas type
5. Select an available SSL certificate
6. Click on “OK”



## 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 file 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 in format .png
      * config.json: file di configurazione del client
      * favicon.ico: favicon dell’applicazione
    - 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 run manually 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 (for the time being only the installation manual)
* **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)

## Building databases

The following database 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:

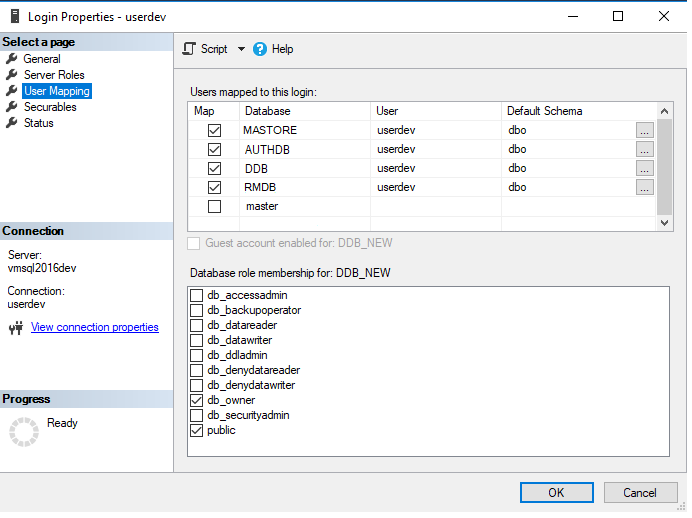


Fig. 11 – DB user mapping

## Web Services deployment

The software supports the following architecture:

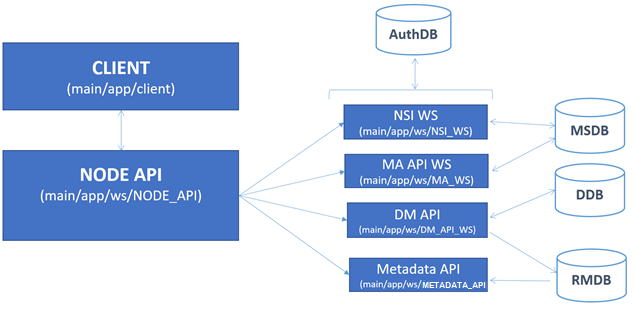


Fig. 12 – Software architecture

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*:

1. Click right on the folder;
2. Select *Property*/*Security*;
3. Click on *Edit*/*Add*;
4. In the section “*Locations”*, select the local computer;
5. In the section “*Enter the object name to select*” write IIS\_IUSRS;
6. Click on “*check names*” and then OK;
7. In the section “*Permission for IIS\_IUSRS*” include “*full control*”;
8. Repeat steps from 3 to 6 for user IUSR
9. In the section “*Permission for IIS\_IUSRS*” include “*write/read*” permissions.

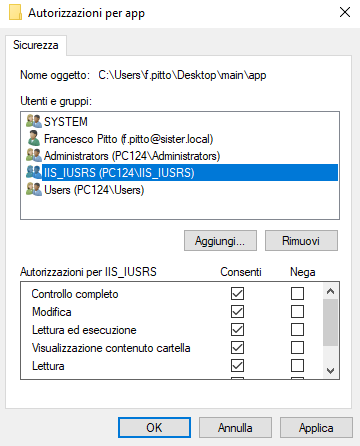


Fig 7 – modify the permissions for the *app* folder

The following Web service must be installed:

* NSI WS
* MA API WS
* DM API WS
* METADATA API
* NodeApi

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

#### NSI WS

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

1. In IIS manager, click right 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 stringsfor MSDB and AuthDB databases.

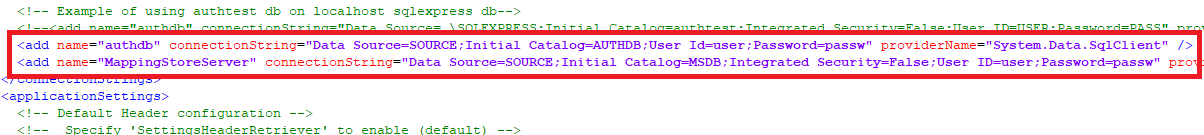


Fig. 13 – set the connection strings for MSDB and AUTHDB databases

1. Set the following parameters as follows:
   * *“enableSubmitStructure"* = “true"
   * “*ignoreProductionFlagForStructure"* = “true"
   * *“InsertNewItems”* = "true"



Fig. 14 – a section of the file app.config

1. Modify the file *main/app/ws/NSI\_WS/log4net.config* to configure del log.

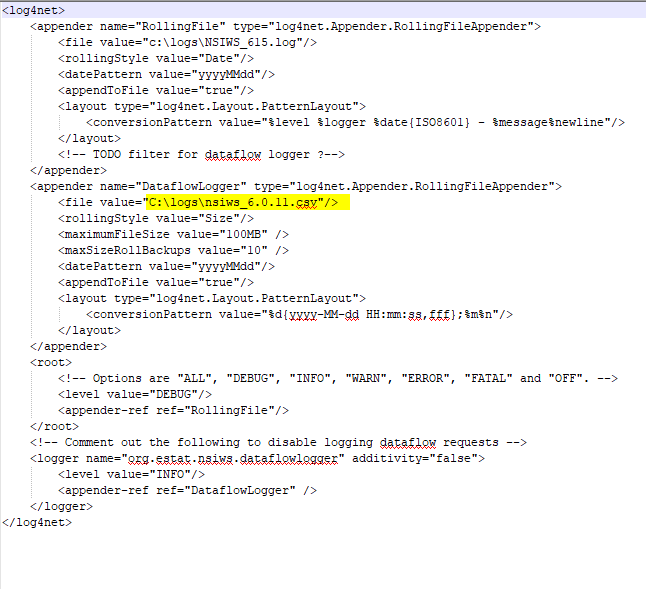


Fig. 15 – a section of the file log4net.config

1. Check the configuration using the following request: *https://localhost/NSI\_WS*. The web service helper of the web service is shown.

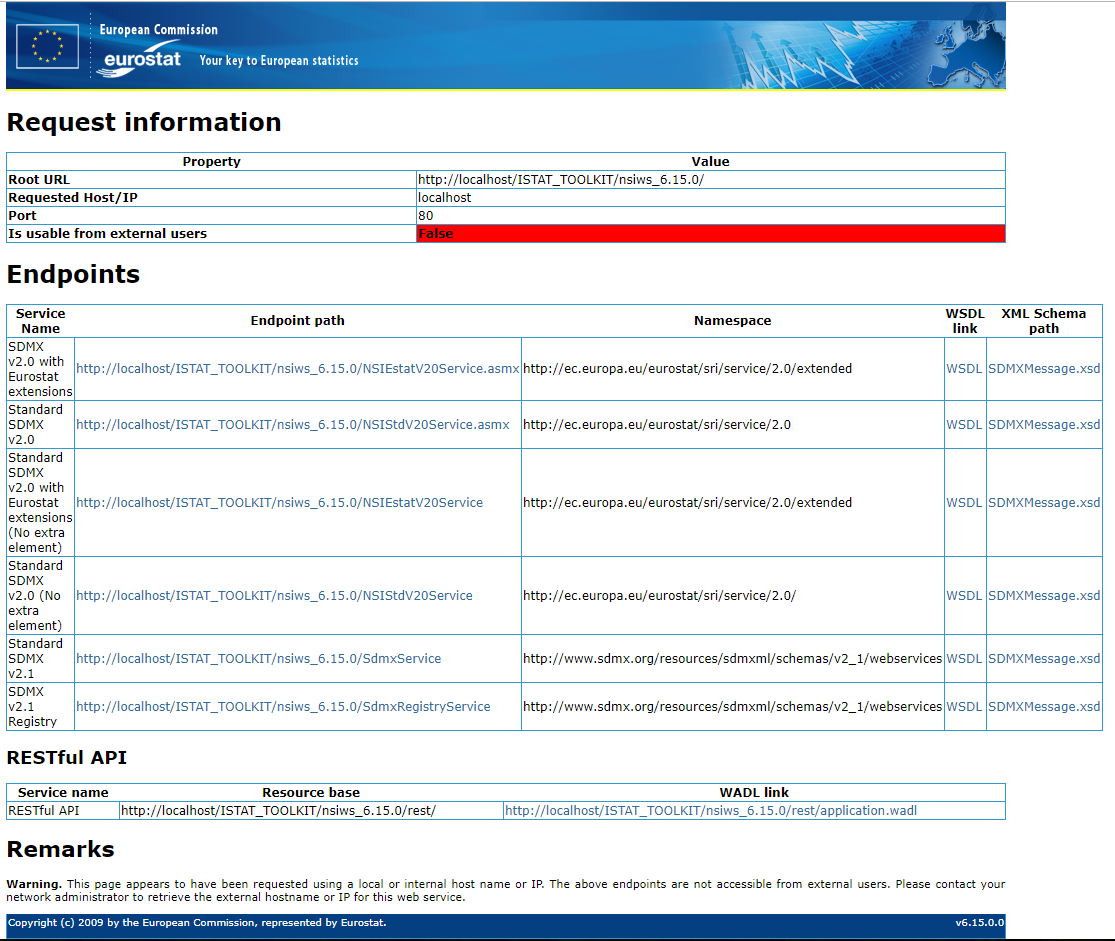


Fig. 16 – the NSI WS helper

1. The request *https://localhost/NSI\_WS/rest/codelist* will answer with a set of code lists that are installed by default.

#### MA API WS

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

1. In IIS manager, click right 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.

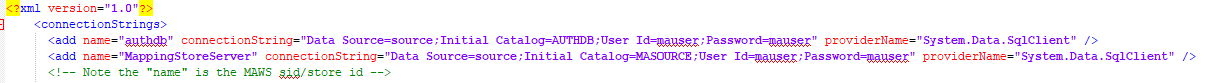


Fig. 17 – a section of the file ConnectionString.config

1. Modify the file *main/app/ws/MA\_WS/log4net.config* to configure the log.
2. Check the configuration using the following request: *https://localhost/MA\_WS*. The helper of the web service is shown.

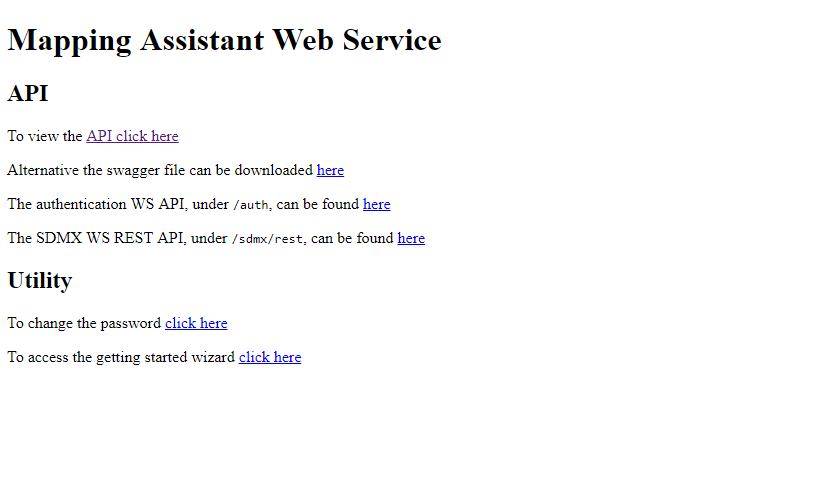


Fig. 18 - MA API WS helper

#### DM API WS

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

1. In IIS manager, click right 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\_PRIOVIDER\_NAME/RM\_DATA/CONN\_STR**

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

1. Modify the path where the files will be stored during the uploading, using the **DMApiSettings/IMPORT\_FILE\_BASE\_DIR** parameter. IIS\_IUSRS and IUSR users must have the permissions read/write on this folder.

Pay attention to the *escape character (es. \\ for \)*

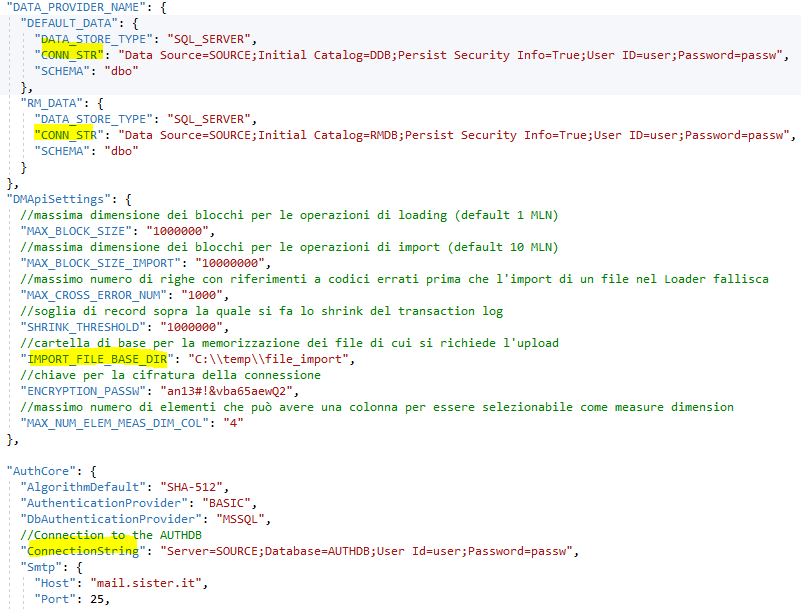


Fig. 19 – file *appsettings.json*.

1. Modify the file *main/app/ws/DM\_API\_WS/config/base/logconfig.xml* to configure the log.
2. Check the web service through the request [*https://localhost/DM\_API\_WS/api/DMApi/Ping*](https://localhost/DM_API_WS/api/DMApi/Ping). If the web service is configured correctly, the version of the web service will be shown.



Fig. 20 - DM API WS is configured correctly

1. The request *https://localhost/DM\_API\_WS/swagger* shows the list of the methods exposed by the web service.

#### METADATA API

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

1. In IIS manager, click right 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.

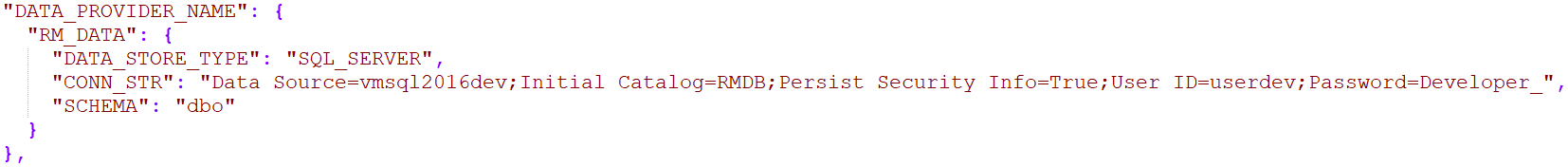




Fig 21 – configuration of the METADATA API.

1. Modify the values of the **NodeConfig** and **NodeId** keys, so that they point to the absolute path of the **nodesConfig.json** file and the default node of the **NodeApi** module, described in the next paragraph.
2. Modify the file *main/app/ws/METADATA\_API/config/base/logconfig.xml* to configure log.
3. Check the web service through the request [*https://localhost/METADATA\_API/it/api/3/action/package\_search?sort=id+asc&start=0&rows=100*](https://localhost/METADATA_API/it/api/3/action/package_search?sort=id+asc&start=0&rows=100). If the web service is configured correctly the answer is like in the below picture

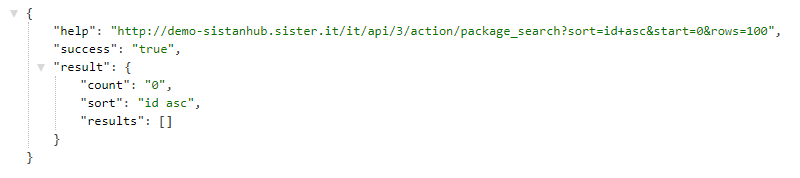


Fig. 22 - METADATA API WS configured correctly

1. The request *https://localhost/METADATA\_API/swagger* shows the list of the methods exposed by the web service.

#### NodeApi

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

1. In IIS manager, click right 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.

## Database initialization

The initialization wizard at the URL: *<WebSiteBasePath of the DM API>/Wizard/Home,* for example[https://localhost/DM\_API\_WS/*Wizard/Home*](https://localhost/DM_API_WS/Landing/Start) .

The initialization actions must be performed by a user with the permissions of administrator.

#### 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*]).

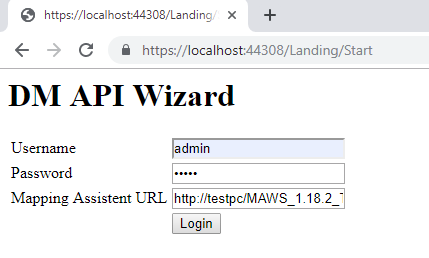


Fig. 23 – Installation Wizard - Login

#### AuthDB + Extensions MSDB

As follows the steps that must be performed in case the AUTHDB is not still initialized.

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

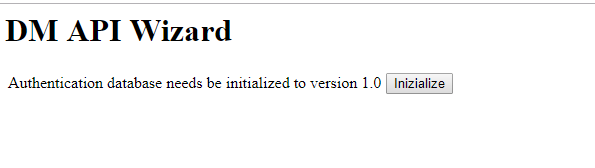


Fig. 24 – Installation Wizard – AuthDB initialization

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

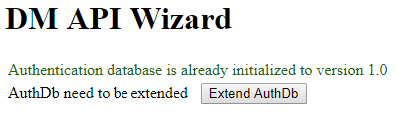
**

Fig. 25 – Installation Wizard – Extension of the AuthDB

#### Check of MSDB status

Select the MappingStoreServer and click on the button [Check];

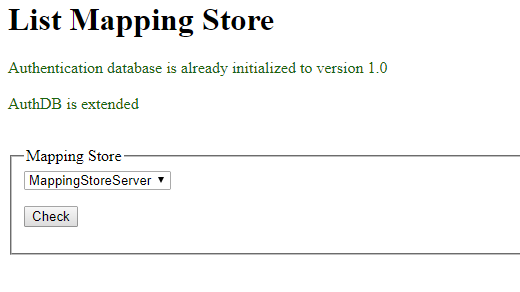


Fig. 26 – status Wizard – Mapping Store

#### Check the Status of MSDB, DDB and RMDB

In this screen 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.

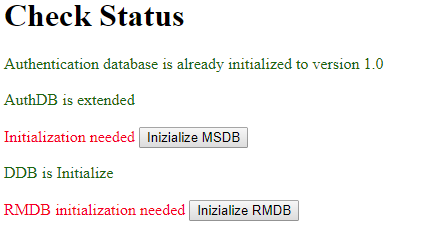


Fig. 27 – Installation Wizard– initialization of the DBs

## Client deployment

Below the steps to install and configure the client module.

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

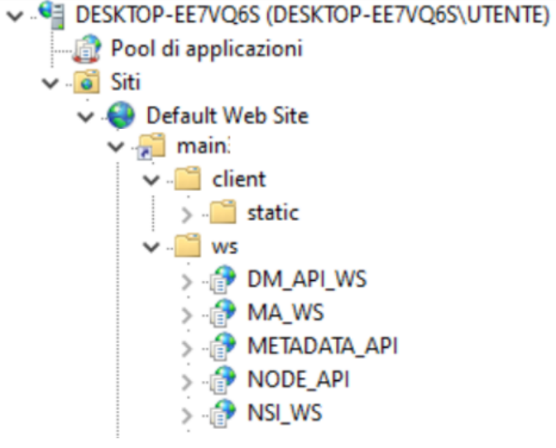


Fig. 28 - How IIS applications should appear

1. 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*](https://localhost/NODE_API).

#### Reference Metadata

The following files in **referenceMetadata** folder of the client must be configured:

* **baseUrl** entry in the **metadataapi.json** file with the MetadataApi url.

## The first time that the application is used

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*](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

|  |  |
| --- | --- |
|  |  |

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

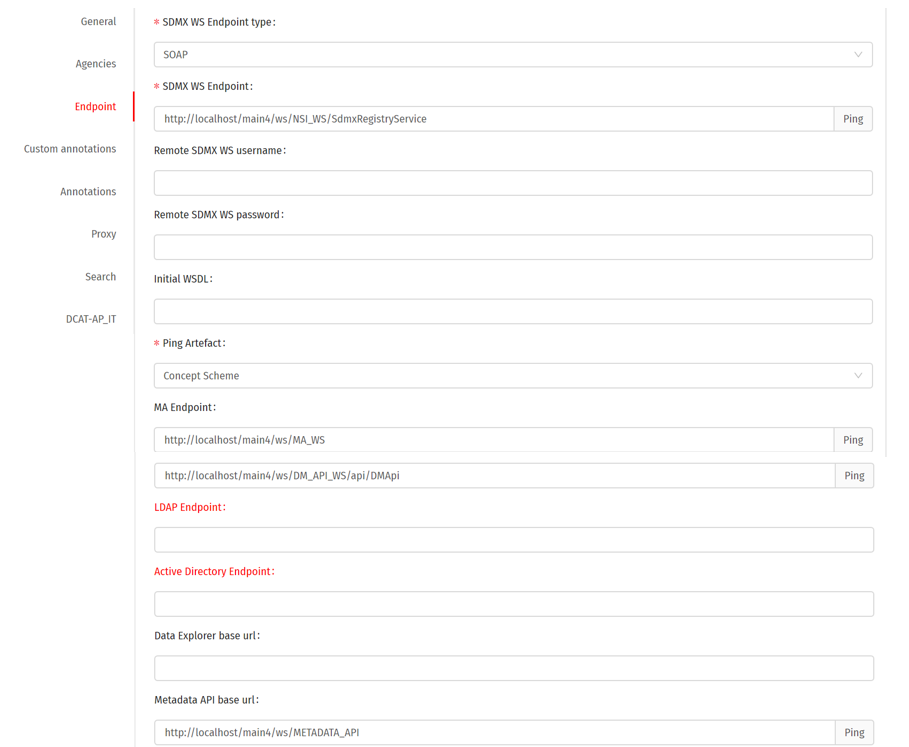


Fig. 29 – Endpoints configuration for a specific Node.

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.

1. Through the menu “Import structure”, import the files contained in the folder *main/files/ReferenceMetadata*, that contains the artefacts used for DCAT\_AP\_IT
2. 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.
3. Through the menu “User management” add new users, and through the menu “Set permissions” assign the suitable authorizations to the new users.

## Appendix

#### Languages management

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

1. Enter in 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* sectionand the code for the flag to be shown for the language in the field ‘Country Code’ (eg. sa for South Arabia).

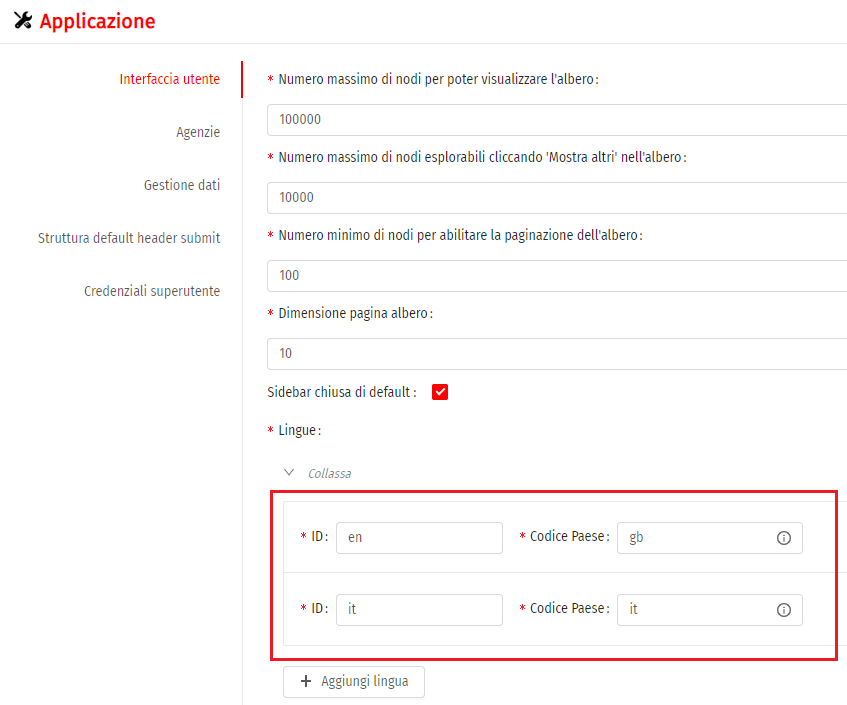


Fig. 30 – Add a new entry in the field *Languages* in order to localize the GUI

To add a new data language:

* Access to 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").

#### 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*.

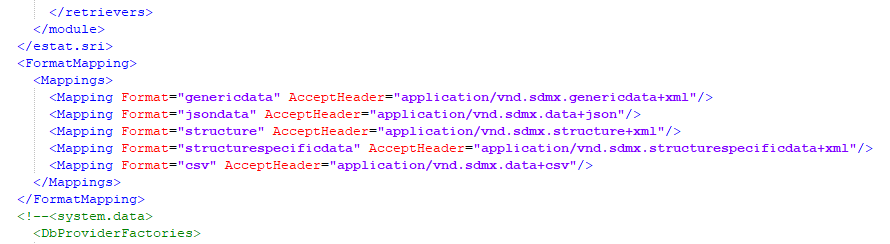


Fig. 31 – Supported formats for download

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

|  |  |
| --- | --- |
| **Formato** | **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 |
| structurespecificdata | 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.

#### Superadmin password change

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

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

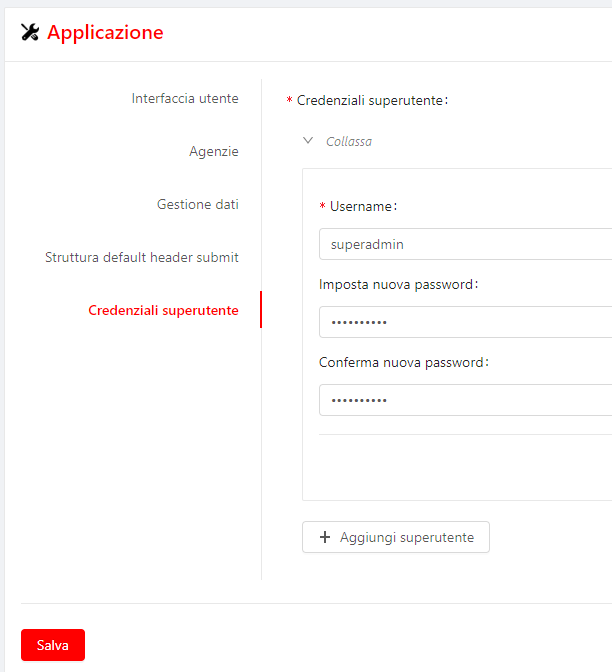


Fig. 32 – Change superadmin credentials

#### 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 is necessary to access in reading and writing mode. In order to access in writing mode to the node is necessary to install and configure for 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) is sufficient to authenticate by using the credentials of a user registered into that AUTHDB database. If instead the SDMX endpoint has the Authentication Middlware enabled (and configured with an its own AUTHDB database, that we call as AUTHDB\_SDMX database)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 have 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 del ws to be used for the authentication

#### 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, have to be set the following configuration parameters in the *app.config* file of the NSI WS or in the *Estat.Sri.Mapping.Ws.ServiceCore.dll.config* file of the MA WS.

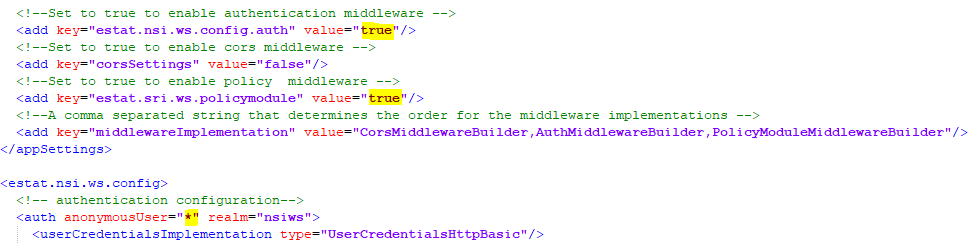
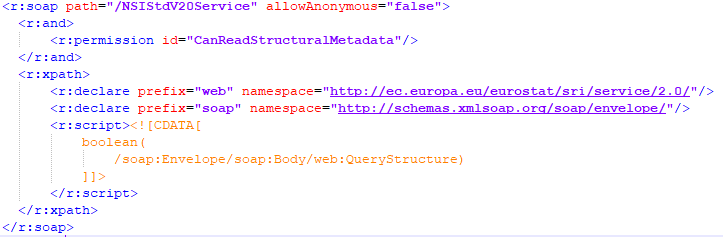


Fig 33 – Configuration parameters for activating the management of the permissions rules

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

For each method of the NSI WS is defined in the *nsiws.xml* file inside the *App\_Data* folder what are the rules that the user has to have in order to access to the various mothods exposed by the web service.

For instance the rule:



says that in order to access to paths as “Envelope/Body/QueryStructure” is needed the “CarReadStrucuturalMetadata” permission 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.

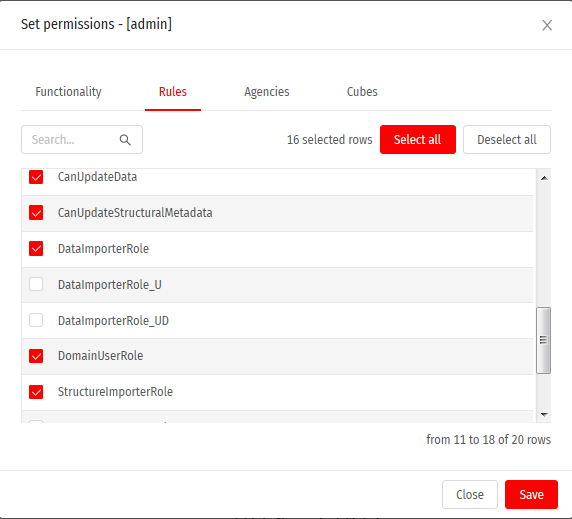


Fig 34 – Management of the rules for the users

Some permissions are hierarchical and then if for instance is assigned to a user the ‘AdminRole’ rule, he implicitely receives all the other rules on cascade.

If, for instance, a user hasn’t got the *“CanImportStructures”* right o a rule (role) that indirectely implies that permission (as for example “DataImporterRole”) he will not able to access to the ‘Import Structure’ function and will receive an error response as ‘Forbidden’.

#### Welcome Page configuration

To configure and/or customize the Welcome Page, modify the file main\client\static\homePage.html by inserting the desired contents.

## Quick steps

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

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.

1. **Creation of the DBs**
   1. **AUTHDB:** create a new db and call it **AUTHDB**;
   2. **MSDB:** create a new db and call it **MASTORE**;
   3. **DDB:** create a new db and call it **DDB**;
   4. **RMDB:** create a new db and call it **RMDB**;
   5. Create a user with Administrator rights and with read/write permissions on the above dbs.
2. **Web services deploy**

* Assegn to the IIS\_IUSRS e IUSERS users read/write grants to the *main/app* folder.
  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.
  1. **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.
  1. **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**
  + **DMApiSettings/IMPORT\_FILE\_BASE\_DIR** must indicate the base filesystem path in which will be saved the loaded data files.
  1. **METADATA API**
* Create under the IIS Default Web Site a new application with “**METADATA\_API”** alias, **POOL\_METAWS** application pooland 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.
* In the same file, modify the values of the **NodeConfig** and **NodeId** keys, so that they point to the absolute path of the **nodesConfig.json** file and to the default node of the **NodeApi** module.
  1. **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*.

1. **DBs initialization** 
   1. **Initialization Wizard**

* Access to the Wizard from the DM API WS
* Inizialize AuthDB
* Extend AuthDb
* Check Mapping Store
* Inizialize Mapping Store
* Inizialize DDB
* Inizialize RMDB

1. **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 thefile **metadataapi.json** file in the folder **referenceMetadata** of the client with the MetadataApi url.

1. **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 clocking on the related ‘Ping’ buttons.
   * *Warning!* After the import of the CategoryScheme for categorizing cubes will be needed to access, from the main menu, to 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.